O'ZBEKISTON RESPUBLIKASI
RAQAMLI TEXNOLOGIYALAR VAZIRLIGI

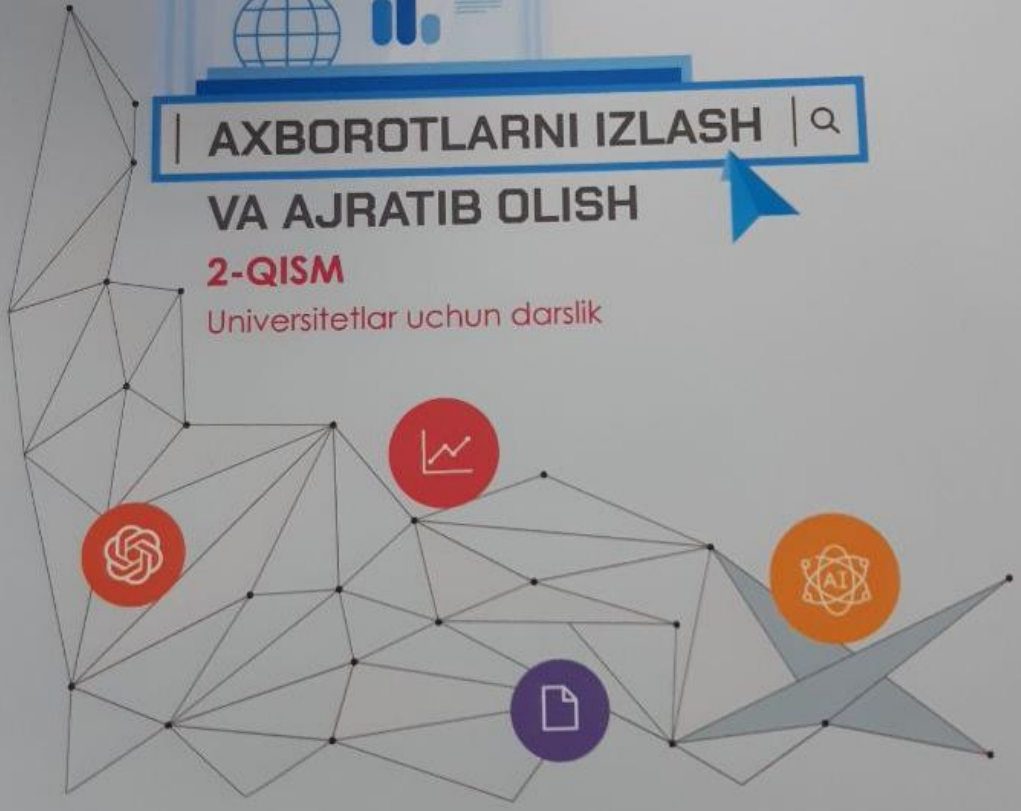
MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT
TEXNOLOGIYALARI UNIVERSITETI

Mallayev O.U.

AXBOROTLARNI IZLASH VA AJRATIB OLISH

2-QISM

Universitetlar uchun darslik



Toshkent - 2024

O'ZBEKISTON RESPUBLIKASI RAQAMLI
TEXNOLOGIYALAR VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT
AXBOROT TEXNOLOGIYALARI UNIVERSITETI

Mallayev Oybek Usmankulovich

AXBOROTLARNI IZLASH VA AJRATIB OLISH

(2-QISM)

Muhammad Al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti tomonidan darslik sifatida tavsiya etilgan

Toshkent
"METODIST NASHRIYOTI"
2024

UO'K: 004.738.5(075)
KBK: 32.973a7
M 22

Mallayev O.U.
Axborotlarni izlash va ajratib olish. (2-qism)/ Darslik /. – Toshkent: "METODIST NASHRIYOTI", 2024. - 306 b.

Darslik qidiruv tizimlari va ularni ishlash strukurasi va algoritmlarini o'rganishga bag'ishlangan bo'lib, unda asosan foydalanuvchi so'rovlarini tokenlash usullari va undagi muammolar, qidiruv tizimlari serveridagi strukturalashmagan ma'lumotlarni tezkor va sifatli qidirish usul va algoritmlari, qidiruv tizimlarida cravling va ranking jarayonlari batafsil ko'rib chiqilgan.

Darslikning dastlabki sakkiz bobi axborotni qidirish asoslariga, xususan, qidiruv tizimlarining asosiga bag'ishlangan, ushbu material ma'lumot olish bo'yicha har qanday kurs uchun asosiy bo'la oladi.

Darslik ikki qismdan iborat. Ushbu darslik ikkinchi qismi bo'lib, unda til modellari, so'rovlarning ehtimoliy modellari, matn tasnifi, Naive Bayes algoritmi, chastotaga asoslangan funksiyalarni qo'llanilishi, vektor fazosining tasniflari, vektorli mashinalarni qo'llab-quvvatlash modellari hamda algoritmlari haqida to'liq ma'lumotlar keltirilgan.

Taqrizchilar:

O'zMU, Axborot xavfsizligi kafedrasi professori, t.f.d.,
professor - **Kabulov A.V.**

Mahammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti "Axborot texnologiyalari" kefedrasi professori, t.f.d.
professor - **Yakubov M.S.**

Ushbu darslik 2022- yil 22- dekabrda Muhammad Al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti Kengashining majlisida ko'rib chiqildi va **5 (727)**- sonli buyruq bo'yicha nashr etishga ruxsat berilgan.

ISBN 978-9910-03-104-5

© Mallayev O.U., 2024.
© "METODIST NASHRIYOTI", 2024.

KIRISH

*So'nggi o'n yil ichida ma'lumotni qidirish samaradorligini tinimsiz rivojlantirish veb-qidiruv tizimlarini ko'plab foydalanuvchilarni qoniqtiradigan yangi sifat darajalariga olib keldi va veb-qidiruv ma'lumot topishning standart va afzal ko'riladigan manbasiga aylandi. Ko'pchilikni ajablantiradigan narsa shundaki, ma'lumot qidirish sohasi birinchi navbatda tartib-intizom bo'lib, **internet foydalanuvchilari** ma'lumot olishni afzal ko'rgan vositalarining asosiga aylandi. Bu darslik ushbu sohaning ilmiy asoslarini aspirantlar, shuningdek, yuqori kurs bakalavriat talabalari uchun tushunarli bo'lgan darajada taqdim etadi.*

Axborot qidirish bo'yicha ilmiy tadqiqotlarning aksariyati internetda amalga oshirildi va axborotni izlashning davomiy amaliyotining aksariyati turli korporativ va hukumat sohalarida strukturalashmagan ma'lumotlarga kirishni ta'minlash bilan shug'ullandi va bu ish darslikning asosiy poydevorini tashkil qiladi.

*Shunga qaramay, so'nggi yillarda innovatsiyalarning asosiy boshqaruvchisi o'n millionlab kontent yaratuvchilari miqyosida nashrlarni chiqargan **World Wide Web** bo'ldi. Agar ma'lumot topilmasa, izoh qo'yilmasa va har bir foydalanuvchining o'z ehtiyojlari uchun tegishli va keng qamrovli bo'lgan ma'lumotlarni tezda topa olishi uchun tahlil qilinmasa, nashr qilingan ma'lumotlarning bunday tarqalishi muammo bo'lar edi. Oxiri 10 yillarda ko'p odamlar **Internet hajmi**ning tezda o'sishi tufayli Internetdagi ma'lumotlarni boshqarish imkonsiz bo'lib qolishini his qilishdi. Ammo asosiy ilmiy yangiliklar, kompyuter jihozlari narxlarining tez pasayib borishi va **veb-qidiruv** uchun tijorat asosining o'sishi — bugungi kunda kuniga yuzlab, millionlab axborot olishning yuqori sifatli natijalarni taqdim eta oladigan asosiy qidiruv tizimlarini kuchaytirishga yordam berdi.*

*Darslikning dastlabki sakkiz bobi axborotni qidirish tizimlarida til modellarini shakillantirishga bag'ishlangan, ushbu material ma'lumot olish bo'yicha har qanday kurs uchun asosiy bo'la oladi. 1-bobda to'plamdagi har bir hujjat uchun **til modeli** yaratiladi. Undan til modeli berilgan savolini yaratish ehtimolini taxmin qilish mumkin. Bu ehtimollik hujjatlarni tartiblashda mumkin bo'lgan yana bir miqdor bo'lib xizmat qiladi. 2-6 boblarda ma'lumot qidirish, mashinani o'rganishning turli shakllari va raqamli usullar haqida ma'lumot berilgan. 2-4 boblarda hujjatlarni ma'lum toifalar to'plamiga tasniflash muammosi, ular tegishli sinflar bilan hujjatlar to'plamini hisobga olgan holda ko'rib chiqiladi. 3-bob statistik tasnifni muvaffaqiyatli qidiruv tizimi uchun zarur bo'lgan asosiy texnologiyalardan*

*biri sifatida rag'batlantiradi, kontseptual jihatdan sodda va samarali matn tasniflash usuli Naive Bayesni taqdim etadi va matn tasniflagichlarini baholashning standart metodologiyasini tavsiflaydi. 4-bobda 6-bobdagi vektor fazo modeli qo'llaniladi va hujjat vektorlarida ishlaydigan ikkita tasniflash usuli - **Rocchio** va **KNN** taqdim etiladi. Shuningdek, u matnni tasniflash muammosi uchun mos usulni tanlash mezonlarini ta'minlovchi o'quv muammolarining muhim tavsifi sifatida qarama-qarshilik almashinuvini taqdim etadi. 5-bobda ko'plab tadqiqotchilar hozirda matnni tasniflashning eng samarali usuli deb hisoblaydigan vektorli mashinalarni qo'llab-quvvatlaydi. Shuningdek, ushbu bobda tasniflash muammosi va o'quv misollari to'plamidan **ball funksiyalarini** kiritish kabi bir-biridan farq qiladigan mavzular o'rtasidagi aloqalar rivojlantiriladi. 6-8-boblar to'plamdan tegishli hujjatlar klasterlarini yaratish muammosi ko'rib chiqiladi. 6-bobda ma'lumot qidirishda tasniflashning bir qator muhim qo'llanilishi haqida umumiy ma'lumot beriladi. Keyin ikkita birlashgan algoritm ta'riflanadi: **K-means algoritm**, samarali va keng qo'llaniladigan hujjatlarni **klasterlash usuli** hisoblash uchun qimmatroq lekin ayni paytda moslashuvchanroq bo'lgan kutishni-maksimallashtirish algoritmi. 7-bob ma'lumot olishda ko'plab ilovalarda ierarxik tuzilgan klasterlar (tekis klasterlar o'rniga) zarurligini rag'batlantiradi va **klasterlar ierarxiyasini** yaratadigan bir qator **klasterlash algoritmlarini** taqdim etadi.*

*Boblar shuningdek, klasterlar uchun yorliqlarni avtomatik hisoblashning qiyin muammosini ham ko'rib chiqadi. 8-bob klasterlashning kengaytmasini tashkil etuvchi chiziqli algebradan metodni ishlab chiqadi, shuningdek, yashirin semantik indekslash yondashuvida qo'llaniladigan ma'lumotni qidirishda algebraik usullarning qiziqarli istiqbollarini taklif qiladi. 9-11-boblar veb-qidiruv muammosini ko'rib chiqadi. 9-bobda veb-qidiruvdagi asosiy qiyinchiliklarning qisqacha mazmuni va veb-ma'lumotni qidirishda keng tarqalgan texnik vositalar to'plami beriladi. 10-bobda asosiy **veb-brauzerning arxitekturasi** va talablari tavsiflanadi. Nihoyat, 11-bob **veb-qidiruvda havolalarni tahlil qilish** kuchi ko'rib chiqiladi, bu jarayonda chiziqli algebra va yuqori ehtimollikning bir nechta usullaridan foydalanadi.*

Ushbu darslikda mavzularni kengroq yoritish uchun bob yakunida bobning mavzularini tayyorlashda foydalanilgan adabiyotlar keltirilgan. Kitobxon qiziqqan mavzu yuzasida tegishli adabiyotlarni topib, o'z bilimlarini yanada rivojlantirishi mumkin.

AXBOROT IZLASH UCHUN TIL MODELLARI

Murakkab so'rovlar bilan ishlash bo'yicha foydalanuvchilarga keng tarqalgan taklif - bu tegishli hujjatda paydo bo'lishi mumkin bo'lgan so'zlarni o'ylab ko'rish va bu so'zlarni so'rov sifatida ishlatishdir. AQ uchun tilni modellashtirish yondashuvi bu fikrni to'g'ridan-to'g'ri modellashtirishdir. Agar hujjat modeli so'rovni yaratishi mumkin bo'lsa, hujjat so'rovga yaxshi mos keladi hamda bu o'z navbatida hujjatda so'rov so'zlari tez-tez uchrasa sodir bo'ladi. Shunday qilib, ushbu yondashuv darslikni birinchi qismnidagi 6.2-bo'limda ko'rib chiqilgan hujjatlarni tartiblash bo'yicha Ba'zi asosiy g'oyalarni boshqacha amalga oshirishni ta'minlaydi. AQ ga an'anaviy ehtimolli yondashuvda bo'lgani kabi **d** hujjatining **q** so'roviga taalluqliligi $\mathbf{P(R = 1|}_{q,d}\mathbf{)}$ ehtimolini ochiq modellashtirish o'rniga, tilni modellashtirishning asosiy yondashuvini ehtimollik tili yaratiladi. Model Md har bir hujjat **d** dan va hujjatlarni modelning so'rovini hosil qilish ehtimoli asosida tartiblaydi $\mathbf{P(q|\ M_d)}$. Ushbu bobda birinchi navbatda til modellari tushunchasi bilan tanishamiz (1.1-bo'lim), so'ngra AQ uchun asosiy va eng ko'p qo'llaniladigan til modellashtirish yondashuvi, so'rovlar ehtimoli modeli (1.2-bo'lim) tavsiflanadi. Tilni modellashtirish yondashuvi va AQga boshqa yondashuvlar o'rtasidagi Ba'zi taqqoslashlardan so'ng (1.3-bo'lim) tilni modellashtirish yondashuvining turli kengaytmalarini qisqacha tavsiflash bilan yakunlaymiz (1.4-bo'lim).

1.1. Til modellari

1.1.1. Cheklangan avtomatlar va til modellari

So'rovni yaratuvchi hujjat modeli deganda nimani tushunasiz? Rasmiy til nazariyasidan tanish bo'lgan tilning an'anaviy generativ modeli - satrlarni tanib olish yoki yaratish uchun ishlatilishi mumkin. Masalan, 1.1-rasmda ko'rsatilgan chekli avtomat ko'rsatilgan misollarni

o'z ichiga olgan satrlarni yaratishi mumkin. Yaratilishi mumkin bo'lgan to'liq satrlar to'plami avtomat tili deb ataladi.

Xohlayman
Men tilayman
Men tilayman, tilayman
Qaniydi hohlasam, hohlasam,
tilayman
GENERATSIYA BO'LMAYDI:
tilayman

1.1- rasm. **Oddiy chekli avtomat va u yaratadigan tildagi Ba'zi qatorlar**

Avtomatning ishga tushirish holatini va qo'sh doira tugatish holatini ko'rsatadi.

$P(STOP|q_1) = 0.2$

the	0.2
a	0.1
frog	0.01
toad	0.01
said	0.03
likes	0.02
that	0.04
...	...

1.2- rasm. **Unigram tili modeli vazifasini bajaradigan chekli avtomat**

Davlat emissiya ehtimolining qisman spetsifikatsiyasi mavjud. Buning o'rniga har bir tugun turli atamalarni yaratish bo'yicha ehtimollik taqsimotiga ega bo'lsa, ularda til modeli mavjud bo'ladi. Til modeli tushunchasi tabiatan ehtimollikdir. Til modeli - bu Ba'zi lug'atlardan olingan satrlarga ehtimollik o'lchovini qo'yadigan funksiyadir. Ya'ni, S alifbosi ustidagi **M** til modeli quyidagicha hisoblanadi:

$$\sum_{s \in \Sigma^*} P(s) = 1 \qquad (1.1)$$

Til modelining oddiy bir turi 1.2-rasmda ko'rsatilganidek, turli xil atamalar hosil qilishda yagona ehtimollik taqsimotiga ega bo'lgan yagona tugundan tashkil topgan ehtimollik chekli avtomatga ekvivalentdir, shuning uchun $\sum t \in V\ P(t) = 1$ bo'ladi. Har bir so'zni yaratgandan so'ng, to'xtash yoki aylanib o'tish va keyin boshqa so'zni ishlab chiqarishga

qaror qiladi va shuning uchun model tugatish holatida to'xtash ehtimolini ham talab qiladi. Bunday model so'zlarning har qanday ketma-ketligi bo'yicha ehtimollik taqsimotini joylashtiradi. Qurilishga ko'ra, u tarqatilishiga ko'ra matn yaratish modelini ham taqdim etadi.

1. Cheklangan avtomatlar o'z holatlariga yoki yoylariga biriktirilgan chiqishlarga ega bo'lishi mumkin. Bu yerda holatlardan foydalaniladi, chunki bu to'g'ridan-to'g'ri ehtimollik avtomatlarining odatda rasmiylashtirilgan yo'li bilan bog'liq:

Model M_1		Model M_2	
the	0.2	the	0.15
a	0.1	a	0.12
frog	0.01	frog	0.0002
toad	0.01	toad	0.0001
said	0.03	said	0.03
likes	0.02	likes	0.04
that	0.04	that	0.04
dog	0.005	dog	0.01
cat	0.003	cat	0.015
monkey	0.001	monkey	0.002
...	...	...	...

1.3- rasm. **Ikki unigram tili modelining qisman spetsifikatsiyasi**

So'zlar ketma-ketligining ehtimolini topish uchun modelning ketma-ketlikdagi har bir so'zga beradigan ehtimolini har bir so'zni ishlab chiqqandan keyin davom etish yoki to'xtatish ehtimoli bilan ko'paytiramiz. Masalan, P (qurbaqa qurbaqani yaxshi ko'rishini aytdi)

$$P = (0.01 \times 0.03 \times 0.04 \times 0.01 \times 0.02 \times 0.01) \times (0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.2) \approx 0.0000000000\ 01573 \qquad (1.2)$$

Ko'rib turganingizdek, ma'lum bir satr/hujjatning ehtimoli odatda juda kichik raqamidir! Bu yerda jarayon qurbaqa ikkinchi marta hosil qilinganidan keyin to'xtaydi. Raqamlarning birinchi qatori emissiya

ehtimoli atamasidir, ikkinchi qator esa har bir so'zni yaratgandan keyin davom etish yoki to'xtash ehtimolini beradi. Aniq to'xtash ehtimoli chekli avtomatning (1.1) tenglamaga muvofiq yaxshi shakllangan til modeli bo'lishi uchun zarurdir. Shunga qaramay, ko'pincha STOP va (1-STOP) ehtimoli qo'shilmaydi. Ma'lumotlar to'plami uchun ikkita modelni solishtirish uchun ularning ehtimollik nisbati hisoblanilishi mumkin. Bu oddiygina bo'linish natijasida yuzaga keladi.

Bir model bo'yicha ma'lumotlarning ehtimolligi boshqa modelga muvofiq ma'lumotlarning ehtimolligidir. To'xtash ehtimoli belgilangan bo'lsa, uning kiritilishi ikki til modelining satr hosil qilish ehtimolini solishtirish natijasida yuzaga keladigan ehtimollik nisbatini o'zgartirmaydi. Shunday qilib, u hujjatlar reytingini o'zgartirmaydi. Shunga qaramay, rasmiy ravishda raqamlar haqiqatda ehtimollik emas, balki faqat ehtimollarga proportsional bo'ladi.

Faraz qilaylik, ularda ikkita **M1** va **M2** til modellari mavjud va u qisman 1.3-rasmda ko'rsatilgan. Har biri 1.1-misolda tasvirlanganidek, atamalar ketma-ketligiga ehtimollik bahosini beradi. Terminlar ketma-ketligiga yuqori ehtimollik beradigan til modeli atama ketma-ketligini hosil qilgan bo'lishi mumkin. Bu safar hisob-kitoblardan STOP ehtimoli o'tkazib yuboriladi. Ko'rsatilgan ketma-ketlik uchun quyidagilarni olinadi:

Olib borayotgan AQ kontekstida to'xtash ehtimolini modellar bo'ylab o'rnatish foydalali ko'rinadi. Buning sababi shundaki, so'rovlarni yaratishda so'rovlar uzunligi taqsimoti belgilangan va til modelini yaratayotgan hujjatdan mustaqildir

s	*frog*	*said*	*that*	*toad*	*likes*	*that*	*dog*
M_1	0.01	0.03	0.04	0.01	0.02	0.04	0.005
M_2	0.0002	0.03	0.04	0.0001	0.04	0.04	0.01

(1.3)

$P(s \mid M_1) = 0.0000000000\ 0048$

$P(s \mid M_2) = 0.0000000000\ 00000384$

va $P(s|M1) > P(s|M2)$ ekanligini ko'ramiz. Bu yerda formulalarni ehtimollar mahsuloti nuqtai nazaridan keltiramiz, lekin ehtimollik ilovalarida keng tarqalganidek, amalda log. ehtimollar yig'indisi bilan ishlash eng yaxshisidir.

1.1.2. Til modellarining turlari

Atamalar ketma-ketligi bo'yicha ehtimollarni quramiz. Har doim (birinchi qismdagi 11.1) tenglamadagi zanjir qoidasidan foydalanib, hodisalar ketma-ketligi ehtimolini oldingi hodisalarga bog'liq bo'lgan har bir keyingi voqea ehtimoliga ajratish mumkin:

$$P(t_1t_2t_3t_4) = P(t_1)P(t_2 \mid t_1)P(t_3 \mid t_1t_2)P(t_4 \mid t_1t_2t_3) \qquad (1.4)$$

Til modelining eng oddiy shakli barcha shartli kontekstni yo'q qiladi va har bir atamani mustaqil ravishda baholaydi. Bunday model **unigram** tili modeli deb ataladi:

$$P_{uni}(t_1t_2t_3t_4) = P(t_1)P(t_2)P(t_3)P(t_4) \qquad (1.5)$$

Til modellarining ko'plab murakkab turlari mavjud, masalan, **bigramma** tili modellari

$$P_{bi}(t_1t_2t_3t_4) = P(t_1)P(t_2 \mid t_1)P(t_3 \mid t_2)P(t_4 \mid t_3) \qquad (1.6)$$

va undan ham murakkab grammatikaga asoslangan til modellari mavjud. Masalan, ehtimollik kontekstidan xoli grammatikalar. Bunday modellar **nutqni aniqlash**, **imloni to'g'rilash** va **mashina tarjimasi** kabi vazifalar uchun juda muhimdir. Bunda sizga atrofdagi kontekstga bog'liq bo'lgan atama ehtimoli kerak bo'ladi. Biroq, AQda tilni modellashtirish ishlarining aksariyati unigram tili modellaridan foydalangan. AQ sizga murakkab til modellari kerak bo'lgan joy emas, chunki AQ nutqni aniqlash kabi boshqa vazifalarni bajaradigan darajada jumlalar tuzilishiga bevosita bog'liq emas. **Unigram** modellari ko'pincha matn mavzusini baholash uchun yetarlidir. Bundan tashqari AQ tilining modellari ko'pincha bitta hujjatda baholanadi va shuning uchun samarali ishlashi uchun yetarli ma'lumotlari mavjudligi shubhalidir. Ma'lumotlarning siyrakligidan yo'qotishlar odatda boyroq modellardan olinadigan har qanday daromaddan ustun turadi. Bu qarama-qarshilik farqiga misol 3.6-bo'limda keltirilgan. Cheklangan o'quv ma'lumotlari bilan, yanada cheklangan model yaxshiroq ishlashga intiladi. Bundan tashqari, **unigram** modellari yuqori tartibli modellarga qaraganda baholash va

qo'llashda samaraliroqdir. Shunga qaramay, AQda iboralar va yaqinlik so'rovlarining ahamiyati, umuman olganda, kelajakdagi ishlarda murakkabroq til modellaridan foydalanish kerakligini ko'rsatadi. Ba'zilari esa 1.5-bo'limda berilgan. Darhaqiqat, bu harakat van Rijsbergenning modeliga mos keladi.

1.1.3. So'zlar bo'yicha ko'p nomli taqsimot

Unigram tili modelida so'zlarning tartibi ahamiyatsiz shuning uchun bunday modellar ko'pincha **"so'zlar sumkasi"** modellari deb ataladi. Oldingi kontekstda hech qanday shart mavjud bo'lmasa ham bu model baribir ma'lum bir tartibga solish ehtimolini bergan edi. Biroq, ushbu shartlar paketining boshqa har qanday tartibi bir xil ehtimolga ega bo'ladi. Shunday qilib, ularda so'zlar bo'yicha multinomial taqsimot mavjud. **Unigram** modellariga sodiq ekanmiz, til modeli nomi va motivatsiyasi zaruriy emas, balki tarixiy deb qaralishi mumkin. Buning o'rniga modelga multinomial model sifatida murojaat qilishimiz mumkin. Shu nuqtai nazardan qaraganda, yuqorida keltirilgan tenglamalar so'zlar sumkasining ko'p nomli ehtimolini ko'rsatmaydi. Chunki ko'p nomli koeffitsient so'zlarning barcha mumkin bo'lgan tartiblarini jamlamaydi. Multinomial modelning standart ko'rinishi quyidagicha:

$$P(d) = \frac{L_d!}{tf_{t_1,d}!\,tf_{t_2,d}!\,\ldots\,tf_{t_M,d}!} P(t_1)^{tf_{t_1,d}} P(t_2)^{tf_{t_2,d}} \ldots P(t_M)^{tf_{t_M,d}} \qquad (1.7)$$

$L_d = \sum_{i \in V} tf_{t_i,d}$ bu yerda **d** hujjatining uzunligi, **M** - atama lug'atining o'lchami va mahsulotlar hujjatdagi pozitsiyalardan emas balki lug'atdagi atamalardan oshib ketgan. Biroq, xuddi STOP ehtimollarida bo'lgani kabi, amalda ham hisob-kitoblarda ko'p nomli koeffitsientni qo'yib yuborish mumkin. Chunki ma'lum bir **so'z sumkasi** uchun u doimiy bo'ladi va shuning uchun u ikki xil ehtimollik nisbatiga ta'sir qilmaydi. Ma'lum bir so'z sumkasini yaratuvchi modellar mavjud. Ko'p nomli taqsimotlar 2.2-bo'limda ham ko'rsatilgan. Til modellarini loyihalashda asosiy muammo shundaki, M_d modeli sifatida aynan nimadan foydalanish kerakligini bilmaymiz. Ammo ularda odatda ushbu modelni ifodalovchi matn namunasi mavjud. Bu muammo til modellarining asl, birlamchi

qo'llanilishida juda ko'p ma'noga ega. Masalan, nutqni aniqlashda og'zaki matnning o'quv namunasi mavjud. Ammo kelajakda foydalanuvchilar ilgari hech qachon kuzatmagan turli xil so'zlarni turli ketma-ketlikda ishlatishadi va shuning uchun model noma'lum so'zlar va ketma-ketliklarga ruxsat berish uchun kuzatilgan ma'lumotlardan tashqarida umumlashtirishi kerak. Hujjat cheklangan va odatda o'zgarmas bo'lgan AQ holatida bu talqin unchalik aniq emas. AQda qabul qiladigan strategiya quyidagicha: **d** hujjati faqat namunaviy taqsimotdan olingan matnning namunaviy namunasi deb o'ylash va unga muhim mavzu sifatida qaraladi. Keyin ushbu namunadan til modeli baholanadi va har qanday so'z ketma-ketligini kuzatish ehtimolini hisoblash uchun ushbu modeldan foydalaniladi va nihoyat, hujjatlar so'rovni yaratish ehtimoli bo'yicha tartiblanadi.

1.2. So'rovlar ehtimoli modeli

1.2.1. AQ da so'rovlar ehtimolligi tili modellaridan foydalanish

Tilni modellashtirish AQga nisbatan juda umumiy rasmiy yondashuv bo'lib, ko'plab variantlarni amalga oshiradi. AQ da til modellaridan foydalanishning asl va asosiy usuli so'rovlar ehtimoli modelidir. Unda to'plamdagi har bir **d** hujjatdan M_d til modeli quriladi. Maqsad hujjatlarni P(d|q) bo'yicha tartiblashdir, bunda hujjatning ehtimolligi uning so'rovga mos kelishi ehtimoli sifatida talqin qilinadi. Bayes qoidasidan foydalanib ularda:

$$P(d \mid q) = P(q \mid d) P(d) / P(q)$$

P(q) barcha hujjatlar uchun bir xil, shuning uchun uni e'tiborsiz qoldirish mumkin. P(d) hujjatining oldingi ehtimoli ko'pincha barcha d bo'ylab bir xil deb hisoblanadi va shuning uchun ham uni e'tiborsiz qoldirish mumkin, lekin avtoritet, uzunlik, janr, yangilik va oldingi hujjatlar soni kabi mezonlarni o'z ichiga olishi mumkin bo'lgan haqiqiy amallarni amalga oshirish mumkin. Ammo, bu soddalashtirishlarni hisobga olgan holda, oddiygina P(q|d), d dan olingan til modeli ostida **q** so'rovining ehtimoli bo'yicha tartiblangan natijalar qaytariladi. Shunday qilib, Tilni Modellashtirish yondashuvi so'rovlarni yaratish jarayonini

modellashtirishga harakat qiladi. Hujjatlar so'rovning tegishli hujjat modelidan tasodifiy namuna sifatida kuzatilish ehtimoli bo'yicha tartiblanadi.

Buning eng keng tarqalgan usuli ko'p nomli unigram tili modelidan foydalanish bo'lib, u ko'p nomli **Naive Bayes** modeliga ekvivalentdir, bunda hujjatlar sinflar bo'lib, har biri alohida "til" sifatida ko'rib chiqiladi. Ushbu model ostida quyidagilar mavjud:

$$P(q|M_d) = K_q \prod_{w \in V} P(t|M_d)^{tf_{t,d}} \quad (1.8)$$

bu yerda yana $K_q = L_d !/(tf_{t1,d} !tf_{t2,d} ! \cdots tf_{tM,d}!)$ - q so'rovi uchun ko'p nomli koeffitsient, chunki u a alohida so'rov uchun o'zgarmasdir.

Til modeli (bundan buyon LM) asosida qidirish uchun so'rovlarni yaratishni tasodifiy jarayon sifatida ko'rib chiqiladi. Bunga yondashuvlar quyidagicha:

1. Har bir hujjat uchun LM xulosasini chiqarish.
2. $P(q|M_d)$, ushbu hujjat modellarining har biri bo'yicha so'rovni yaratish ehtimolini baholash.
3. Hujjatlarni shu ehtimolliklarga qarab tartiblash.

Asosiy modelning sezgisi shundan iboratki, foydalanuvchi prototip hujjatga ega bo'lib, ushbu hujjatda paydo bo'lgan so'zlar asosida so'rov hosil qiladi. Ko'pincha foydalanuvchilar qiziqtirgan hujjatlarda paydo bo'lishi mumkin bo'lgan atamalar haqida foydali tasavvurga ega bo'ladi va ushbu hujjatlarni to'plamdagi boshqalardan ajratib turadigan so'rov shartlarini tanlaydi. Yig'ish statistikasi ajralmas hisoblanadi. Ko'plab boshqa yondashuvlarda bo'lgani kabi evristik tarzda qo'llanilmaydi balki til modelining bir qismi sifatida qo'llaniladi.

1.2.2. So'rovlarni yaratish ehtimolini baholash

Ushbu bo'limda $P(q|M_d)$ ni qanday baholash tasvirlanadi. Maksimal ehtimollik smetasi (MLE) va unigramlik taxmindan foydalangan holda **d** hujjatining LM va Md berilgan so'rovini ishlab chiqarish ehtimoli hosil qilinadi

Albatta, boshqa hollarda bu mumkin emas. 1.4-bo'limda qisqacha muhokama qilinganidek, tilni modellashtirish yondashuvida bunga javob tarjima tili modellaridir.

$$\hat{P}(q|M_d) = \prod_{t \in q} \hat{P}_{mle}(t|M_d) = \prod_{t \in q} \frac{tf_{t,d}}{L_d}$$

Bu yerda $\mathbf{M_d}$ **d** hujjatning til modeli, $\mathbf{tf_{t,d}}$ **d** hujjatidagi **t** terminining atama chastotasi, $\mathbf{L_d}$ esa **d** hujjatidagi tokenlar soni. Ya'ni, har bir so'z qanchalik tez-tez sodir bo'lgani hisoblanadi va hujjatdagi so'zlarning umumiy soniga bo'linadi. Bu MLEni hisoblashning bir xil usulidir ammo endilikda so'zlarni hisoblashda multinomialdan foydalaniladi. Til modellaridan foydalanishning klassik muammosi bu baholashdir. Hujjatlarda atamalar juda kam uchraydi. Xususan, Ba'zi so'zlar hujjatda umuman ko'rinmaydi, lekin foydalanuvchi so'rovda foydalangan bo'lishi mumkin bo'lgan ma'lumotga ehtiyoj uchun mumkin bo'lgan so'zlardir. Agar **d** hujjatida yetishmayotgan atama uchun P(t| Md) = 0 ni taxmin qilsak, qat'iy kon'yunktiv semantikaga ega bo'lamiz. Hujjatlar faqat so'rovning barcha shartlari hujjatda paydo bo'lsa, so'rovga nolga teng bo'lmagan ehtimollik beradi. Nutqni aniqlash ilovasida keyingi so'zni bashorat qilish kabi til modellarining boshqa qo'llanishlarida nol ehtimollik muammosi aniq chunki o'quv ma'lumotlarida ko'p so'zlar kam ifodalanadi. Bu AQ ilovasida muammoli yoki yo'qligi unchalik aniq emasdek tuyulishi mumkin. Buni inson-kompyuter interfeysi muammosi deb hisoblash mumkin. Vektor kosmik tizimlari odatda yumshoqroq moslashishni afzal ko'radi ammo so'nggi veb-qidiruv ishlanmalari ko'proq bunday kon'yunktiv semantika bilan qidiruvlarni amalga oshirishga moyil bo'ldi. Bu yerda qanday yondashuvdan qat'i nazar, umumiyroq baholash muammosi mavjud. Sodir bo'lgan so'zlar ham yomon baholanadi. Xususan, hujjatda so'zlarning bir marta paydo bo'lish ehtimoli odatda, ortiqcha baholanadi. Chunki ularning bir marta sodir bo'lishi qisman tasodifan sodir bo'lgan. Ammo odamlar LM yondashuvini yaxshiroq tushunishgani sababli, ushbu modelda silliqlashning roli nafaqat nol ehtimollikdan qochish emasligi ma'lum bo'ldi. Terminlarni tekislash aslida atamaning og'irlik komponentining asosiy qismlarini amalga oshiradi. Shunchaki silliq bo'lmagan modelda kon'yunktiv semantika mavjud emas. Silliq bo'lmagan model yomon ishlaydi, chunki unda tortish komponenti atamasining qismlari yo'q. Shunday qilib, hujjat tili modellarimizda ehtimollarni tekislashimiz kerak. Nolga teng bo'lmagan ehtimolliklarni kamaytirish va ko'rinmas so'zlarga bir oz ehtimollik massasini berish kerak. Ushbu muammoni hal qilish uchun ehtimollik taqsimotini yumshatishga yondashuvlarning keng

doirasi mavjud. Kuzatilgan sonlarga raqam (1, 1/2 yoki kichik a) qo'shish va ehtimollik taqsimotini berish uchun qayta normallashtirishni muhokama qilish kerak bo'ladi.

Ushbu bo'limda yana bir nechta silliqlash usullarini eslatib o'tamiz. Bu jarayon kuzatilgan sonlarni umumiy mos yozuvlar ehtimoli taqsimoti bilan birlashtirishni o'z ichiga oladi. Umumiy yondashuv shundan iboratki, so'rovda uchramaydigan atama mumkin bo'lishi kerak, lekin uning ehtimoli bir oz yaqin bo'lishi kerak.. Ya'ni, agar $\mathbf{tf}_{t,d}= 0$ bo'lsa,

$$\hat{P}(t \mid M_d) \leq \epsilon f_t / T .$$

Bu yerda $\mathbf{cf}_t$ - to'plamdagi atamaning noaniq miqdori va **T** - butun to'plamning noaniq hajmi (tokenlar soni). Amalda yaxshi ishlaydigan oddiy g'oya hujjatga xos multinomial taqsimot va butun to'plamdan hisoblangan multinomial taqsimot o'rtasidagi aralashmani qo'llashdir.

$$\hat{P}(t \mid d) = \lambda \hat{P}_{mle}(t \mid M_d) + (1-\lambda)\hat{P}_{mle}(t \mid M_c) \qquad (1.10)$$

bu yerda $0<1<1$ va **Mc** butun hujjatlar to'plamidan tuzilgan til modelidir. Bu hujjatdagi ehtimollikni so'zning umumiy yig'ish chastotasi bilan aralashtirib yuboradi. Bunday model chiziqli interpolyatsiya tili modeli deb ataladi.

L ni to'g'ri sozlash ushbu modelning yaxshi ishlashi uchun muhimdir. Muqobil variant - Bayesian yangilash jarayonida oldingi taqsimot sifatida butun to'plamdan tuzilgan til modelidan foydalanishdir. Keyin quyidagi tenglamani olamiz:

$$\hat{P}(t \mid d) = \frac{tf_{t,d} + \alpha \hat{P}(t \mid M_c)}{L_d + \alpha} \qquad (1.11)$$

Ushbu silliqlash usullarining ikkalasi ham AQ tajribalarida yaxshi ishlashi ko'rsatilgan. Ushbu bo'limning qolgan qismida chiziqli interpolyatsiya silliqlash usuliga to'xtalamiz. Tafsilotlari jihatidan farq qilsa-da, ikkalasi ham kontseptual jihatdan o'xshashdir: Ikkala holatda ham hujjatda mavjud bo'lgan so'z uchun ehtimollik bahosi chegirmali MLE va hujjatda mavjud bo'lmagan so'zlar uchun uning butun to'plamda tarqalishi bahosining bir qismini birlashtiradi. Smeta butun to'plamda so'zning tarqalishini baholashning faqat bir qismidir. AQ uchun LMlarda silliqlashning roli shunchaki yoki asosan baholash muammolarini oldini

olish uchun emas. Modellar birinchi marta taklif qilinganda bu aniq emas edi, ammo endi modellarning yaxshi xususiyatlari uchun silliqlash muhim ahamiyatga ega ekanligi aniqlandi.

Bu modellarda silliqlash l va a parametrlari bilan boshqariladi: l ning kichik qiymati yoki a ning katta qiymati ko'proq tekislashni bildiradi. Ushbu parametr chiziqli qidiruv yordamida ishlashni optimallashtirish uchun sozlanishi mumkin (yoki chiziqli interpolyatsiya modeli uchun, kutishni maksimallashtirish algoritmi kabi boshqa usullar bilan; 16.5-bo'lim, 368-betga qarang). Qiymat doimiy bo'lishi shart emas. Yondashuvlardan biri qiymatni so'rov hajmining funksiyasiga aylantirishdir. Bu foydali, chunki qisqa so'rovlar uchun kichik miqdordagi silliqlash ("kon'yunktivga o'xshash" qidiruv) ko'proq mos keladi, ko'p silliqlash esa uzoq so'rovlar uchun ko'proq mos keladi. Xulosa qilib aytadigan bo'lsak, ko'rib chiqayotgan AQ uchun asosiy LM ostida **q** so'rovi uchun qidiruv reytingi quyidagicha berilgan:

$$P(d \mid q) \propto P(d) \prod_{t \in q} ((1-\lambda)P(t \mid M_c) + \lambda P(t \mid M_d)) \qquad (1.12)$$

Bu tenglama foydalanuvchi ko'zlagan hujjatning haqiqatda **d** bo'lishi ehtimolini aks ettiradi.

Misol. Hujjatlar to'plamida ikkita hujjat bor deylik:

d_1: Xyzzy foyda haqida xabar beradi, lekin daromad pasayadi;

d_2: Quorus chorakdagi yo'qotishlarni qisqartiradi, ammo daromad yanada kamayadi.

Model hujjatlar va to'plamdagi MLE unigram modellari bo'ladi, l = 1/2 bilan aralashtiriladi. Aytaylik, so'rovda daromad kamaygan. Keyin:

$$\begin{aligned} P(q \mid d_1) &= [(1/8 + 2/16)/2] \times [(1/8 + 1/16)/2] \\ &= 1/8 \times 3/32 = 3/256 \\ P(q \mid d_2) &= [(1/8 + 2/16)/2] \times [(0/8 + 1/16)/2] \\ &= 1/8 \times 1/32 = 1/256 \end{aligned} \qquad (1.13).$$

1.2.3. Ponte va Kroft tajribalari

Ponte va Croft (1998) ma'lumotni qidirishda tilni modellashtirish yondashuvi bo'yicha birinchi tajribalarini taqdim etadi. Ularning asosiy yondashuvi hozirgacha taqdim etgan modeldir. Biroq, Ponte va Kroftning

ko'p o'zgaruvchan Bernulli modelidan ko'ra (Miller va boshq. 1999, Hiemstra 2000) til modeli ikkita multinomlarning aralashmasi bo'lgan yondashuvni taqdim etdi. LM yondashuvidagi keyingi ishlarning aksariyatida multinomiyalardan foydalanish standart bo'lib kelgan va AQda eksperimental natijalar, shuningdek, 2.3-bo'limda ko'rib chiqadigan matn tasnifidan olingan dalillar uning ustun ekanligini ko'rsatadi.

	Precision			
Rec.	tf-idf	LM	%chg	
0.0	0.7439	0.7590	+2.0	
0.1	0.4521	0.4910	+8.6	
0.2	0.3514	0.4045	+15.1	*
0.3	0.2761	0.3342	+21.0	*
0.4	0.2093	0.2572	+22.9	*
0.5	0.1558	0.2061	+32.3	*
0.6	0.1024	0.1405	+37.1	*
0.7	0.0451	0.0760	+68.7	*
0.8	0.0160	0.0432	+169.6	*
0.9	0.0033	0.0063	+89.3	
1.0	0.0028	0.0050	+76.9	
Ave	0.1868	0.2233	+19.55	*

1.4-rasm. **Ponte va Croft (1998) tomonidan tf-idf ni tilni modellashtirish (LM) atamalarining vazni bilan solishtirish natijalari**

INQUERY AQ tizimidagi **tf-idf** versiyasi **tf** uzunligini normallashtirishni o'z ichiga oladi. Jadvalda Wilcoxon tomonidan imzolangan darajali test bo'yicha * belgisi bilan belgilangan 11 ballli o'rtacha aniqlik bo'yicha baho berilgan.

Tilni modellashtirish yondashuvi bu tajribalarda har doim yaxshi natija beradi lekin shuni yodda tutingki, bu yondashuv sezilarli yutuqlarni ko'rsatsagina eslab qolishning yuqori darajalariga erishiladi.

Ponte va Kroft an'anaviy **tf-idf** og'irliklariga nisbatan tilni modellashtirish yondashuvidan kelib chiqadigan og'irliklar atamasi samaradorligini qat'iy bahslashdi. Ularning natijalarini bir qismi 1.4-rasmda taqdim etilgan, bu yerda TREC 202-250 mavzularini TREC 2 va 3 disklari orqali baholashni **tf-idf** ni til modellashtirish bilan solishtiriladi. So'rovlar jumla uzunligidagi tabiiy til so'rovlaridir. Tilni modellashtirish yondashuvi **tf-idf** asosidagi atamalarni baholash yondashuviga qaraganda

ancha yaxshi natijalar beradi. Haqiqatan ham bu yerda ko'rsatilgan yutuqlar keyingi ishlarda kengaytiriladi.

1.3. AQ da boshqa yondashuvlarga nisbatan tilni modellashtirish

Tilni modellashtirish yondashuvi matnni qidirish muammosiga qarashning yangi usulini taqdim etadi, bu uni nutq va tilni qayta ishlash bo'yicha ko'plab so'nggi ishlar bilan bog'laydi. Ponte va Croft (1998) ta'kidlaganidek, AQ uchun tilni modellashtirish yondashuvi so'rovlar va hujjatlar o'rtasidagi mosliklarni baholashga boshqacha yondashuvni ta'minlaydi va umid tilni modellashtirishning ehtimollik asosi qo'llaniladigan og'irliklarni yaxshilaydi va shuning uchun dastur samaradorligini oshiradi. Asosiy masala - hujjat modelini baholash masalan, uni qanday qilib samarali yumshatishni tanlash. Model juda yaxshi qidirish natijalariga erishdi. 1- qismning 11-bobdagi BIM kabi boshqa ehtimoliy yondashuvlar bilan solishtirganda dastlab asosiy farq shundaki, LM yondashuvi aniq modellashtirish ahamiyatini yo'qotadi (holbuki bu BIM yondashuvida baholanadigan markaziy o'zgaruvchidir). Ammo bu narsalar haqida fikr yuritishning to'g'ri yo'li bo'lmasligi mumkin chunki 1.5-bo'limda batafsil muhokama qilinadi. LM yondashuvi hujjatlar va axborot ehtiyojlarini ifodalash bir xil turdagi obyektlar ekanligini nazarda tutadi va nutq va tabiiy tilni qayta ishlashdan tilni modellashtirish vositalari va usullarini import qilish orqali ularning mosligini baholaydi.

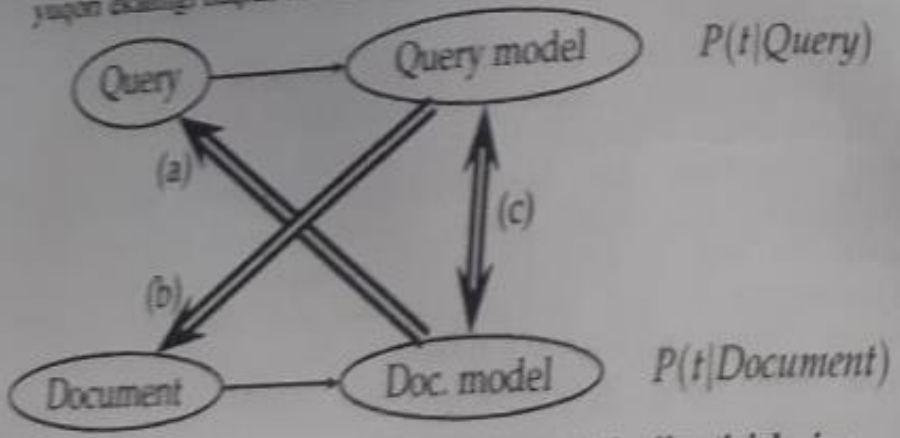
Olingan model matematik jihatdan aniq, kontseptual jihatdan sodda, hisoblash uchun qulay va intuitiv jihatdan jozibali. Bu XML qidirish bilan bog'liq vaziyatga o'xshaydi: U yerda so'rovlar va hujjatlarni bir xil turdagi obyektlar deb hisoblaydigan yondashuvlar ham eng muvaffaqiyatli hisoblanadi. Boshqa tomondan, barcha AQ modellari singari, siz ham modelga e'tiroz bildirishingiz mumkin. Hujjat va ma'lumotlarga bo'lgan ehtiyojni ifodalash o'rtasidagi ekvivalentlik haqidagi taxmin haqiqatga to'g'ri kelmaydi. Hozirgi LM yondashuvlari tilning juda oddiy modellaridan, odatda unigramma modellaridan foydalanadi.

Muvofiqlikning aniq tushunchasi bo'lmasa, tegishli fikr-mulohazalarni modelga integratsiya qilish qiyin, chunki foydalanuvchining afzalliklari. Bundan tashqari, iboralar yoki parchalarni

moslashtirish yoki mantiqiy qidiruv operatorlari tushunchalarini joylashtirish uchun unigram modelidan tashqariga o'tish kerak. LM yondashuvidagi keyingi ishlar ushbu xavotirlarning Ba'zilarini jumladan, modelga tegishlilikni qaytarish va so'rovlar tili va hujjat tili o'rtasidagi til nomuvofiqligiga yo'l qo'yishni ko'rib chiqdi.

Model an'anaviy **tf-idf** modellari bilan muhim aloqalarga ega. Termin chastotasi to'g'ridan-to'g'ri **tf-idf** modellarida ifodalanadi va ko'plab so'nggi ishlar hujjat uzunligini normallashtirish muhimligini tan oldi. Hujjatlarni yaratish ehtimolini yig'ish ehtimoli bilan aralashtirishning ta'siri biroz **idf** ga o'xshaydi. Umumiy to'plamda kam uchraydigan, lekin Ba'zi hujjatlarda keng tarqalgan atamalar hujjatlar reytingiga ko'proq ta'sir qiladi. Ko'pgina aniq amalga oshirishda modellar atamalarni mustaqil bo'lgandek ko'rib chiqadi. Boshqa tomondan, segular geometrik emas, balki ehtimollikdir, matematik modellar evristik emas, balki printsipialdir va atama chastotasi va hujjat uzunligi kabi statistik ma'lumotlardan qanday foydalanish tafsilotlari farqlanadi.

Agar asosan ishlash ko'rsatkichlari bilan bog'liq bo'lsangiz, yaqinda olib borilgan ishlar LM yondashuvining tf-idf va BM25 og'irliklarini yengib o'tish tajribalarida juda samarali ekanligini ko'rsatdi. Shunga qaramay, mavjud dasturni o'zgartirishni oqlash uchun uning ishlashi yaxshi sozlangan an'anaviy vektor makonini qidirish tizimidan ancha yuqori ekanligi haqida hali ham dalillar yetarli emas.



1.5-rasm. **Tilni modellashtirish yondashuvini rivojlantirishning uchta usuli**

(a) so'rovlar ehtimoli, (b) hujjatning ehtimolligi va (c) modelni taqqoslash

1.4. Tilni modellashtirishning kengaytirilgan yondashuvlari

Ushbu bo'limda tilni modellashtirishning asosiy yondashuvini kengaytiradigan Ba'zi ishlarni qisqacha eslatib o'tamiz.

AQ sozlamalarida tilni modellashtirish g'oyasidan foydalanishni o'ylashning boshqa usullari mavjud va ularning ko'pchiligi ishlarda sinab ko'rilgan. Md hujjat tili modelining so'rovni yaratish ehtimolini ko'rib chiqish o'rniga, Mq so'rov tili modelining hujjatni yaratish ehtimolini ko'rib chiqish mumkin. Ushbu yo'nalishda ishlarni bajarish va hujjatning ehtimollik modelini yaratish unchalik jozibador emasligining asosiy sababi, so'rov matni asosida til modelini baholash uchun kamroq matn mavjud va shuning uchun model yomonroq baholanadi va kerak bo'ladi. Ko'proq boshqa til modeli bilan silliqlanilishiga bog'liq. Boshqa tomondan, bunday modelga tegishli fikr-mulohazalarni qanday kiritish mumkinligini ko'rish oson: so'rovni tegishli hujjatlardan olingan atamalar bilan odatiy tarzda kengaytirishingiz va shuning uchun Mq til modelini yangilashingiz mumkin (Zhai and Lafferty 2001a). Haqiqatan ham, mos modellashtirish tanlovlari bilan bu yondashuv BIM modeliga olib keladi. Lavrenko va Croft (2001) ning dolzarblik modeli hujjatning ehtimollik modelining namunasi bo'lib, soxta moslik bilan bog'liq fikr-mulohazalarni tilni modellashtirish yondashuviga kiritadi. Bu juda kuchli empirik natijalarga olib keladi.

To'g'ridan-to'g'ri ikkala yo'nalishni yaratish o'rniga, hujjat va so'rovdan til modelini yaratish mumkin va keyin bu ikki til modeli bir-biridan qanchalik farq qilishini so'rash mumkin. Lafferty va Zhai (2001) 1.5-rasmda ko'rsatilgan muammo haqida fikr yuritishning ushbu uchta usulini ishlab chiqdi va hujjatlarni qidirishda xavflarni minimallashtirishning umumiy yondashuvini ishlab chiqdi. Masalan, **q** so'roviga tegishli bo'lgan **d** hujjatini qaytarish xavfini modellashtirishning bir usuli bu ularning tegishli til modellari orasidagi Kullback-Leibler (KL) farqidan foydalanishdir:

$$R(d;q) = KL(M_d \| M_q) = \sum_{t \in V} P(t \mid M_q) \log \frac{P(t \mid M_q)}{P(t \mid M_d)} \qquad (1.14).$$

KL divergensiyasi axborot nazariyasidan kelib chiqqan assimetrik tabaqalanish o'lchovidir, u Md modellashtirishda Mq ehtimollik taqsimoti qanchalik yomonligini o'lchaydi (Cover va Tomas 1991, Manning va

Schütze 1999). Lafferty va Zhai (2001) namunaviy taqqoslash yondashuvi so'rovlar ehtimoli va hujjatning ehtimollik yondashuvlaridan ustun ekanligini ko'rsatadigan natijalarni taqdim etadi. KL divergensiyasidan reyting funksiyasi sifatida foydalanishning kamchiliklaridan biri shundaki, ballarni so'rovlar bo'yicha taqqoslab bo'lmaydi. Bu maxsus qidiruv uchun muhim emas, lekin mavzuni kuzatish kabi boshqa ilovalarda muhim ahamiyatga ega. Kraaij va Spitters (2003) o'xshashlikni normallashtirilgan log-ehtimollik nisbati (yoki teng ravishda, o'zaro entropiyalar orasidagi farq) sifatida modellashtiradigan muqobil taklifni taklif qiladi.

Asosiy LMlar muqobil ifoda, ya'ni sinonimiya yoki so'rovlar va hujjatlar o'rtasidagi tildan foydalanishdagi har qanday og'ish masalalarini hal qilmaydi. Berger va Lafferty (1999) ushbu so'rov-hujjat bo'shlig'ini ko'paytirish uchun tarjima modellarini taqdim etadilar. Tarjima modeli sizga hujjatda bo'lmagan so'rov so'zlarini o'xshash ma'noli muqobil atamalarga tarjima qilish orqali yaratish imkonini beradi. Bu shuningdek, tillararo IRni amalga oshirish uchun asos yaratadi. Tarjima modelini lug'at terminlari orasidagi $T(\cdot|\cdot)$ shartli ehtimollik taqsimoti bilan ifodalash mumkin, deb faraz qiladilar. Tarjima so'rovlarini yaratish modelining shakli quyidagicha:

$$P(q \mid M_d) = \prod_{t \in q} \sum_{v \in V} P(v \mid M_d) T(t \mid v) \qquad (1.15)$$

P(v| Md) atamasi hujjat tilining asosiy modeli bo'lib, T(t|v) atamasi tarjimani amalga oshiradi. Ushbu model aniqroq hisoblashni talab qiladi va ular tarjima modelini yaratishimiz kerak. Tarjima modeli odatda alohida resurslar (masalan, an'anaviy tezaurus yoki ikki tilli lug'at yoki statistik mashina tarjimasi tizimining tarjima lug'ati) yordamida tuziladi, lekin agar matnning boshqa qismlarini tabiiy ravishda ifodalovchi yoki umumlashtiradigan matn bo'laklari mavjud bo'lsa, hujjatlar to'plamidan foydalangan holda matn tuzilishi mumkin. Hujjatlar va ularning sarlavhalari yoki tezislari yoki gipermatn muhitida ularga ishora qiluvchi hujjatlar va langar-matn nomzodlik namunalaridir. Kengaytirilgan LM yondashuvlarini yaratish faol tadqiqot sohasi bo'lib qolmoqda. Umuman olganda, tarjima modellari, tegishli fikr-mulohaza modellari va modellarni taqqoslash yondashuvlari asosiy so'rovlar ehtimolligi LM bo'yicha ishlashni yaxshilash uchun namoyish etilgan.

Cleverdon, Cyril W.
1991.
The significance of the Cranfield tests on index languages.
In *Proc. SIGIR*, pp. 3-12. ACM Press.
Coden, Anni R., Eric W. Brown, and Savitha Srinivasaneds.).
2002.
Information Retrieval Techniques for Speech Applications.
Springer.
Cohen, Paul R.
1995.
Empirical methods for artificial intelligence.
MIT Press.
Cohen, William W.
1998.
Integration of heterogeneous databases without common domains using queries based on textual similarity.
In *Proc. SIGMOD*, pp. 201-212. ACM Press.
Cohen, William W., Robert E. Schapire, and Yoram Singer.
1998.
Learning to order things.
In *Proc. NIPS*. The MIT Press.
URL: *citeseer.ist.psu.edu/article/cohen98learning.html*.
Cohen, William W., and Yoram Singer.
1999.
Context-sensitive learning methods for text categorization.
TOIS 17 (2): 141-173.
Liddy, Elizabeth D.
2005.
Automatic document retrieval.
In *Encyclopedia of Language and Linguistics*, 2nd edition. Elsevier.
List, Johan, Vojkan Mihajlovic, Georgina Ramírez, Arjen P. Vries, Djoerd Hiemstra, and Henk Ernst Blok.
2005.
TIJAH: Embracing IR methods in XML databases.

IR 8 (4): 547-570.
DOI: *dx.doi.org/10.1007/s10791-005-0747-2.*

Lita, Lucian Vlad, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla.
2003.
tRuEcasIng.
In *Proc. ACL*, pp. 152-159.

Littman, Michael L., Susan T. Dumais, and Thomas K. Landauer.
1998.
Automatic cross-language information retrieval using latent semantic indexing.
In Gregory Grefenstetteed.), *Proc. Cross-Language Information Retrieval*. Kluwer.
URL: *citeseer.ist.psu.edu/littman98automatic.html*.

Liu, Tie-Yan, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma.
2005.
Support vector machines classification with very large scale taxonomy.
ACM SIGKDD Explorations 7 (1): 36-43.

Liu, Xiaoyong, and W. Bruce Croft.
2004.
Cluster-based retrieval using language models.
In *Proc. SIGIR*, pp. 186-193. ACM Press.
DOI: *doi.acm.org/10.1145/1008992.1009026*.

Mooers, Calvin E.
1950.
Coding, information retrieval, and the rapid selector.
American Documentation 1 (4): 225-229.

Moschitti, Alessandro.
2003.
A study on optimal parameter tuning for Rocchio text classifier.
In *Proc. ECIR*, pp. 420-435.

Moschitti, Alessandro, and Roberto Basili.
2004.
Complex linguistic features for text classification: A comprehensive study.
In *Proc. ECIR*, pp. 181-196.

Murata, Masaki, Qing Ma, Kiyotaka Uchimoto, Hiromi Ozaku, Masao Utiyama, and Hitoshi Isahara.
2000.
Japanese probabilistic information retrieval using location and category information.
In *International Workshop on Information Retrieval With Asian Languages*, pp. 81-88.
URL: *portal.acm.org/citation.cfm?doid=355214.355226*.

Muresan, Gheorghe, and David J. Harper.
2004.
Topic modeling for mediated access to very large document collections.
JASIST 55 (10): 892-910.
DOI: *dx.doi.org/10.1002/asi.20034*.

Murtagh, Fionn.
1983.
A survey of recent advances in hierarchical clustering algorithms.
Computer Journal 26 (4): 354-359.

Lee, Whay C., and Edward A. Fox.
1988.
Experimental comparison of schemes for interpreting Boolean queries.
Technical Report TR-88-27, Computer Science, Virginia Polytechnic Institute and State University.

Lempel, Ronny, and Shlomo Moran.
2000.
The stochastic approach for link-structure analysis (SALSA) and the TKC effect.
Computer Networks 33 (1-6): 387-401.
URL: *citeseer.ist.psu.edu/lempel00stochastic.html*.

Lesk, Michael.
1988.
Grab - Inverted indexes with low storage overhead.
Computing Systems 1: 207-220.

Lesk, Michael.
2004.
Understanding Digital Libraries, 2nd edition.
Morgan Kaufmann.

1- bob bo'yicha nazariy va amaliy test savollari

1. Indekslash quyidagi qaysi jarayonlardan iborat bo'lib amalga oshiriladi?
   A) Mavzular, tizimlashtirish va koordinatsion indekslash
   B) Faqat mavzular
   C) Faqat tizimlashtirish
   D) To'g'ri javob yo'q

2. Tizimalashtirish – bu nim?
   A) Hujjat va (yoki (yoki) talabi iPI (COG) qoidalariga muvofiq indekslash turi, IPI (COG). Indekslashning ushbu printsipi tasniflanadi. Bu ierarxik asosda ma'lumotlarni qidirishni tashkil etish imkoniyatini beradi. Kutubxonalar va axborot markazlarida tizimlashtirish jarayonini amalga oshirish jarayonini muntazam deb ataladi
   B) Indekslashning ob'ektiv printsipi alifbo tartibida ma'lumotlarni olishni tashkil etish imkoniyatini beradi. Mavzular jarayonlari kutubxonalar va axborot markazlarida o'tkaziladi
   C) Hujjatning semantik tarkibi turli xil kalit so'zlar yoki ta'riflar yordamida indekslash turi
   D) To'g'ri javob yo'q

3. Quyidagilarning qaysi biri ierarxik asosda ma'lumotlarni qidirishni tashkil etish imkoniyatini beradi?
   A) Tizimlashtirish
   B) Mavzular
   C) Koorsinatsion indekslash
   D) Barchasi to'g'ri

4. Mavzu - indekslash turini belgilang.
   A) Indekslashning ob'ektiv printsipi alifbo tartibida ma'lumotlarni olishni tashkil etish imkoniyatini beradi. Mavzular jarayonlari kutubxonalar va axborot markazlarida o'tkaziladi
   B) Hujjat qoidalariga muvofiq indekslash turi, IPI (COG) va indekslashning ushbu printsipi tasniflanadi. Bu ierarxik asosda ma'lumotlarni qidirishni tashkil etish imkoniyatini beradi. Kutubxonalar va axborot markazlarida tizimlashtirish jarayonini amalga oshirish jarayonini muntazam deb ataladi
   C) Hujjatning semantik tarkibi turli xil kalit so'zlar yoki ta'riflar yordamida indekslash turi

   D) To'g'ri javob yo'q

5. Indekslash turlarini toping?
   A) Mavzular, tizimlashtirish va koordinatsion indekslash
   B) Mavzular, tizimlashtirish
   C) Tizimlashtirish
   D) Koordinatsion

6. Quyidagi qaysi jarayonlar kutubxonalar va axborot markazlarida o'tkaziladi?
   A) Mavzular
   B) Tizimlashtirish
   C) Koorsinatsion indekslash
   D) Barchasi to'g'ri

7. Koordinatsion indeksing – bu nima?
   A) Hujjatning semantik tarkibi turli xil kalit so'zlar yoki ta'riflar yordamida indekslash turi. Koordinatsion indeksatsiyalarni amalga oshiradigan mutaxassislar indekslarni indekslar deb ataladi
   B) Hujjat yoki so'rovning matnidagi so'z yoki ibora, unda muhim semantik yukni olib boradi
   C) Faqat rasmiy protseduralarni hisoblash uchun faqat rasmiy protseduralar yordamida hisoblash orqali amalga oshiriladigan hujjat yoki so'rovni qayta ishlash uchun
   D) To'g'ri javob yo'q

8. Deskriptor tili nima?
   A) Koordinata indeksing uchun mo'ljallangan IPI
   B) Tizimlashtirish
   C) Koorsinatsion indekslash
   D) To'g'ri javob yo'q

9. Avtomatlashtirilgan indekslash – bu nima?
   A) Texnologiyalar hisoblash texnologiyasi yordamida amalga oshirilayotgan rasmiy protseduralardan foydalanishni ko'zda tutadigan va qidiruv imidjini tuzish bo'yicha asosiy qarorlar qabul qilishda intellektual protseduralardan foydalanishni o'z ichiga olishi mumkin
   B) Faqat rasmiy protseduralarni hisoblash uchun faqat rasmiy protseduralar yordamida hisoblash orqali amalga oshiriladigan hujjat yoki so'rovni qayta ishlash uchun
   C) Hujjat yoki so'rovning matnidagi so'z yoki ibora, unda muhim semantik yukni olib boradi
   D) To'g'ri javob yo'q

10. Texnologiyalar hisoblash texnologiyasi yordamida amalga oshirilayotgan rasmiy protseduralardan foydalanishni ko'zda tutadigan va qidiruv imidjini tuzish bo'yicha asosiy qarorlar qabul qilishda intellektual protseduralardan foydalanishni o'z ichiga olishi mumkin bo'lgan jarayonni belgilang.

A) Avtomatlashtirilgan indekslash
B) Avtomatik indeksatsiya
C) Informatsion so'z
D) To'g'ri javob yo'q

II BOB. MATN TASNIFI VA NAIVE BAYES MODELI

Hozirgacha ushbu kitob asosan maxsus qidiruv jarayonini muhokama qildi, bunda foydalanuvchilarning vaqtinchalik axborot ehtiyojlari mavjud bo'lib, ular qidiruv tizimiga bir yoki bir nechta so'rovlar yuborish orqali hal qilishga harakat qilishadi. Biroq, ko'p foydalanuvchilarning doimiy ma'lumotga ehtiyoji bor. Masalan, ko'p yadroli kompyuter chiplaridagi ishlanmalarni kuzatishingiz kerak bo'lishi mumkin. Buning bir usuli - har kuni ertalab so'nggi newswire maqolalari indeksiga ko'p yadroli AND kompyuter va chip so'rovini berishdir. Keyingi ikki bobda ushbu so'rovlar ko'rib chiqiladi.

Bu takrorlanuvchi vazifani qanday avtomatlashtirish mumkin? Shu maqsadda ko'pgina tizimlar doimiy so'rovlarni qo'llab-quvvatlaydi. Doimiy so'rov boshqa har qanday so'rovga o'xshaydi, bundan tashqari u vaqti-vaqti bilan yangi hujjatlar vaqt o'tishi bilan qo'shiladigan to'plamdan qidiriladi. Agar sizning doimiy so'rovingiz shunchaki ko'p yadroli AND kompyuter va chip bo'lsa, siz ko'p yadroli protsessorlar kabi boshqa atamalar ishlatadigan ko'plab tegishli yangi maqolalarni o'tkazib yuborasiz. Yaxshi eslab qolish uchun doimiy so'rovlar vaqt o'tishi bilan aniqlanishi kerak va asta-sekin murakkablashishi mumkin. Ushbu misolda, mantiqiy qidiruv tizimidan foydalanib, (ko'p yadroli AND chip OR protsessor OR mikroprotsessor) kabi so'rovga ega bo'lishingiz mumkin. Doimiy so'rovlar tegishli bo'lgan muammo maydonining umumiyligi va ko'lamini olish uchun endi tasniflash muammosining umumiy tushunchasini kiritamiz. Sinflar to'plamini hisobga olgan holda, berilgan obyekt qaysi sinf(lar)ga tegishli ekanligini aniqlashga harakat qilinadi. Misolda, doimiy so'rov yangi yangiliklar maqolalarini ikkita sinfga bo'lish uchun xizmat qiladi: ko'p yadroli kompyuter chiplari haqidagi hujjatlar va ko'p yadroli kompyuter chiplari haqidagi hujjatlar. Buni ikki toifali tasnif deb ataymiz. Doimiy so'rovlar yordamida tasniflash marshrutlash yoki filtrlash deb ham ataladi va 4.3.1-bo'limda batafsil muhokama qilinadi. Sinf doimiy so'rovlari ko'p yadroli kompyuter chiplari kabi tor yo'naltirilgan bo'lishi shart emas. Ko'pincha sinf Xitoy yoki qahva kabi umumiyroq mavzudir. Bunday ko'proq umumiy sinflar odatda mavzular deb ataladi va tasniflash vazifasi keyin matn tasnifi, matnlarni turkumlash, mavzularni tasniflash yoki mavzuni aniqlash deb ataladi. Xitoyga misol 2.1-rasmda keltirilgan. Doimiy so'rovlar va mavzular o'ziga xoslik darajasida farqlanadi, lekin

marshrutlash, filtrlash va matn tasnifini hal qilish usullari asosan bir xil. Shuning uchun ushbu va keyingi boblarda matn tasnifi rubrikasi ostida marshrutlash va filtrlashni o'z ichiga oladi.

Tasniflash tushunchasi juda umumiy bo'lib, axborot qidirish (AQ) doirasida va undan tashqarida ko'plab qo'llanmalarga ega. Masalan, kompyuterni ko'rishda klassifikator tasvirlarni landshaft, portret va boshqa sinflarga bo'lish uchun ishlatilishi mumkin. Bu yerda ma'lumot olish misollariga e'tibor qaratish kerak, masalan:

• indekslash uchun zarur bo'lgan bir qancha dastlabki ishlov berish bosqichlari: hujjatning kodlanishini aniqlash (ASCII, Unicode UTF-8 va boshqalar), so'z segmentatsiyasi (Ikki harf orasidagi bo'shliq so'z chegarasimi yoki yo'qmi?), truecasing va hujjatning tilini aniqlash;

• Spam-sahifalarni avtomatik aniqlash (keyinchalik qidiruv tizimi indeksiga kiritilmaydi);

• Jinsiy mazmundagi kontentni avtomatik aniqlash (foydalanuvchi Xavfsiz qidiruv kabi opsiyani o'chirib qo'ygan taqdirdagina qidiruv natijalariga kiritiladi);

• His-tuyg'ularni aniqlash yoki film yoki mahsulot sharhini ijobiy yoki salbiy deb avtomatik tasniflash. Misol ilovasi - foydalanuvchi kamerani sotib olishdan oldin uning nomaqbul xususiyatlari yoki sifat muammosi yo'qligiga ishonch hosil qilish uchun salbiy sharhlarni qidiradi;

• Shaxsiy elektron pochtani saralash. Foydalanuvchi suhbat e'lonlari, elektron hisob-kitoblar, oila va do'stlarning elektron pochta xabarlari va boshqalar kabi papkalarga ega bo'lishi mumkin va klassifikator har bir kiruvchi elektron pochtani tasniflashni va uni avtomatik ravishda tegishli jildga ko'chirishni xohlashi mumkin. Xabarlarni saralangan papkalarda topish juda katta kirish qutisiga qaraganda osonroq. Ushbu ilovaning eng keng tarqalgan holati barcha shubhali spam xabarlarni saqlaydigan spam papkasidir;

• Mavzuga oid yoki vertikal qidiruv. Vertikal qidiruv tizimlari ma'lum bir mavzu bo'yicha qidiruvlarni cheklaydi. Masalan, Xitoy mavzusi bo'yicha vertikal qidiruv tizimidagi informatika so'rovi Xitoy kompyuter fanlari bo'limlari ro'yxatini umumiy maqsadli qidiruv tizimidagi Xitoy informatika so'roviga qaraganda yuqori aniqlik va eslab qolish bilan qaytaradi. Buning sababi shundaki, vertikal qidiruv tizimi o'z indeksiga chinni atamasini boshqa ma'noda o'z ichiga olgan veb-sahifalarni o'z ichiga olmaydi (masalan, qattiq oq keramika haqida), lekin

Xitoy atamasini aniq eslatib o'tmagan bo'lsa ham, tegishli sahifalarni o'z ichiga oladi;

• Nihoyat, maxsus ma'lumotni qidirishda tartiblash funksiyasi ham hujjat tasniflagichiga asoslanishi mumkin, chunki ular 4.4-bo'limda tushuntiriladi.

Ushbu ro'yxat AQda tasniflashning umumiy ahamiyatini ko'rsatadi. Ko'pgina qidiruv tizimlari bugungi kunda klassifikatorning qandaydir shakllaridan foydalanadigan bir nechta komponentlarni o'z ichiga oladi. Ushbu kitobda misol sifatida ishlatadigan tasniflash vazifasi matn tasnifidir. Tasniflash uchun kompyuter muhim emas. Ko'pgina tasniflash vazifalari an'anaviy ravishda qo'lda hal qilingan. Kutubxonadagi kitoblarga kutubxonachi tomonidan Kongress kutubxonasi toifalari beriladi. Ammo qo'lda tasniflash qimmatga tushadi. Ko'p yadroli kompyuter chiplari misoli bir muqobil yondashuvni ko'rsatadi. Doimiy so'rovlardan foydalanish orqali tasniflash - qoida sifatida ko'rib chiqilishi mumkin - ko'pincha qo'lda yoziladi. Ularning misolimizda bo'lgani kabi (ko'p yadroli YOKI ko'p yadroli) VA (chip YOKI protsessor YOKI mikroprotsessor), qoidalar ba'zan mantiqiy ifodalarga ekvivalent bo'ladi. Qoida sinfni ko'rsatadigan kalit so'zlarning ma'lum birikmasini qamrab oladi. Qo'lda kodlangan qoidalar yaxshi masshtablash xususiyatlariga ega, ammo vaqt o'tishi bilan ularni yaratish va saqlash juda ko'p mehnat talab qiladi. Texnik jihatdan malakali shaxs (masalan, muntazam iboralarni yozishni yaxshi biladigan domen mutaxassisi) qisqa vaqt ichida muhokama qiladigan avtomatik tarzda yaratilgan tasniflagichlarning aniqligi bilan raqobatlashadigan yoki undan yuqori bo'lgan qoidalar to'plamini yaratishi mumkin. Ammo, bu maxsus mahoratga ega bo'lgan odamni topish qiyin bo'lishi mumkin. Qo'lda tasniflash va qo'lda tayyorlangan qoidalardan tashqari, matn tasnifiga uchinchi yondashuv mavjud, ya'ni mashinani o'rganishga asoslangan matn tasnifi. Bu keyingi bir necha boblarda e'tibor qaratadigan yondashuvdir. Mashinani o'rganishda qoidalar to'plami yoki umuman olganda, matn tasniflagichining qaror mezoni o'quv ma'lumotlaridan avtomatik ravishda o'rganiladi. Agar o'rganish usuli statistik bo'lsa, bu yondashuv statistik matn tasnifi deb ham ataladi.

Statistik matnlarni tasniflashda har bir sinf uchun bir qator yaxshi namunali hujjatlarni (o'quv hujjatlarini) talab qiladi. Qo'lda tasniflash zarurati yo'qolmaydi, chunki o'quv hujjatlari ularni belgilagan shaxsdan keladi - bu yerda etiketlash har bir hujjatga o'z sinfi bilan izoh berish

jarayonini anglatadi. Ammo etiketkalash qoidalar yozishdan ko'ra osonroq ishdir. Deyarli har bir kishi hujjatni ko'rib chiqishi va qaror qabul qilishi mumkin. Ba'zida bunday yorliq allaqachon mavjud ish oqimining bir qismidir. Misol uchun, siz har kuni ertalab doimiy so'rov orqali qaytariladigan yangiliklar maqolalarini ko'rib chiqishingiz va tegishli maqolalarni ko'p yadroli protsessorlar kabi maxsus papkaga ko'chirish orqali tegishli fikr-mulohazalaringizni bildirishingiz mumkin. Ushbu bobni matnni tasniflash muammosiga umumiy kirish, shu jumladan rasmiy ta'rif bilan boshlaymiz (2.1-bo'lim). Keyin Naive Bayesni, ayniqsa oddiy va samarali tasniflash usuli ko'rib chiqiladi (2.2-2.4-bo'limlar). O'rganayotgan barcha tasniflash algoritmlari hujjatlarni yuqori o'lchamli bo'shliqlarda ifodalaydi. Ushbu algoritmlarning samaradorligini oshirish uchun, odatda, bu bo'shliqlarning o'lchamlarini kamaytirish maqsadga muvofiqdir. Shu maqsadda, matn tasnifida xususiyat tanlash deb nomlanuvchi usul qo'llaniladi. 2.6-bo'lim matn tasnifini baholashni o'z ichiga oladi. Keyingi boblarda, 3 va 4-boblarda tasniflash usullarining yana ikkita turkumini, vektor fazosi tasniflagichlarini va vektorli mashinalar ko'rib chiqiladi.

2.1. Matnni tasniflash muammosi

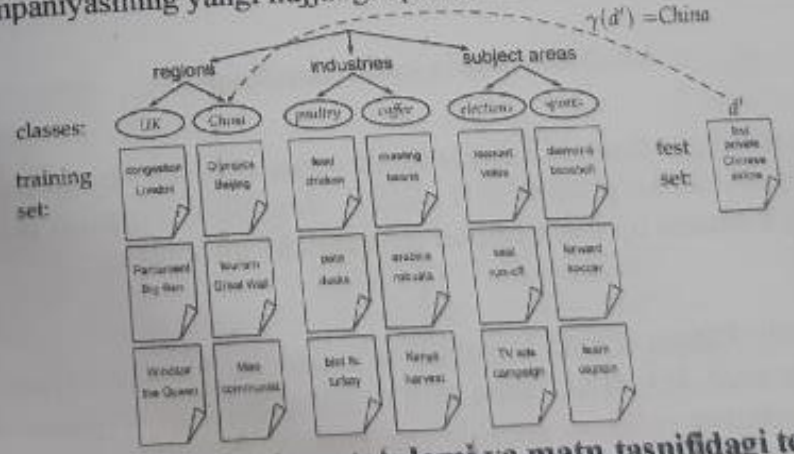
Matn tasnifida ularga hujjatning $d \in x$ tavsifi beriladi, bu yerda X hujjat maydoni va qat'iy sinflar to'plami $C = \{c_1, c_2, ..., c_j\}$. Sinflar toifalar yoki teglar deb ham ataladi. Odatda, X hujjat maydoni yuqori o'lchamli makonning bir turi bo'lib, sinflar Xitoy misollarida va yuqoridagi ko'p yadroli kompyuter chiplari haqida gapiradigan hujjatlarda bo'lgani kabi dastur ehtiyojlari uchun aniqlangan odamlardir. Ularga **hd**, **ci** kletirish mumkin. Bu yerda **hd**, $\langle d, c \rangle \in X \times C$ etiketli hujjatlardan iborat **D** o'quv to'plami hisoblanadi. Masalan:

$$\langle d, c \rangle = \langle Beijing\ \ joins\ the\ World\ \ Trade\ \ Organizati\ \ on, China \rangle$$

bir jumla hujjat uchun Pekin Jahon Savdo Tashkilotiga va sinf Xitoyga qo'shiladi. O'rganish usuli yoki o'rganish algoritmidan foydalanib, hujjatlarni sinflarga taqqoslaydigan klassifikator yoki tasniflash funksiyasini o'rganish kerak bo'lsa:

$$\lambda : X \to C \qquad (2.1)$$

Ushbu turdagi ta'lim nazorati ostida o'rganish deb ataladi, chunki nazoratchi (sinflarni belgilaydigan va o'quv hujjatlarini belgilovchi inson) o'quv jarayonini boshqaradigan o'qituvchi bo'lib xizmat qiladi. Nazorat ostidagi o'rganish usuli **D** bilan belgilanadi va **D(D) = g** ni yozamiz. **D** o'rganish usuli **D** o'quv to'plamini kirish sifatida qabul qiladi va o'rganilgan tasniflash funksiyasini qaytaradi. **D** o'rganish usullarining ko'p nomlari **g** tasniflagichlari uchun ham qo'llaniladi. **Naive Bayes**(NB) o'rganish usuli **D** haqida "Naive Bayes mustahkam" deganda gapiramiz, ya'ni u juda ko'p turli xil o'quv muammolariga qo'llanilishi mumkin va halokatli tarzda muvaffaqiyatsiz bo'lgan tasniflagichlarni ishlab chiqarishi dargumon. Ammo "Naive Bayes xato darajasi 20% edi" deganda, ma'lum bir NB tasniflagichi **g** (NB o'rganish usuli bilan ishlab chiqarilgan) ilovada 20% xatolik darajasiga ega bo'lgan tajribani tasvirlaymiz. 2.1-rasmda keltirilgan **Reuters-RCV1** to'plamidan matn tasniflash misoli ko'rsatilgan. Oltita sinf (Buyuk Britaniya, Xitoy, ..., sport) mavjud bo'lib, ularning har birida uchta o'quv hujjatlari mavjud. Har bir hujjat mazmuni uchun bir nechta mnemonik so'zlarni ko'rsatish mumkin. O'quv to'plamida har bir sinf uchun ba'zi tipik misollar keltirilgan, shuning uchun **g** tasniflash funksiyasini o'rganish mumkin. **g** ni o'rganganimizdan so'ng, uni test to'plamiga (yoki test ma'lumotlariga), masalan, sinfi noma'lum bo'lgan birinchi xususiy Xitoy aviakompaniyasining yangi hujjatiga qo'llashimiz mumkin.



2.1- rasm. **Sinflar, o'quv to'plami va matn tasnifidagi testlar to'plami**

2.1-rasmda tasniflash funksiyasi yangi hujjatni **g(d)** = Xitoy sinfiga belgilaydi, bu to'g'ri tayinlashdir.

Matn tasnifidagi sinflar ko'pincha 2.1-rasmdagi ierarxiya kabi qiziqarli tuzilishga ega. Mintaqa toifalari, sanoat toifalari va mavzu sohasi toifalarining har birida ikkita misol mavjud. Bu ierarxiya tasniflash masalasini hal qilishda muhim yordamchi bo'lishi mumkin. Keyingi muhokama uchun 4.3.2-bo'limga qarang. Ungacha, matn tasniflash boblarida ..... hech qanday kichik to'plam munosabatlari bo'lmagan to'plamni tashkil qiladi. Ta'rif (2.1) hujjatning aynan bitta sinfga tegishli ekanligini ko'rsatadi. Bu 2.1-rasmdagi ierarxiya uchun eng mos model emas. Masalan, 2008 yilgi Olimpiada haqidagi hujjat ikkita sinfning a'zosi bo'lishi kerak: Xitoy sinfi va sport sinfi. Ushbu turdagi tasniflash muammosi har qanday muammo deb ataladi va ular unga 14.5-bo'limda qaytamiz (306-bet). Hozircha ular hujjat aynan bitta sinfga tegishli bo'lgan masalalardan faqat bittasini ko'rib chiqiladi. Matnni tasniflashda ularning maqsadimiz test ma'lumotlari yoki yangi ma'lumotlarning yuqori aniqligidir - masalan, ertaga ertalab ular ko'p yadroli chip misolida duch keladigan newswire maqolalari. Trening to'plamida yuqori aniqlikka erishish oson (masalan, ular teglarni oddiygina yodlashimiz mumkin). Ammo o'quv majmuasidagi yuqori aniqlik klassifikator ilovadagi yangi ma'lumotlarda yaxshi ishlashini anglatmaydi. Sinov ma'lumotlari uchun klassifikatorni o'rganish uchun o'quv to'plamidan foydalanganda, ular o'quv ma'lumotlari va test ma'lumotlari o'xshash yoki bir xil taqsimotdan deb taxmin qiladilinadi. Ushbu tushunchaning aniq ta'rifini 3.6-bo'limga qoldiramiz.

2.2. Naive Bayes matn tasniflagichi

Joriy qiladigan birinchi nazorat ostida o'rganish usuli - bu ko'p nomli **Naive Bayes** yoki ko'p nomli NB modeli, ehtimollik o'rganish usuli. **d** hujjatining **c** sinfida bo'lish ehtimoli quyidagicha hisoblanadi:

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c) \qquad (2.2)$$

Bu yerda **P(tk|c)** **c** sinf hujjatida t_k muddatni yuzaga kelishining shartli ehtimolidir. P(tk|c) ni c ning to'g'ri sinf ekanligiga t_k qancha dalil hissa qo'shishining o'lchovi sifatida izohlaymiz. P(c) - c sinfida sodir bo'lgan hujjatning oldingi ehtimolidir. Agar hujjat shartlari bir sinfga nisbatan boshqasiga aniq dalil keltirmasa, oldingi ehtimoli yuqori bo'lgani tanlanadi. ht1, t2, . . . , tn

d_i - tasniflash uchun foydalanadigan lug'at tarkibiga kiruvchi **d** dagi leksemalar va n_d - d dagi tokenlar soni. Masalan, t_1, t_2, . . . , t_{nd} bir jumladan iborat hujjat uchun Pekin va Taypey JSTga qo'shilishlari mumkin Beijing, Taipei, join, WTOi, with n_d= 4 bo'ladi agar shartlar va to'xtash so'zlari ko'rib chiqilsa. Matn tasnifida ularning maqsadi hujjat uchun eng yaxshi sinfni topishdir. NB tasnifidagi eng yaxshi klass bu eng ehtimol yoki maksimal posteriori sinf c xaritasidir:

$$c_{map} = \arg\max_{c \in C} \hat{P}(c|d) = \arg\max_{c \in C} \hat{P}(c) \qquad (2.3)$$

P(c) va P(tk|c) parametrlarining haqiqiy qiymatlarini bilmaganimiz sababli P uchun ^P yozamiz, lekin ularni bir lahzadan keyin ko'rib turganimizdek o'quv majmuasidan baholaymiz. (2.3) tenglamada ko'plab shartli ehtimollar ko'paytiriladi, har bir pozitsiya uchun bittadan $1 \le k \le n_d$. Bu suzuvchi nuqtaning quyi oqimiga olib kelishi mumkin. Shuning uchun hisoblashni ehtimollarni ko'paytirish o'rniga ehtimollarning logarifmlarini qo'shish orqali amalga oshirish yaxshiroqdir. Eng yuqori jurnal ehtimoli ballga ega bo'lgan sinf hali ham eng yuqori ehtimolli hisoblanadi. ***log(xy) = log(x) + log(y)*** va logarifm funksiyasi monotonikdir. Shunday qilib, NB ning aksariyat ilovalarida amalga oshiriladigan maksimallashtirish quyidagicha hisoblanadi:

$$c_{map} = \arg\max_{c \in C} \left[\log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c) \right]. \qquad (2.4)$$

(2.4) tenglama oddiy talqinga ega. Har bir shartli parametr logi $\hat{P}(t_k|c)$ indikatorining **c** uchun qanchalik yaxshi ekanligini ko'rsatuvchi og'irlikdir. Xuddi shunday, oldingi log^P(c) ham **c** ning nisbiy chastotasini ko'rsatadigan og'irlikdir. Tez-tez o'tkaziladigan darslar kamdan-kam uchraydigan sinflarga qaraganda to'g'ri sinf bo'lish ehtimoli ko'proq. Jurnaldan oldingi va muddat og'irliklarining yig'indisi hujjatning sinfda mavjudligi uchun qancha dalillar mavjudligini o'lchovidir va (2.4) tenglama ularda eng ko'p dalillarga ega bo'lgan sinfni tanlaydi. Dastlab ko'p nomli NB modelining ushbu intuitiv talqini bilan ishlaymiz va rasmiy hosilani 2.4-bo'limga qoldiramiz. $\hat{P}(t_k|c)$ parametrlarini qanday baholaymiz? Birinchi navbatda maksimal ehtimollik taxminini sinab ko'riladi bu oddiygina nisbiy chastotadir va

o'quv ma'lumotlari berilgan har bir parametrning eng ehtimoliy qiymatiga mos keladi.

$$\hat{P}(c) = \frac{N_c}{N}, \quad (2.5)$$

Bu yerda Nc - c sinfidagi hujjatlar soni va N - hujjatlarning umumiy soni. Shartli ehtimollik $\hat{P}(t|c)$ ni c sinfiga kiruvchi hujjatlardagi t terminining nisbiy chastotasi sifatida baholaymiz:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}, \quad (2.6)$$

Bu yerda T_{ct} - c sinfidagi o'quv hujjatlarida t ning takrorlanish soni, shu jumladan hujjatdagi atamaning bir necha marta takrorlanishidir. Bu yerda pozitsion mustaqillik farazini keltirildi, bu haqda keyingi bobda batafsil to'xtalib o'tiladi. T_{ct} – o'quv majmuasidagi hujjatlardagi barcha k pozitsiyalarda sodir bo'lgan holatlar soni. Shunday qilib, turli pozitsiyalar uchun turli baholar hisoblanadi va masalan, agar so'z hujjatda k_1 va k_2 pozitsiyalarida ikki marta uchrasa, u holda

$$\hat{P}(t_{k_1}|c) = \hat{P}(t_{k_2}|c).$$

Maxsimal ehtimollik tahmini (MET) bahosi bilan bog'liq muammo shundaki, u o'quv ma'lumotlarida uchramagan davr-sinf kombinatsiyasi uchun nolga teng. Agar o'quv ma'lumotlaridagi **JST** atamasi faqat Xitoy hujjatlarida bo'lsa, boshqa sinflar uchun MLE baholari, masalan, Buyuk Britaniya hujjatlarida nolga teng bo'ladi: $\hat{P}(WTO|UK) = 0$

Buyuk Britaniya JST a'zosi bo'lgan bir jumladan iborat hujjat Buyuk Britaniya uchun nolga teng shartli ehtimollikni oladi chunki (2.2) tenglamadagi barcha shartlar uchun shartli ehtimollarni ko'paytiriladi. Shubhasiz, model bo'lishi kerak.

Muammo shundaki, boshqa xususiyatlardan Buyuk Britaniya sinfi uchun qanchalik kuchli dalillar bo'lishidan qat'i nazar, JSTning nol ehtimolini "shart qilib bo'lmaydi". Taxminan siyraklik tufayli 0 ga teng: O'quv ma'lumotlari hech qachon kamdan-kam uchraydigan hodisalar chastotasini, masalan, Buyuk Britaniya hujjatlarida uchraydigan JST chastotasini yetarli darajada ifodalash uchun yetarli darajada katta emas. Nollarni yo'q qilish uchun qo'shimcha bir yoki **Laplas** silliqlashdan foydalaniladi. Oddiygina har bir sanaga bittadan qo'shiladi:

$$\hat{P}(t|c) = \frac{T_{ct}+1}{\sum_{t' \in V}(T_{ct'}+1)} \quad (2.7)$$

```
TRAINMULTINOMIALNB(C, D)
1  V ← EXTRACTVOCABULARY(D)
2  N ← COUNTDOCS(D)
3  for each c ∈ C
4  do N_c ← COUNTDOCSINCLASS(D, c)
5     prior[c] ← N_c/N
6     text_c ← CONCATENATETEXTOFALLDOCSINCLASS(D, c)
7     for each t ∈ V
8     do T_ct ← COUNTTOKENSOFTERM(text_c, t)
9     for each t ∈ V
10    do condprob[t][c] ← (T_ct+1)/Σ_t'(T_ct'+1)
11 return V, prior, condprob

APPLYMULTINOMIALNB(C, V, prior, condprob, d)
1  W ← EXTRACTTOKENSFROMDOC(V, d)
2  for each c ∈ C
3  do score[c] ← log prior[c]
4     for each t ∈ W
5     do score[c] += log condprob[t][c]
6  return arg max_c∈C score[c]
```

2.2-rasm. **Naive Bayes algoritmi (ko'p nomli model): Trening va test**

bu yerda B = |V| - lug'at tarkibidagi atamalar soni. Bir qo'shimchani tekislash bir xil oldingi (har bir atama har bir sinf uchun bir marta sodir bo'ladi) deb talqin qilinishi mumkin, u keyinchalik o'quv ma'lumotlaridan dalolat sifatida yangilanadi. Hujjat darajasida (2.5) tenglamada baholagan sinfning oldingi ehtimolidir.

13.1-jadval. **Parametrlarni baholash misollari uchun ma'lumotlar**

	docID	words in document	in c = China?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

13.2-jadval. **NB uchun mashg'ulotlar va sinov vaqtlari**

mode	time complexity
training	$\Theta(\|\mathbb{D}\|L_{ave} + \|\mathbb{C}\|\|V\|)$
testing	$\Theta(L_a + \|\mathbb{C}\|M_a) = \Theta(\|\mathbb{C}\|M_a)$

NB tasniflagichini o'rgatish va qo'llash uchun zarur bo'lgan barcha elementlarni kiritdik. To'liq algoritm 2.2-rasmda tasvirlangan.

Misol: 13.1-jadvaldagi misol uchun test hujjatini tasniflashimiz kerak bo'lgan ko'p nomli parametrlar $\hat{P}(c) = 3/4$ va $\hat{P}(\bar{c}) = 1/4$ va quyidagi shartli ehtimollardir:

$$\hat{P}(Chinese \mid c) = (5+1)/(8+6) = 6/14 = 3/7$$

$$\hat{P}(Tokyo \mid c) = \hat{P}(Japan \mid c) = (0+1)/(8+6) = 1/14$$

$$\hat{P}(Chinese \mid \bar{c}) = (1+1)/(3+6) = 2/9$$

$$\hat{P}(Tokyo \mid \bar{c}) = \hat{P}(Japan \mid \bar{c}) = (1+1)/(3+6) = 2/9$$

Maxrajlar (8 + 6) va (3 + 6) bo'ladi, chunki $\mathbf{text}_c$ va $\mathbf{text}_{\bar{c}}$ uzunligi mos ravishda 8 va 3 ga teng hamda (2.7) tenglamadagi **B** doimiysi 6 ga teng chunki lug'at oltita terminдan iborat. Ular quyidagicha hisoblanadi:

$$\hat{P}(c \mid d_5) \propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003 .$$

$$\hat{P}(\bar{c} \mid d_5) \propto 1/4 \cdot (2/9)^3 \cdot 2/19 \cdot 2/9 \approx 0.0001 .$$

Shunday qilib, tasniflagich test hujjatini c = Xitoyga belgilaydi. Ushbu tasniflash qarorining sababi shundaki, $\mathbf{d}_5$-da Xitoyning ijobiy ko'rsatkichining uchta hodisasi ikkita salbiy ko'rsatkich Yaponiya va Tokioning paydo bo'lishidan ustundir.

NB ning vaqt murakkabligi qanday? Parametrlarni hisoblashning murakkabligi $\Theta(|C||V|)$ ga teng chunki parametrlar to'plami |C||V| dan iborat. shartli ehtimollar va |C| avvalgi boblarda hisoblangandek hisoblanadi. Parametrlarni hisoblash uchun zarur bo'lgan dastlabki ishlov berish (lug'atni ajratib olish, atamalarni sanash va boshqalar) o'quv ma'lumotlari orqali bir marta amalga oshirilishi mumkin. Shuning uchun bu komponentning vaqt murakkabligi $\Theta(|D|L_{ave})$ bu yerda **|D|** hujjatlar soni va $\mathbf{L}_{ave}$ - hujjatning o'rtacha uzunligi.

Bu yerda **D(T)** belgisi sifatida **D(|D| L_{ave})** dan foydalaniladi, bu yerda **T** o'quv yig'indisining uzunligi. Bu nostandart hisoblanadi. **D(.)** o'rtacha uchun aniqlanmagan. Vaqt murakkabligini **D** va $\mathbf{L}_{ave}$ ko'rinishida ifodalanilishi afzal chunki bular o'quv to'plamlarini tavsiflash uchun ishlatiladigan asosiy statistik ma'lumotlardir. 2.2-rasmdagi APPLYMULTINOMIALNB ning vaqt murakkabligi **D(|C|L_{ave}). L_a** va **M_a** test hujjatidagi mos ravishda tokenlar va turlarning raqamlaridir. APPLYMULTINOMIALNB ni $\Theta(L_a + |C|M_a) = \Theta(|C|M_a)$ qilib o'zgartirish mumkin. Nihoyat, test hujjatlarining uzunligi chegaralangan deb faraz qilsak, $\Theta(L_a + |C|M_a) = \Theta(|C|M_a)$ because $L_a < b|C|M_a$ ruxsat etilgan doimiy uchun M_a va b dir. 2.2-jadvalda vaqt murakkabliklari jamlangan. Umuman olganda, ularda $|\mathbf{C}||\mathbf{V}| < |\mathbf{D}|\mathbf{L}_{ave}$, shuning uchun o'qitish va sinovning murakkabligi ma'lumotlarni skanerlash uchun zarur bo'lgan vaqt oralig'ida chiziqli hisoblanadi. Ma'lumotlarga kamida bir marta qarashimiz kerakligi sababli, NB optimal vaqt murakkabligiga ega deb aytish mumkin. Uning samaradorligi NB matnni tasniflashning mashhur usuli ekanligining sabablaridan biridir.

2.2.1. Ko'p nomli unigramma tili modeli

Ko'p nomli unigramma tili modeli bilan rasmiy ravishda bir xil nomli NB modeli mavjud (1.2.1-bo'lim). Xususan, (2.2) tenglama, (1.12) tenglamaning maxsus holati bo'lib, bu yerda l = 1 uchun takrorlanadi:

$$P(d \mid q) \propto P(d) \prod_{t \in q} P(t \mid M_d) \qquad (2.8)$$

Matn tasnifidagi hujjat **d** (tenglama (1.2) tilni modellashtirishda so'rov rolini o'ynaydi (Tenglama (1.8) va matn tasnifidagi **c** sinflari tilni modellashtirishda **d** hujjat rolini oladi. Hujjatlarni **q** so'roviga mos kelishi ehtimoli bo'yicha tartiblash uchun (1.8) tenglamadan foydalanildi. NB tasnifida odatda faqat yuqori darajadagi sinf ahamiyatli hisoblanadi. Shuningdek, 1.2.2-bo'limda MLE taxminlaridan foydalandik va kam ma'lumotlar tufayli nol baholash muammosiga duch keldik. Lekin bitta qo'shimcha silliqlash o'rniga u yerda muammoni hal qilish uchun ikkita taqsimot aralashmasidan foydalandik.

Misol. Nima uchun |C||V| < |D| 1.2-jadvaldagi ko'p matn to'plamlari uchun mos bo'lishi kutilmoqda?

```
TRAINBERNOULLINB(C, D)
1  V ← EXTRACTVOCABULARY(D)
2  N ← COUNTDOCS(D)
3  for each c ∈ C
4  do N_c ← COUNTDOCSINCLASS(D, c)
5     prior[c] ← N_c/N
6     for each t ∈ V
7     do N_ct ← COUNTDOCSINCLASSCONTAININGTERM(D, c, t)
8        condprob[t][c] ← (N_ct + 1)/(N_c + 2)
9  return V, prior, condprob

APPLYBERNOULLINB(C, V, prior, condprob, d)
1  V_d ← EXTRACTTERMSFROMDOC(V, d)
2  for each c ∈ C
3  do score[c] ← log prior[c]
4     for each t ∈ V
5     do if t ∈ V_d
6           then score[c] += log condprob[t][c]
7           else score[c] += log(1 − condprob[t][c])
8  return arg max_{c∈C} score[c]
```

2.3- rasm. **NB algoritmi (Bernulli modeli)**

Trening va test. 8- qatordagi (yuqorida) qo'shimcha tekislash B = 2 bo'lgan (1.7) tenglamaga o'xshaydi.

2.3. Bernulli modeli

NB klassifikatorini o'rnatishning ikki xil usuli mavjud. Oldingi bo'limda tanishtirgan model multinomial modeldir. U hujjatning har bir pozitsiyasidagi lug'atdan bitta atama hosil qiladi, bu yerda generativ modelni qabul qiladi, bu 1- qismning 11.4-bo'limda batafsilroq muhokama qilingan.

Ko'p nomli modelga alternativa ko'p o'lchovli Bernoulli modelidir. Bu 1- qismning 11.3-bo'limning ikkilik mustaqillik modeliga ekvivalent bo'lib, u lug'atning har bir atamasi uchun ko'rsatkich hosil qiladi yoki hujjatda atama mavjudligini ko'rsatadigan 1 bor yoki yo'qligini ko'rsatadigan 0. 2.3- rasmda Bernulli modeli uchun o'qitish va sinovdan o'tkazish algoritmlari keltirilgan. Bernulli modeli multinomial model bilan bir xil vaqt murakkabligiga ega. Turli avlod modellari turli baholash

strategiyalarini va turli tasniflash qoidalarini nazarda tutadi. Bernulli modeli **P(t|c)** ni **t** atamasini o'z ichiga olgan **c** toifadagi hujjatlarning ulushi sifatida baholaydi (2.3- rasm, TRAINBERNOULLI- NB, 8-qator). Aksincha, multinomial model **P(t|c)** ni **t** atamasini o'z ichiga olgan **c** sinfidagi hujjatlardagi tokenlarning ulushi yoki pozitsiyalar ulushi sifatida baholaydi (2.7 tenglama). Sinov hujjatini tasniflashda Bernulli modeli hodisalar sonini e'tiborsiz qoldirib, ikkilik hodisa ma'lumotlaridan foydalanadi, multinomial model esa bir nechta hodisalarni kuzatib boradi. Natijada, Bernoulli modeli odatda uzun hujjatlarni tasniflashda ko'p xatolarga yo'l qo'yadi. Misol uchun, Xitoy atamasi bir marta kelganligi sababli u butun kitobni Xitoy sinfiga belgilashi mumkin. Modellar tasniflashda noaniq atamalardan foydalanishda ham farqlanadi. Multinomial model tasniflash qaroriga ta'sir qilmaydi. Ammo Bernulli modelida **P(c|d)** ni hisoblashda yuzaga kelmaslik ehtimoli hisobga olinadi (2.3- rasm, A PPLYBERNOULLINB, 7-qator). Buning sababi, faqat Bernoulli NB modeli atamalarning yo'qligini aniq modellaydi.

Misol. Bernulli modelini 2.1- jadvaldagi misolga tadbiq qilsak, oldingilar uchun xuddi shunday baholarga ega bo'lamiz: $\hat{P}(c) = 3/4, \hat{P}(\bar{c}) = 1/4$. Shartli ehtimollar quyidagilardir:

$$\hat{P}(Chinese \mid c) = (3+1)/(3+2) = 4/5$$

$$\hat{P}(Japan \mid c) = \hat{P}(Tokyo \mid c) = (0+1)/(3+2) = 1/5$$

$$\hat{P}(Beijing \mid c) = \hat{P}(Macao \mid c) = \hat{P}(Shanghai \mid c) = (1+1)/(3+2) = 2/5$$

$$\hat{P}(Chinese \mid \bar{c}) = (1+1)/(1+2) = 2/3$$

$$\hat{P}(Japan \mid \bar{c}) = \hat{P}(Tokyo \mid \bar{c}) = (1+1)/(1+3) = 2/3$$

$$\hat{P}(Beijing \mid \bar{c}) = \hat{P}(Macao \mid \bar{c}) = \hat{P}(Shanghai \mid \bar{c}) = (0+1)/(1+2) = 1/3$$

Maxrajlar (3 + 2) va (1 + 2), chunki **c** da uchta hujjat va **c** da bitta hujjat va (2.7) tenglamadagi **B** doimiysi 2 ga teng bo'lganligi sababli - har bir atama, hodisa uchun ikkita holatni ko'rib chiqish kerak. Ikki sinf uchun test hujjatining ballari

$$\hat{P}(c \mid d_s) \propto \hat{P}(c) \cdot \hat{P}(Chinese \mid c) \cdot \hat{P}(Japan \mid c) \cdot \hat{P}(Tokyo \mid c)$$

$$\cdot (1 - \hat{P}(Beijing \mid c)) \cdot (1 - \hat{P}(Shanghai \mid c)) \cdot (1 - \hat{P}(Macao \mid c))$$

$$= 3/4 \cdot 4/5 \cdot 1/5 \cdot 1/5 \cdot (1 - 2/5) \cdot (1 - 2/5) \cdot (1 - 2/5) \approx 0.005$$

va shunga o'xshash,

$\hat{P}(\bar{c}\mid d_5) \approx 1/4 \cdot 2/3 \cdot 2/3 \cdot 2/3 \cdot (1-1/3)\cdot(1-1/3)\cdot(1-1/3) \approx 0.022$

Shunday qilib, tasniflagich test hujjatini **c** != Xitoy. Termin chastotasiga emas, faqat ikkilik hodisaga qaralganda, Yaponiya va Tokio **c (2/3 > 1/5)** uchun ko'rsatkichdir va xitoychaning **c** uchun shartli ehtimolliklari yetarlicha farq qilmaydi ***(4/5 va 2/3)*** tasniflash qaroriga ta'sir qilishi mumkin.

2.4. Naive Bayesning xususiyatlari

Ikki modelni va ular qilgan taxminlarni yaxshiroq tushunish uchun keling. 1-boblarda ularning tasnifi qoidalarini qanday olganlini ko'rib chiqiladi. Posteriori ehtimoli quyidagi tarzda hisoblanadi:

$$c_{map} = \arg\max_{c \in C} P(c \mid d) = \arg\max_{c \in C} \frac{P(d \mid c)P(c)}{P(d)} = \arg\max_{c \in C} P(d \mid c)P(c), \quad (2.10)$$

Bu yerda (2.9) da Bayes qoidasi (1- qismning 11.4 tenglamasi) qo'llaniladi va oxirgi bosqichda maxrajni tushiramiz, chunki **P(d)** barcha sinflar uchun bir xil va argmaxga ta'sir qilmaydi. (2.10) tenglamani Bayes matn tasnifida qabul qilgan generativ jarayonning tavsifi sifatida talqin qilish mumkin. Hujjatni yaratish uchun birinchi navbatda **P(c)** ehtimoli bilan **c** sinfini tanlaymiz (2.4 va 2.5- rasmlardagi yuqori tugunlar). Ikki model ikkinchi bosqichni rasmiylashtirishda, **P(d|c)** shartli taqsimotiga mos keladigan sinf berilgan hujjatni yaratishda farqlanadi:

Multinomial $P(d \mid c) = P(\langle t_1, \ldots, t_k, \ldots, t_{n_d} \rangle \mid c)$ (2.11)

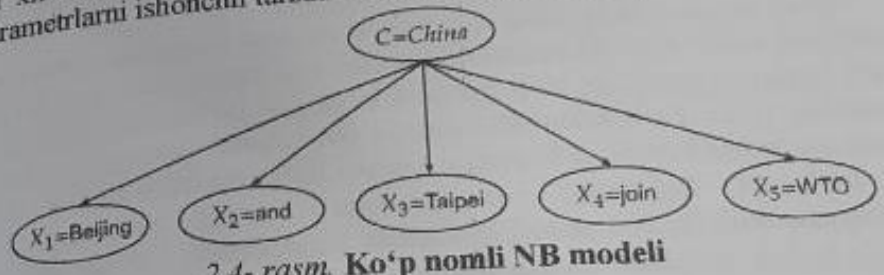
Bernoulli $P(d \mid c) = P(\langle e_1, \ldots, e_i, \ldots, e_M \rangle \mid c)$, (2.12)

bu yerda $t_1, \ldots, t_{nd}$ i - atamalar ketma-ketligi, chunki u **d** (lug'atdan chiqarib tashlangan atamalar minus) va $e_1, \ldots, e_i, \ldots, e_M$ i_M o'lchamlilik ikkilik vektori bo'lib, u har bir a'zo uchun **d** da bo'ladimi yoki yo'qligini ko'rsatadi.

Endi tasniflash masalasini aniqlanganda nima uchun (1.1) tenglamaga X hujjat maydoni kiritilgani aniqroq bo'lishi kerak. Matnni tasniflash muammosini hal qilishda muhim qadam hujjat tasvirini

tanlashdir. $ht_1, \ldots, t_{nd}$ i va $u_1, \ldots, e_M$ i ikki xil hujjat tasviri. Birinchi holda, X - barcha terminlar ketma-ketligi (yoki aniqrog'i, termin belgilarining ketma-ketligi). Ikkinchi holda, X {0, 1} M.

Matnni tasniflash uchun (2.11) va (2.12) tenglamalardan bevosita foydalana olmaymiz. Bernulli modeli uchun $2^M|C|$ ni taxmin qilishimiz kerak edi. Turli parametrlar, M qiymatlari e_i va sinfning har bir mumkin bo'lgan kombinatsiyasi uchun bitta. Ko'p nomli holatda parametrlar soni bir xil kattalik tartibiga ega. Bu juda katta miqdor bo'lgani uchun bu parametrlarni ishonchli tarzda baholash mumkin emas.



2.4- rasm. **Ko'p nomli NB modeli**

Multinomial $P(d \mid c) = P(\langle t_1, \ldots, t_k, \ldots, t_{n_d} \rangle \mid c) = \prod_{1 \le k \le n_d} P(X_k = t_k \mid c)$ (2.13)

Bernoulli $P(d \mid c) = P(\langle e_1, \ldots, e_i, \ldots, e_M \rangle \mid c) = \prod_{1 \le i \le M} P(U_i = e_i \mid c)$ (2.14)

Ikki xil generativ modelni aniq ko'rsatish uchun bu yerda ikkita tasodifiy o'zgaruvchini kiritdik. X_k - hujjatdagi **k** pozitsiyasi uchun tasodifiy o'zgaruvchidir va lug'atdan qiymatlar sifatida qabul qiladi. $\hat{P}$ **(Xk= t|c)** - **c** sinfidagi hujjatda **t** atamasi **k** pozitsiyasida paydo bo'lish ehtimoli. U_i lug'at termini **i** uchun tasodifiy o'zgaruvchi bo'lib, 0 (yo'qligi) va 1 (mavjudligi) qiymatlarini qabul qiladi. $\hat{P}$ **(U i = 1|c)** - **c** sinfidagi hujjatda t_i atamasi har qanday holatda va ehtimol bir necha marta paydo bo'lish ehtimoli.

Shartli mustaqillik farazini 2.4 va 2.5-rasmlarda tasvirlash mumkin. Xitoy sinfi boshqa atributlarning qiymatlaridan qat'i nazar, ma'lum bir ehtimollik bilan besh atama atributining (ko'p nomli) yoki oltita ikkilik atributning (Bernulli) har biri uchun qiymatlarni yaratadi. Xitoy sinfidagi hujjat **Taypey** atamasini o'z ichiga olganligi uning **Pekin**ni ham o'z ichiga olishi ehtimolini ko'proq yoki kamroq qilmaydi. Shartlar shartli ravishda bir-biriga bog'liq. Ammo qisqa vaqt ichida muhokama

qiladiganidek. NB modellari shartli mustaqillik faraziga qaramay yaxshi ishlaydi.

2.5- rasm. **Bernoulli NB modeli**

Shartli mustaqillikni qabul qilganda ham, agar hujjatdagi har bir **k** pozitsiyasi uchun turli xil ehtimollik taqsimotini qabul qilsak ham ko'p nomli model uchun juda ko'p parametrlarga ega bo'lamiz. Hujjatdagi atamaning o'rni o'z-o'zidan sinf haqida ma'lumotni bermaydi. Xitoy Fransiyani sudga berishi va Fransiya Xitoyni sudga berishi o'rtasida farq bor bo'lsa-da, hujjatning 3-pozitsiyasiga nisbatan Xitoyning 1-pozitsiyada bo'lishi NB tasnifida foydali emas chunki har bir atama alohida ko'rib chiqiladi. Shartli mustaqillik farazi ularni dalillarni shu tarzda qayta ishlashga majbur qiladi. Bundan tashqari, agar har bir **k** pozitsiyasi uchun har xil atamalar taqsimotini qabul qilsak, har bir **k** uchun har xil parametrlar to'plamini taxmin qilishimiz kerak bo'ladi. Fasolning qahva hujjatining birinchi atamasi sifatida paydo bo'lish ehtimoli ikkinchi muddat sifatida paydo bo'lishidan farq qilishi mumkin va hokazo. Bu yana ma'lumotlarning siyrakligi tufayli baholashda muammolarni keltirib chiqaradi. Shu sabablarga ko'ra, ko'p nomli model uchun ikkinchi mustaqillik farazini, pozitsion mustaqillikga o'zgartiradi. Termin uchun shartli ehtimollar hujjatdagi pozitsiyadan mustaqil ravishda bir xil.

$$P(X_{k_1} = t \mid c) = P(X_{k_2} = t \mid c)$$

barcha lavozimlar uchun k_1, k_2, shartlar **t** va sinflar **c** hisoblanadi. Shunday qilib, k_i barcha pozitsiyalar uchun amal qiladigan atamalarning yagona taqsimotiga ega va uning belgisi sifatida **X** dan foydalanish mumkin. Pozitsion mustaqillik 1- qismning 6-bobida maxsus qidiruv kontekstida kiritgan **so'zlar sumkasi** modelini qabul qilish bilan tengdir.

Shartli va pozitsion mustaqillik farazlari bilan faqat **D(M|C|)** parametrlarini **P(tk|c)** (ko'p nomli model) yoki **P(ei|c)** (Bernulli modeli),

har bir atama-sinf birikmasi uchun bittadan baholanilishi kerak, **M**-da hech bo'lmaganda eksponent bo'lgan raqam emas lug'at hajmi baholanadi. Mustaqillik taxminlari baholanishi kerak bo'lgan parametrlar sonini bir necha darajaga qisqartiradi.

2.3- jadval. **Multinomial va Bernulli modeli**

	multinomial model	Bernoulli model
event model	generation of token	generation of document
random variable(s)	$X = t$ iff t occurs at given pos	$U_t = 1$ iff t occurs in doc
document representation	$d = \langle t_1, \ldots, t_k, \ldots, t_{n_d} \rangle, t_k \in V$	$d = \langle e_1, \ldots, e_i, \ldots, e_M \rangle$, $e_i \in \{0,1\}$
parameter estimation	$\hat{P}(X = t\|c)$	$\hat{P}(U_i = e\|c)$
decision rule: maximize	$\hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(X = t_k\|c)$	$\hat{P}(c) \prod_{i \in V} \hat{P}(U_i = e_i\|c)$
multiple occurrences	taken into account	ignored
length of docs	can handle longer docs	works best for short docs
# features	can handle more	works best with fewer
estimate for term the	$\hat{P}(X = \text{the}\|c) \approx 0.05$	$\hat{P}(U_{\text{the}} = 1\|c) \approx 1.0$

Xulosa qilib aytadigan bo'lsak, multinomial modelda hujjat hosil qilinadi (2.4-rasm) birinchi navbatda **P(c)** bilan **C = c** sinfini tanlanadi, bu yerda **C** tasodifiy o'zgaruvchidir, **C** dan qiymatlarni qiymat sifatida qabul qiladi. Keyin hujjatning har bir n_d pozitsiyasi uchun $P(X_k = t_k|c)$ bilan **k** pozitsiyasida t_k atama hosil qiladi. X_k hammasi berilgan **c** uchun atamalar bo'yicha bir xil taqsimotga ega. 2.4-rasmdagi misolda Pekin va Taypey JSTga qo'shilish to'g'risidagi bir jumladan iborat hujjatga mos keladigan t_1, t_2, t_3, t_4, t_5 $i = h$ Beijing and Taypei join WTO i avlodini ko'rsatamiz. To'liq aniqlangan hujjat yaratish modeli uchun ham shunday bo'lar edi.

K uzunliklar bo'yicha $\mathbf{P(n_d|c)}$ taqsimotini aniqlash kerak. Busiz multinomial model hujjatlarni yaratish modeli emas, balki tokenlarni yaratish modeli bo'ladi. Hujjatni Bernulli modelida (2.5- rasm) avval **P(c)** bilan **C=c** sinfini tanlab, so'ngra lug'atning har bir t_i atamasi uchun $(1 \le i \le M)$ ikkilik indikator e_i hosil qiladi. 2.5-rasmdagi misolda bir gapli Pekin va Taypey hujjatiga mos keladigan $\langle e_1, e_2, e_3, e_4, e_5, e_6 \rangle = \langle 0, 1, 0, 1, 1, 1 \rangle$ avlodini ko'rsatamiz. JSTga a'zo bo'ling, bu yerda ular taxmin qildik va bu to'xtash so'zi.

2.3- jadvaldagi ikkita modelni jumladan, baholash tenglamalari va qaror qabul qilish qoidalari solishtirilaid. ***Sodda Bayes*** shunday deb ataladi chunki hozirgina qilgan mustaqillik haqidagi taxminlar tabiiy til modeli uchun juda soddadir. Shartli mustaqillik farazi sinfni hisobga

olgan holda xususiyatlar bir-biridan mustaqil ekanligini bildiradi. Bu hujjatlardagi shartlar uchun deyarli hech qachon to'g'ri kelmaydi. Ko'p hollarda buning aksi bo'ladi. 2.7- rasmdagi gong va kong yoki london va ingliz juftliklari juda bog'liq atamalarga misoldir. Bundan tashqari, ***multinomial model*** pozitsion mustaqillik farazini keltirib chiqaradi. ***Bernoulli modeli*** hujjatlardagi pozitsiyalarni umuman e'tiborsiz qoldiradi, chunki u so'zni faqat yo'qligi yoki mavjudligi haqida qayg'uradi. Ushbu ***so'zlar sumkasi modeli*** tabiiy tildagi jumlalardagi so'zlar tartibi bilan bog'langan barcha ma'lumotlarni yo'q qiladi. ***Tabiiy til modeli*** juda soddalashtirilgan bo'lsa, NB qanday qilib yaxshi matn tasniflagichi bo'lishi mumkin?

2.4-jadval. **Aniq baholash jadvali**

	c_1	c_2	class selected
true probability $P(c\|d)$	0.6	0.4	c_1
$\hat{P}(c)\prod_{1\le k\le n_d}\hat{P}(t_k\|c)$ (Equation (13.13))	0.00099	0.00001	
NB estimate $\hat{P}(c\|d)$	0.99	0.01	c_1

To'g'ri baholash to'g'ri bashorat qilishni anglatadi, ammo aniq bashorat to'g'ri baholashni anglatmaydi. Javob shundaki, NB ning ehtimollik baholari past sifatga ega bo'lsa-da, uning tasniflash qarorlari hayratlanarli darajada yaxshi. Haqiqiy ehtimolliklari **$P(c_1|d) = 0,6$ va $P(c_2|d) = 0,4$** bo'lgan **d** hujjatni ko'rib chiqaylik.

2.4- jadvalda ko'rsatilganidek. Faraz qilaylik, **d** ning tarkibida c_1 uchun ijobiy ko'rsatkichlar bo'lgan ko'plab atamalar va c_2 uchun salbiy ko'rsatkichlar bo'lgan ko'plab atamalar mavjud. Shunday qilib, (2.13) tenglamada ko'p nomli modeldan foydalanilganda $\hat{P}(c_1)\prod_{1\le k\le n_d}\hat{P}(t_k|c_1)$ $\hat{P}(c_2)\prod_{1\le k\le n_d}\hat{P}(t_k|c_2)$ dan ancha katta bo'ladi (jadvaldagi 0,00099 ga nisbatan 0,00001). $P(c|d)$ uchun to'g'ri tuzilgan ehtimolliklarni olish uchun 0,001 ga bo'lingandan so'ng 1,0 ga yaqin va 0,0 ga yaqin bo'lgan bitta taxminga erishiladi. Bu keng tarqalgan. NB tasnifida g'olib sinf odatda boshqa sinflarga qaraganda ancha katta ehtimolga ega va taxminlar haqiqiy ehtimollardan sezilarli darajada farq qiladi. Ammo tasniflash qarori qaysi sinf eng yuqori ball olganiga asoslanadi. Hisob-kitoblarning qanchalik to'g'ri ekanligi muhim emas. Noto'g'ri baholarga qaramay, NB c_1 uchun yuqori ehtimollikni baholaydi va shuning uchun **d** ni 2.4- jadvalda to'g'ri sinfga belgilaydi.

To'g'ri baholash to'g'ri bashorat qilishni anglatadi, ammo aniq bashorat to'g'ri baholashni anglatmaydi. NB tasniflagichlari yomon baholaydi, lekin ko'pincha yaxshi tasniflanadi. Matn uchun eng yuqori aniqlikka ega bo'lgan usul bo'lmasa ham NB uni matn tasnifi uchun kuchli da'vogarga aylantiradigan ko'plab fazilatlarga ega. Tasniflash qaroriga birgalikda hissa qo'shadigan bir xil darajada muhim xususiyatlar mavjud bo'lsa, u ustundir. Shuningdek, u shovqin xususiyatlariga (keyingi bo'limda) va kontseptsiyaning o'zgarishiga nisbatan mustahkamdir - vaqt o'tishi bilan AQSh prezidenti Bill Klintondan Jorj V. Bushgacha bo'lgan sinfning asosini tashkil etuvchi kontseptsiyaning bosqichma-bosqich o'zgarishi (2.7- bo'lim). KNN kabi tasniflagichlar (3.3- bo'lim) ma'lum bir vaqt davrining o'ziga xos xususiyatlariga ehtiyotkorlik bilan sozlanishi mumkin. Bu keyingi davrdagi hujjatlar biroz boshqacha xususiyatlarga ega bo'lsa, ularga zarar keltiradi.

2.5-jadval. **Milliy bankning mustaqilligi haqidagi taxminlari muammoli bo'lgan hujjatlar to'plami.**

(1) He moved from London, Ontario, to London, England.
(2) He moved from London, England, to London, Ontario.
(3) He moved from England to London, Ontario.

Bernulli modeli kontseptsiyaning tarkibiga nisbatan ayniqsa mustahkamdir. 2.8- rasmda ko'rib chiqiladiki, u o'ndan kam atamalardan foydalanganda munosib ishlashga ega bo'lishi mumkin. Sinf uchun eng muhim ko'rsatkichlar o'zgarishi ehtimoli kamroq. Shunday qilib, faqat ushbu xususiyatlarga tayanadigan model, kontseptsiyaning tarkibida ma'lum bir aniqlik darajasini saqlab qolish ehtimoli ko'proq. NB ning asosiy kuchi uning samaradorligidir. O'qitish va tasniflash ma'lumotlar orqali bir marta amalga oshirilishi mumkin. U samaradorlikni yaxshi aniqlik bilan birlashtirgani uchun u ko'pincha matn tasnifini tadqiq qilishda asos sifatida ishlatiladi. Ko'pincha, agar (i) matnni tasniflash ilovasida aniqlikning bir necha qo'shimcha foiz punktlarini siqib chiqarish qiyinchilik tug'dirmasa, (ii) juda katta miqdordagi o'quv ma'lumotlari mavjud bo'lsa va undan ko'p narsalarni olish kerak bo'lsa tanlov usuli hisoblanadi. Kichikroq o'quv to'plamida yaxshiroq tasniflagichdan foydalanishdan ko'ra ko'p ma'lumotlarga o'rgatishdan yoki (iii) agar uning kontseptsiyaga chidamliligidan foydalanish mumkin.

Ushbu kitobda NBni matn uchun tasniflagich sifatida muhokama qilinadi. Mustaqillik haqidagi taxminlar matn uchun amal qilmaydi. Shu bilan birga, NB mustaqillik taxminlari mavjud bo'lgan ma'lumotlar uchun optimal tasniflagich (yangi ma'lumotlarda minimal xatolik darajasi ma'nosida) ekanligini ko'rsatish mumkin.

2.4.1. Ko'p nomli modelning varianti

Ko'p nomli modelning muqobil rasmiylashtirilishi har bir **d** hujjatni $\langle tf_{t1,d}, \ldots, tf_{tM,d} \rangle$ sonlarning **M** o'lchovli vektori sifatida ifodalaydi. Bu yerda $tf_{ti,d}$ - **d** dagi t_i ning muddatli chastotasi. Keyin **P(d|c)** quyidagicha hisoblanadi (Tenglama (1.8)).

$$P(d|c) = P(\langle tf_{t_1,d}, \ldots, tf_{t_M,d} \rangle | c) \propto \prod_{1 \le i \le M} P(X = t_i | c)^{tf_{t_i,d}} \qquad (2.15)$$

E'tibor bering, **multinomial** omilni o'tkazib yubordik. (1.8) tenglamaga qarang. (2.15) tenglama (2.2) tenglamadagi ketma-ketlik modeliga ekvivalent bo'lib, $\mathbf{d}(tf_{i,d} = 0)$ da uchramaydigan hadlar uchun $P(X = t_i|c)\ tf_{ti,d} = 1$ va shunday hadda: $tf_{ti,d} \geq 1$ marta sodir bo'lsa, (2.2) tenglamada ham (2.15) tenglamada ham $tf_{ti,d}$ omillar hissa qo'shadi.

SELECTFEATURES($\mathbb{D}, c, k$)
1 $V \leftarrow$ EXTRACTVOCABULARY($\mathbb{D}$)
2 $L \leftarrow []$
3 **for each** $t \in V$
4 **do** $A(t,t) \leftarrow$ COMPUTEFEATUREUTILITY($\mathbb{D}, t, c$)
5 APPEND($L, \langle A(t,c), t \rangle$)
6 **return** FEATURESWITHLARGESTVALUES(L, k)

2.6- rasm. **K eng yaxshi xususiyatni tanlash uchun asosiy xususiyatni tanlash algoritmi**

Misol. 2.5- jadvaldagi hujjatlarning qaysi birida (i) Bernulli modeli (ii) multinomial model uchun bir xil va turli xil so'zlar to'plami mavjud? Agar farqlar bo'lsa, ularni tavsiflang.

Misol. Pozitsiyaviy mustaqillik taxminining asosi shundan iboratki, hujjatning k pozitsiyasida atama paydo bo'lishida foydali ma'lumotlar yo'q. Istisnolarni toping. Ruxsat etilgan hujjat tuzilishiga ega formulali hujjatlarni ko'rib chiqing.

Misol. 2.3- jadvalda so'z uchun Bernoulli va multinomial baholar berilgan. Farqni tushuntiring.

2.5. Xususiyatlarni tanlash

Xususiyatlarni tanlash - o'quv to'plamida uchraydigan atamalar to'plamini tanlash va matn tasnifida faqat shu kichik to'plamdan xususiyatlar sifatida foydalanish jarayonidir. Xususiyatlarni tanlash ikkita asosiy maqsadga xizmat qiladi. Birinchidan, lug'at o'qitish va klassifikatorni qo'llashni samaraliroq hajmini kamaytirish orqali samaraliroq qiladi. Bu NB dan farqli o'laroq, o'qitish qimmat bo'lgan tasniflagichlar uchun alohida ahamiyatga ega. Ikkinchidan, xususiyatni tanlash ko'pincha shovqin xususiyatlarini yo'q qilish orqali tasniflash aniqligini oshiradi. Shovqin xususiyati hujjat ko'rinishiga qo'shilganda yangi ma'lumotlarda tasniflash xatosini oshiradi. Aytaylik, kamdan-kam uchraydigan atama, aytaylik, araxnosentrik, hech qanday ma'lumotga ega emas.

Sinf deylik Xitoy lekin araxnosentrikning barcha holatlari ularning o'quv majmuamizdagi Xitoy hujjatlarida uchraydi. Keyin o'rganish usuli Xitoyga araxnosentrik bo'lgan test hujjatlarini noto'g'ri tayinlaydigan tasniflagichni ishlab chiqishi mumkin. Ta'limning tasodifiy xususiyatidan bunday noto'g'ri umumlashtirish to'plami ***ortiqcha moslama*** deb ataladi. Xususiyat tanlashni murakkab klassifikatorini (barcha xususiyatlardan foydalangan holda) oddiyroq (xususiyatlarning kichik to'plamidan foydalangan holda) bilan almashtirish usuli sifatida ko'rishimiz mumkin.

Statistik matnni tasniflashda zaifroq ko'rinadigan klassifikatorning foydali ekanligi dastlab qarama-qarshi bo'lib tuyulishi mumkin, ammo 3.6- bo'limda ikki tomonlama farqni muhokama qilganda, cheklangan o'quv ma'lumotlari mavjud bo'lganda, ko'pincha zaifroq modellar afzalligini ko'ramiz. Asosiy xususiyatni tanlash algoritmi 2.6- rasmda ko'rsatilgan. Berilgan **c** sinf uchun ***lug'at***ning har bir a'zosi uchun foydali o'lchov **A(t,c)** hisoblab chiqamiz va A(t, c) ning eng yuqori qiymatiga ega bo'lgan k atamani tanlaymiz. Boshqa barcha atamalar bekor qilinadi va tasniflashda ishlatilmaydi. Ushbu bo'limda uch xil foydali chora-tadbirlarni kiritamiz: o'zaro ma'lumot, $\mathbf{A(t,c) = I(U_t; C_c)}$; ch2 testi, $\mathbf{A(t,c) = X_2(t,c)}$; va chastota, **A(t,c)=N(t,c).** Ikki NB modelidan Bernoulli modeli shovqin xususiyatlariga ayniqsa sezgir. Bernoulli NB klassifikatori xususiyat tanlashning qandaydir shaklini talab qiladi aks holda uning aniqligi past bo'ladi. Ushbu bo'lim, asosan, Xitoy va Xitoy bo'lmaganlar kabi ikki toifali tasniflash vazifalari uchun xususiyatlarni tanlashga

qaratilgan. 2.5.5- bo'limda ikkitadan ortiq sinfga ega tizimlar uchun optimallashtirish qisqacha muhokama qilinadi.

2.5.1. O'zaro ma'lumotlar

Xususiyatlarni tanlashning umumiy usuli **A(t,c)** ni **t** atamasi va **c** sinfining kutilgan o'zaro ma'lumoti (**MI**) sifatida hisoblashdir. MI atamaning mavjudligi/yo'qligi **c** bo'yicha to'g'ri tasniflash qarorini qabul qilishga qancha ma'lumot berishini o'lchaydi. Rasmiy ravishda quyidagicha hisoblanadi:

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) \log_2 \frac{P(U = e_t, C = e_c)}{P(U = e_t)P(C = e_c)}, \qquad (2.16)$$

Bu yerda $e_t = 1$ (hujjatda **t** atamasi mavjud) va $e_t = 0$ (hujjatda **t** mavjud emas) qiymatlarini oladigan tasodifiy o'zgaruvchidir, **C** esa e_c=1 qiymatlarini qabul qiluvchi tasodifiy o'zgaruvchidir (hujjat **c** sinfida) va e_c=0 (hujjat **c** sinfida emas). Kontekstdan qaysi **t** atamasi va **c** sinfini nazarda tutayotganimiz aniq bo'lmasa, **Ut** va C_c deb yozamiz. Ehtimolliklarning **MLE** ko'rinishida (2.16) tenglama (2.17) tenglamaga teng:

$$I(U;C) = \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_1 N_{\cdot 1}} + \frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_0 N_{\cdot 1}} + \frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_1 N_{\cdot 0}} + \frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_0 N_{\cdot 0}} \qquad (2.17)$$

Bu yerda **N** lar e_t va e_c qiymatlariga ega bo'lgan hujjatlar soni bo'lib, ikkita pastki belgisi bilan ko'rsatilgan. Misol uchun, N_{10} - **t(e_t = 1)** ni o'z ichiga olgan va **c(e_c=0)** da bo'lmagan hujjatlar soni. $N_1 = N_{10} + N_{11}$ - **t(e_t=1)** ni o'z ichiga olgan hujjatlar soni va hujjatlarni sinfga a'zolikdan mustaqil ravishda hisoblaymiz ($e_c \in \{0,1\}$). $N = N_{00}+N_{01}+N_{10}+N_{11}$ - hujjatlarning umumiy soni. (2.16) tenglamani (2.17) tenglamaga aylantiruvchi **MLE** baholaridan biriga misol $\mathbf{P(U = 1, C = 1) = N11 /N.}$

Misol. Reuters-RCV1 da parrandachilik sinfini va eksport atamasini ko'rib chiqing. Ko'rsatkich qiymatlarining to'rtta mumkin kombinatsiyasi bo'lgan hujjatlar sonining umumiy soni quyidagicha hisoblanadi:

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$
$e_t = e_{export} = 1$	$N_{11} = 49$	$N_{10} = 27{,}652$
$e_t = e_{export} = 0$	$N_{01} = 141$	$N_{00} = 774{,}106$

$$I(U;C) = \frac{49}{801{,}948}\log_2\frac{801{,}948 \cdot 49}{(49 + 27{,}652)(49 + 141)} + \frac{141}{801{,}948}\log_2\frac{801{,}948 \cdot 141}{(141 + 774{,}106)(49 + 141)} + \frac{27{,}652}{801{,}948}\log_2\frac{801{,}948 \cdot 27{,}652}{(49 + 27{,}652)(27{,}652 + 774{,}106)} + \frac{774{,}106}{801{,}948}\log_2\frac{801{,}948 \cdot 774{,}106}{(141 + 774{,}106)(27{,}652 + 774{,}106)}$$

$$\approx 0.0001105$$

k shartni tanlash uchun t_1, ..., t_k berilgan sinf uchun 2.6- rasmdagi xususiyatni tanlash algoritmidan foydalaniladi: Foydali o'lchovni **A(t,c)=I(U_t,C_c)** ko'rinishida hisoblaymiz va eng katta qiymatga ega bo'lgan **k** shartni tanlaymiz. O'zaro ma'lumot atama sinf haqida qancha ma'lumot mavjudligini o'lchaydi - nazariy ma'noda. Agar atamaning taqsimlanishi butun to'plamdagi kabi sinfda ham bir xil bo'lsa, u holda **I(U; C) =0**. Agar atama sinfga a'zolik uchun mukammal ko'rsatkich bo'lsa, ya'ni hujjat faqat hujjat sinfidan iborat bo'lsa, **MI** o'zining maksimal qiymatiga yetadi.

2.7- rasmda 2.1- rasmdagi oltita sinf uchun yuqori o'zaro ma'lumotga ega bo'lgan atamalar ko'rsatilgan. Tanlangan atamalar (masalan, Buyuk Britaniya sinfi uchun london, uk, britaniya) o'z sinflari uchun tasniflash qarorlarini qabul qilish uchun aniq foydalidir. Buyuk Britaniya uchun ro'yxatning pastki qismida tashqi qurilmalar va bu kecha (rasmda ko'rsatilmagan) kabi atamalarni topamiz, hujjat sinfda so' bor yoki yo'qligini aniqlashda aniq yordam bermaydi. Siz kutganingizdek, ma'lumot beruvchi shartlarni saqlash va ma'lumotga ega bo'lmaganlarini yo'q qilish shovqinni kamaytiradi va tasniflagichning aniqligini oshiradi.

Xususiyat ballari dastlabki 100000 hujjat bo'yicha hisoblab chiqilgan, parranda go'shtidan tashqari, 800000 hujjat ishlatilgan noyob sinf. O'nta ro'yxatdagi raqamlar va boshqa maxsus so'zlarni o'tkazib yuborish mumkin.

UK	
london	0.1925
uk	0.0755
british	0.0596
stg	0.0555
britain	0.0469
plc	0.0357
england	0.0238
pence	0.0212
pounds	0.0149
english	0.0126

China	
china	0.0997
chinese	0.0523
beijing	0.0444
yuan	0.0344
shanghai	0.0292
hong	0.0198
kong	0.0195
xinhua	0.0155
province	0.0117
taiwan	0.0108

poultry	
poultry	0.0013
meat	0.0008
chicken	0.0006
agriculture	0.0005
avian	0.0004
broiler	0.0003
veterinary	0.0003
birds	0.0003
inspection	0.0003
pathogenic	0.0003

coffee	
coffee	0.0111
bags	0.0042
growers	0.0025
kg	0.0019
colombia	0.0018
brazil	0.0016
export	0.0014
exporters	0.0013
exports	0.0013
crop	0.0012

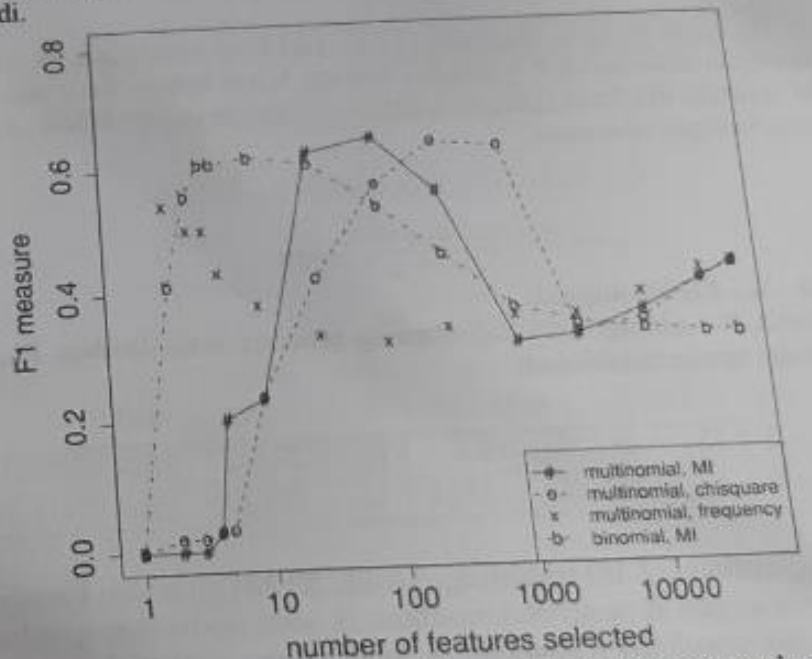
elections	
election	0.0519
elections	0.0342
polls	0.0339
voters	0.0315
party	0.0303
vote	0.0299
poll	0.0225
candidate	0.0202
campaign	0.0202
democratic	0.0198

sports	
soccer	0.0681
cup	0.0515
match	0.0441
matches	0.0408
played	0.0388
league	0.0386
beat	0.0301
game	0.0299
games	0.0284
team	0.0264

2.7- rasm. **Oltita Reuters-RCV1 sinflari uchun yuqori, o'zaro ma'lumot ballariga ega xususiyatlar**

Bunday aniqlikning o'sishini Reuters-RCV1 uchun xususiyat tanlagandan so'ng F_1 lug'at hajmining funksiyasi sifatida ko'rsatilgan 2.8- rasmda kuzatish mumkin. F_1 ni 132-776 xususiyatda (barcha funksiyalar tanloviga mos keladi) va 10–100 ta xususiyatda solishtirsak, MI funksiyasini tanlash F_1 ni ko'p nomli model uchun taxminan 0,1 ga va **Bernulli modeli** uchun 0,2 dan ortiq oshirishini ko'ramiz. Bernoulli modeli uchun F_1 cho'qqisiga o'nta xususiyat tanlangan. O'sha paytda Bernulli modeli multinomial modeldan yaxshiroqdir. Tasniflash qarorini faqat bir nechta xususiyatga asoslaganda faqat ikkilik hodisani hisobga

olish yanada ishonchli bo'ladi. Ko'p nomli model (MI xususiyatini tanlash) uchun cho'qqi keyinroq 100 ta xususiyatda sodir bo'ladi va barcha xususiyatlardan foydalanganda uning samaradorligi oxirida biroz tiklanadi.



2.8- rasm. **Ko'p nomli va Bernulli modellari uchun xususiyatlar to'plami o'lchamining aniqlikka ta'siri**

Buning sababi, multinomial parametrlarni baholash va tasniflashda hodisalar sonini hisobga oladi va shuning uchun Bernulli modeliga qaraganda ko'proq xususiyatlardan yaxshiroq foydalanadi. Ikkala usul o'rtasidagi farqlardan qat'i nazar, diqqat bilan tanlangan xususiyatlar to'plamidan foydalanish barcha xususiyatlardan foydalanishdan ko'ra yaxshiroq samaradorlikka olib keladi.

2.5.2. X^2 xususiyatlarni tanlash usuli

Yana bir mashhur xususiyatni tanlash usuli X^2 dir. Statistikada X^2 testi ikkita hodisaning mustaqilligini tekshirish uchun qo'llaniladi. Bunda ikkita **A** va **B** hodisalari mustaqil deb aniqlanadi, agar $P(AB) = P(A)\,P(B)$ yoki ekvivalenti $P(A|B)=P(A)$ va $P(B|A)=P(B)$. Xususiyatlarni tanlashda

ikkita hodisa atamasining paydo bo'lishi va sinfning paydo bo'lishidir. Keyin shartlarni quyidagi miqdorga qarab ajratamiz:

$$X^2(D,t,c) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} \frac{(Ne_te_c - Ee_te_c)^2}{Ee_te_c} \qquad (2.18)$$

Bu yerda e_t va e_c tenglama (2.16) dagi kabi aniqlanadi. **N - D** da kuzatilgan chastota va **E** kutilgan chastota. Misol uchun, E_{11} - atama va sinf mustaqil deb faraz qilingan hujjatda birgalikda yuzaga keladigan **t** va **c** ning kutilgan chastotasi.

$$E_{11} = N \times P(t) \times P(c) = N \times \frac{N_{11} + N_{01}}{N} \times \frac{N_{11} + N_{01}}{N}$$

$$= N \times \frac{49 + 141}{N} \times \frac{49 + 27652}{N} = 6.6$$

Misol. A_{ave} E_{11} hisoblanadi.

Bu yerda **N** - oldingi kabi hujjatlarning umumiy soni. Boshqa E_{etec} ni xuddi shu tarzda hisoblanadi:

	$e_{poultry} = 1$	$e_{poultry} = 0$
$e_{export} = 1$	$N_{11} = 49 \quad E_{11} \approx 6.6$	$N_{10} = 27{,}652 \quad E_{10} \approx 27{,}694.4$
$e_{export} = 0$	$N_{01} = 141 \quad E_{01} \approx 183.4$	$N_{00} = 774{,}106 \quad E_{00} \approx 774{,}063.6$

Ushbu qiymatlarni (2.18) tenglamaga kiritib, X^2 284 qiymatini bermiz:

X^2 - kutilgan **E** soni va kuzatilgan **N** soni bir-biridan qanchalik og'ishini ko'rsatadigan o'lchovdir. X^2 ning yuqori qiymati kutilgan va kuzatilgan hisoblar o'xshashligini bildiruvchi mustaqillik gipotezasi noto'g'ri ekanligini ko'rsatadi. $X^2 \approx 284 > 10{,}83$ bo'ladi, 2.6- jadvalga asoslanib, parrandachilik va eksport mustaqil ekanligi haqidagi gipotezani faqat 0,001 xatolik ehtimoli bilan rad etishimiz mumkin. Shunga o'xshab, $X^2 \approx 284 > 10{,}83$ natijasi 0,001 darajasida statistik ahamiyatga ega ekanligini aniqlangan. Agar ikkita hodisa bog'liq bo'lsa, unda atamaning paydo bo'lishi sinfning paydo bo'lish ehtimolini oshiradi shuning uchun u xususiyat sifatida foydali bo'lishi kerak.

Bu X^2 xususiyatni tanlashning asosidir. X^2 ni hisoblashning arifmetik jihatdan sodda usuli quyidagicha:

$$X^2(D,t,c) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} \frac{(Ne_te_c - Ee_te_c)^2}{Ee_te_c} \approx 284$$

Bu (2.18) tenglamaga ekvivalentdir degan xulosani chiqarish mumkin chunki agar ikkita hodisa mustaqil bo'lsa, $X^2 \sim X^2$ bo'ladi, bu yerda X^2 ning taqsimotidir.

2.6- jadval. **Bir darajadagi erkinlik bilan X^2 taqsimotining kritik qiymatlari**

p	χ^2 critical value
0.1	2.71
0.05	3.84
0.01	6.63
0.005	7.88
0.001	10.83

Masalan, agar ikkita hodisa mustaqil bo'lsa, u holda $P(X^2 > 6{,}63) < 0{,}01$. Shunday qilib, $X^2 > 6.63$ uchun mustaqillik taxminini 99% ishonch bilan rad etish mumkin.

$$X^2(D,t,c) = \frac{(N_{11} + N_{10} + N_{01} + N_{00}) \times (N_{11}N_{00} - N_{10}N_{01})^2}{(N_{11} + N_{01}) \times (N_{11} + N_{10}) \times (N_{10} + N_{00}) \times (N_{01} + N_{00})} \qquad (2.19)$$

X^2 ni xususiyatni tanlash usuli sifatida baholash Statistik nuqtai nazardan X^2 xususiyatni tanlash muammosi deyish mumkin. Bir darajadagi erkinlik bilan sinov uchun *Yates* tuzatish deb ataladigan narsadan foydalanish kerak (2.7- bo'lim), bu statistik ahamiyatga ega bo'lishni qiyinlashtiradi. Bundan tashqari, statistik test bir necha marta ishlatilsa, kamida bitta xatoga yo'l qo'yish ehtimoli ortadi. Har birida 0,05 xatolik ehtimoli bo'lgan 1000 ta gipoteza rad etilsa, testning **0,05 × 1000 = 50** chaqiruvi o'rtacha noto'g'ri bo'ladi. Biroq, matn tasnifida xususiyatlar to'plamiga bir nechta qo'shimcha shartlar qo'shilishi yoki undan olib tashlanishi kamdan-kam hollarda ahamiyatga ega. Aksincha, xususiyatlarning nisbiy ahamiyati muhimdir. Modomiki X^2 xususiyatni tanlash faqat xususiyatlarni ularning foydaliligiga qarab tartiblaysa va o'zgaruvchilarning statistik bog'liqligi yoki mustaqilligi to'g'risida xabar berish uchun foydalanilmaydi, uning statistik nazariyaga qat'iy amal qilmasligidan ortiqcha tashvishlanish shart emas.

2.5.3. Chastotaga asoslangan xususiyatni tanlash

Uchinchi xususiyatni tanlash usuli - bu chastotaga asoslangan xususiyatni tanlash ya'ni sinfda eng keng tarqalgan atamalarni tanlashdir. Chastotani hujjat chastotasi (**c** toifasidagi **t** atamasini o'z ichiga olgan hujjatlar soni) yoki yig'ish chastotasi (**c** dagi hujjatlarda uchraydigan **t** belgilari soni) sifatida aniqlash mumkin. Hujjat chastotasi ***Bernoulli modeli*** uchun ko'proq mos keladi, yig'ish chastotasi ***multinomial model*** uchun moslashtirilgan. Chastotaga asoslangan xususiyatni tanlash sinf haqida aniq ma'lumotga ega bo'lmagan ba'zi tez-tez uchraydigan atamalarni tanlaydi masalan, haftaning kunlari (dushanba, seshanba, ...), ***newswire*** matnida sinflar bo'ylab tez-tez uchraydi. Minglab xususiyatlar tanlansa, chastotaga asoslangan funksiya tanlash ko'pincha yaxshi natija beradi. Shunday qilib, agar biroz ***suboptimal aniqlik*** maqbul bo'lsa, chastotaga asoslangan xususiyatni tanlash murakkabroq usullarga yaxshi alternativ bo'lishi mumkin. Biroq, 2.8- rasmda chastotaga asoslangan xususiyatni tanlash ***MI*** va X^2 dan ancha yomonroq bo'lgan va foydalanilmasligi kerak bo'lgan holatdir.

2.5.4. Ko'p tasniflagichlar uchun xususiyat tanlash

Ko'p sonli klassifikatorga ega bo'lgan operatsion tizimda har bir klassifikator uchun boshqa xususiyatlar o'rniga bitta xususiyatlar to'plamini tanlash maqsadga muvofiqdir. Bunday usullardan biri n×2 jadval uchun X^2 statistikasini hisoblashdir. Bu yerda ustunlar atamaning paydo bo'lishi va bo'lmasligi va har bir satr sinflardan biriga mos keladi. Keyin avvalgidek eng yuqori X^2 statistikasi bilan **k** shartni tanlash mumkin. Odatda, xususiyatlarni tanlash statistikasi birinchi navbatda **c** va **c** ***ikki sinfli*** tasniflash vazifasi bo'yicha har bir sinf uchun alohida hisoblanadi va keyin birlashtiriladi. Bitta kombinatsiyalangan usul har bir xususiyat uchun bitta yutuq ko'rsatkichini hisoblab chiqadi masalan, **t** xususiyati uchun **A(t,c)** qiymatlarini o'rtacha hisoblab, so'ngra eng yuqori ko'rsatkichlarga ega bo'lgan **k** xususiyatni tanlaydi. Yana bir tez-tez qo'llaniladigan kombinatsiyalash usuli **n** ta tasniflagichning har biri uchun eng yuqori **k/n** xususiyatlarini tanlaydi va keyin bu **n** to'plamni bitta global xususiyatlar to'plamiga birlashtiradi. Tasniflash aniqligi ko'pincha **k** o'lchamdagi **n** xil to'plamdan farqli ravishda **n** ta

tasniflagichga ega tizim uchun **k** umumiy xususiyatni tanlashda pasayadi. Ammo shunday bo'lsa ham, umumiy hujjat taqdimoti tufayli samaradorlikning oshishi aniqlikdagi yo'qotishlarga arziydi.

2.5.5. Xususiyatlarni tanlash usullarini solishtirish

O'zaro ma'lumot va X^2 bir-biridan farq qiluvchi xususiyat tanlash usullarini ifodalaydi. **t** atamasi va **c** ***sinfining mustaqilligi*** ba'zida **t** hujjatning **c** ga a'zoligi haqida kam ma'lumotga ega bo'lsa ham yuqori ishonch bilan rad etilishi mumkin. Bu ayniqsa, noyob atamalar uchun to'g'ri keladi. Agar atama katta to'plamda bir marta bo'lsa va bu hodisa parrandalar sinfida bo'lsa bu statistik ahamiyatga ega. Lekin birgina hodisa axborotning informatsion-nazariy ta'rifiga ko'ra unchalik informatsion emas. Uning mezoni ahamiyatlilik bo'lganligi sababli X^2 o'zaro ma'lumotlarga qaraganda ko'proq nodir atamalarni (ko'pincha kamroq ishonchli ko'rsatkichlar) tanlaydi.

Ammo o'zaro ma'lumotni tanlash mezoni ham tasniflashning aniqligini maksimal darajada oshiradigan shartlarni tanlashi shart emas. Ikkala usul o'rtasidagi farqlarga qaramay, X^2 va **MI** bilan tanlangan xususiyatlar to'plamining tasniflash aniqligi tizimli ravishda farq qilmaydi. Ko'pgina matnlarni tasniflash muammolarida bir nechta kuchli ko'rsatkichlar va ko'plab zaif ko'rsatkichlar mavjud. Barcha kuchli ko'rsatkichlar va ko'p sonli zaif ko'rsatkichlar tanlanganligi sababli aniqlik yaxshi bo'lishi kutilmoqda. Buni ikkala usul ham bajaradi. 2.8-rasmda ***multinomial model*** uchun **MI** va X^2 xususiyat tanlash solishtiriladi.

Yuqori samaradorlik ikkala usul uchun ham deyarli bir xil. X^2 bu cho'qqiga keyinroq 300 ta xususiyatga erishadi, ehtimol u tanlagan noyob lekin juda muhim xususiyatlar dastlab sinfdagi barcha hujjatlarni qamrab olmagani uchundir. Biroq keyinroq tanlangan xususiyatlar (100-300 oralig'ida) **MI** tomonidan tanlanganlarga qaraganda yaxshiroq sifatga ega. Barcha uchta usul - **MI**, X^2 va chastotaga asoslangan usullardir. Avval tanlangan funksiyalarga nisbatan qo'shimcha ma'lumot bermaydigan xususiyatlarni tanlashi mumkin. 2.7- rasmda kong yettinchi atama sifatida tanlangan, garchi u ilgari tanlangan hong bilan ortiqcha bo'lsa ham yuqori darajada korrelyatsiya qilingan. Garchi bunday ortiqchalik aniqlikka salbiy ta'sir ko'rsatishi mumkin bo'lsa-da, yaxshi

usullar (2.7- bo'lim) hisoblash xarajatlari tufayli matn tasnifida kamdan-kam qo'llaniladi.

Misol. *Reuters-RCV1* ning dastlabki 100000 hujjatlarida to'rt muddat uchun qahva sinfi uchun quyidagi chastotalarni ko'rib chiqing:

term	N_{00}	N_{01}	N_{10}	N_{11}
brazil	98,012	102	1835	51
council	96,322	133	3525	20
producers	98,524	119	1118	34
roasted	99,824	143	23	10

(i) X^2, (ii) o'zaro ma'lumot, (iii) chastota asosida ushbu to'rtta atamadan ikkitasini tanlang.

2.6. Matn tasnifini baholash

Tarixiy jihatdan klassik *Reuters-21578* to'plami matn tasnifini baholash uchun asosiy mezon bo'lgan. Bu *CONSTRUE* matn tasniflash tizimini ishlab chiqish jarayonida **Carnegie Group, Inc.** va **Reuters, Ltd.** tomonidan dastlab to'plangan va etiketlangan 21,578 newswire maqolalari to'plamidir. U I- qismning 4-bobida muhokama qilingan *Reuters-RCV1* to'plamidan ancha kichikroq va undan oldinroq.

Maqolalar 118 ta mavzu toifasidan iborat sinflarga ajratilgan. Hujjatga bir nechta sinflar berilishi mumkin yoki hech kim bo'lmasligi mumkin, lekin eng keng tarqalgan holat bitta topshiriqdir (kamida bitta sinfga ega hujjatlar o'rtacha 1,24 sinf olgan).

Ushbu har qanday muammoga standart yondashuv (3-bob) har bir sinf uchun bittadan 118 ta ikki sinfli klassifikatorni o'rganishdir. Bunda **c** sinf uchun ikki sinfli klassifikator ikki sinf **c** klassifikatori va uning to'ldiruvchisi hisoblanadi.

Ushbu tasniflagichlarning har biri uchun ***eslab qolish***, ***aniqlik*** va ***aniqlikni o'lchash*** mumkin. So'nggi ishlarda odamlar deyarli har doim *ModApte* bo'linmasidan foydalanadi, bu faqat inson indeksi tomonidan ko'rilgan va baholangan hujjatlarni o'z ichiga oladi va 9,603 ta o'quv hujjatlari va 3,299 ta test hujjatlarini o'z ichiga oladi. Hujjatlarni sinflarda taqsimlash juda notekis bo'lib, ba'zi ishlarni ya'ni eng katta o'nta sinfdagi faqat hujjatlar bo'yicha tizimlarni baholaydi. Ular 2.7- jadvalda keltirilgan.

Mavzular bilan odatiy hujjat 2.9- rasmda ko'rsatilgan.

2.7- jadval. **Reuters-21578** to'plamidagi eng katta o'nta sinf o'quv va test to'plamlaridagi hujjatlar soni.

class	# train	# testclass	# train	# test
earn	2877	1087 *trade*	369	119
acquisitions	1650	179 *interest*	347	131
money-fx	538	179 *ship*	197	89
grain	433	149 *wheat*	212	71
crude	389	189 *corn*	182	56

Hujjatlarni sinflarga taqsimlash juda notekis va ba'zi ishlarda tizimlar faqat eng katta o'nta sinfdagi hujjatlar bo'yicha baholanadi. Ular 2.7-jadvalda keltirilgan. Mavzular bilan odatiy hujjat 2.9- rasmda ko'rsatilgan. 2.1- bo'limda ular matnni tasniflashda test ma'lumotlaridagi tasniflash xatosini minimallashtirishni maqsad qilingan. Tasniflash xatosi - 1,0 minus tasniflash aniqligi, to'g'ri qarorlar nisbati, ular 1- qismning 8.3-bo'limiga kiritilgan o'lchovdir. Agar sinfdagi hujjatlar ulushi yuqori bo'lsa, ehtimol 10% dan 20% gacha va undan yuqori bo'lsa, bu o'lchov mos keladi. Ammo avvalgi qismning 8.3- bo'limida muhokama qilganimizdek, "kichik" sinflar uchun aniqlik yaxshi o'lchov emas, chunki har doim "yo'q" xabarini beradi, klassifikatorni yaratish maqsadini buzadigan strategiya yuqori aniqlikka erishadi. Har doim yo'q tasniflagichi nisbiy chastotasi 1% bo'lgan sinf uchun 99% aniq. Kichik sinflar uchun aniqlik eslab qolish va F_1 yaxshiroq o'lchovlardir. Ular samaradorlikni tasniflash qarorlari sifatini, jumladan, aniqlik, eslab qolish, F_1 va aniqlikni baholovchi chora-tadbirlar uchun umumiy atama sifatida foydalaniladi. Ishlash ushbu kitobdagi tasniflash va AQ tizimlarining hisoblash samaradorligini anglatadi. Biroq, ko'pgina tadqiqotchilar ishlash atamasini ishlatganda matn tasnifining samaradorligini anglatadi. To'plamni bir nechta ikki sinfli klassifikatorlar bilan qayta ishlanganda (masalan, ***Reuters-21578118*** sinfga ega), ular ko'pincha individual tasniflagichlar uchun o'lchovlarni birlashtirgan yagona agregat o'lchovni hisoblash mumkin. Buni amalga oshirishning ikkita usuli mavjud: Makro qoralama sinflar bo'yicha oddiy o'rtachani hisoblaydi; Mikro o'rtacha har bir hujjat bo'yicha qarorlarni sinflar bo'yicha to'playdi va keyin birlashtirilgan favqulodda vaziyatlar jadvalida samaradorlik o'lchovini hisoblaydi. 2.8- jadvalda misol keltirilgan. Ikki usul o'rtasidagi farqlar katta bo'lishi mumkin. Makro

o'rtacha har bir sinfga teng og'irlik beradi, mikro o'rtacha esa har bir hujjat uchun tasniflash qaroriga teng og'irlik beradi. F_1 o'lchovi haqiqiy negativlarni e'tiborsiz qoldirganligi va uning kattaligi asosan haqiqiy musbatlar soni bilan belgilanadiganligi sababli mikroo'rtacha o'lchashda kichik sinflarda katta sinflar ustunlik qiladi. Misolda, mikroo'rtacha aniqlik **(0,83)** c_1**(0,5)** aniqligiga qaraganda c_2**(0,9)** aniqligiga yaqinroq, chunki c_2 c dan besh baravar katta bo'ladi.

1. Mikroo'rtacha natijalar, shuning uchun testlar to'plamidagi katta sinflar bo'yicha samaradorlik o'lchovidir. Kichik sinflarda samaradorlikni his qilish uchun siz makroo'rtacha natijalarni hisoblashingiz kerak. Bir tasnifda (14.5-bo'lim, 306-bet) mikroo'rtacha F

l aniqlik bilan bir xil (13.6-mashq)

```
<REUTERS TOPICS=''YES'' LEWISSPLIT=''TRAIN''
CGISPLIT=''TRAINING-SET'' OLDID=''12981'' NEWID=''798''>
<DATE> 2-MAR-1987 16:51:43.42</DATE>
<TOPICS><D>livestock</D><D>hog</D></TOPICS>
<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>
<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork
Congress kicks off tomorrow, March 3, in Indianapolis with 160
of the nations pork producers from 44 member states determining
industry positions on a number of issues, according to the
National Pork Producers Council, NPPC.
Delegates to the three day Congress will be considering 26
resolutions concerning various issues, including the future
direction of farm policy and the tax law as it applies to the
agriculture sector. The delegates will also debate whether to
endorse concepts of a national PRV (pseudorabies virus) control
and eradication program, the NPPC said. A large
trade show, in conjunction with the congress, will feature
the latest in technology in all areas of the industry, the NPPC
added. Reuter
\&\#3;</BODY></TEXT></REUTERS>
```

2.9- rasm. **Reuters-21578 to'plamidan namunaviy hujjat**

2.9-jadval *Reuters-21578* ning **ModApte** bo'linishi uchun *Naive Bayes*ning mikroo'rtacha va makroo'rtacha samaradorligini beradi. NB ning nisbiy samaradorligini his qilish uchun uni chiziqli ***SVM***lar bilan solishtiriladi (eng o'ng ustun; 4- bob), eng samarali tasniflagichlardan biri, ammo NBga qaraganda o'rgatish qimmatroqdir. NB mikroo'rtacha F_1 80% ga ega, bu SVM (89%) dan 9% kamroq, 10% nisbiy pasayish (qator "mikro-avg-L (90 sinf)"). Shunday qilib, uning soddaligi va samaradorligi uchun juda kichik samaradorlik pasayishi mavjud. Biroq kichik sinflarda ularning Ba'zilari o'quv majmuasida faqat o'nta ijobiy misolga ega bo'lsa, NB bundan ham yomonroq ishlaydi. Uning makroo'rtacha F_1 qiymati ***SVM*** dan **13%** past, nisbiy pasayish ***22%***

("makro-o'rtacha (90 sinf)" qatori). Jadvalda NB ni ushbu kitobda ko'rib chiqiladigan boshqa tasniflagichlar bilan solishtiriladi:

2.8- jadval. **Makro va mikroo'rtacha**

class 1	truth: yes	truth: no
call: yes	10	10
call: no	10	970

class 2	truth: yes	truth: no
call: yes	90	10
call: no	10	890

pooled table	truth: yes	truth: no
call: yes	100	20
call: no	20	1860

"Haqiqat" - bu haqiqiy sinf va tasniflagichning qarorini chaqiradi. Ushbu misolda makroo'rtacha aniqlik $[10/(10 + 10) + 90/(10 + 90)]/2 = (0,5 + 0,9)/2 = 0,7$. Mikroo'rtacha aniqlik $100/(100 + 20) \approx 0,83$.

Li va Yang (2003) (a), Joachims (1998) (b: kNN) va Dumais va boshqalar natijalari. (1998) (b: NB, Rocchio, daraxtlar, SVM).

Rokkio va kNN qaror daraxtlari uchun raqamlarni beradi, ular muhim raqamlarni qamrab olmaydigan muhim tasniflash usulidir. Jadvalning pastki qismi sinfdan sinfga sezilarli farq borligini ko'rsatadi. Misol uchun, NB to'plamda kNN ni ustun lekin *pul-fx* bo'yicha ancha yomonroq ishlaydi. Jadvalning (a) va (b) qismlarini solishtirganda, keltirilgan hujjatlarning natijalari qanchalik farq qilishiga hayron bo'lish mumkin. Bu qisman (b) dagi raqamlar zararsiz ballar 118 sinf bo'yicha o'rtacha hisoblanganligi, (a) dagi raqamlar esa haqiqiy F_1 ballari (test haqida hech qanday ma'lumotsiz hisoblangan) bilan bog'liq bo'lgan to'plam) o'rtacha to'qsondan ortiq sinfdir.

2.9-jadval. **F_1 uchun *Reuters-21578* dagi matn tasnifi samaradorligi raqamlari (foizda)**

(a)

	NB	Rocchio	kNN	SVM
micro-avg-L (90 classes)	80	85	86	89
macro-avg (90 classes)	47	59	60	60

(b)

	NB	Rocchio	kNN	trees	SVM
earn	96	93	97	98	98
acq	88	65	92	90	94
money-fx	57	47	78	66	75
grain	79	68	82	85	95
crude	80	70	86	85	89
trade	64	65	77	73	76
interest	65	63	74	67	78
ship	85	49	79	74	86
wheat	70	69	77	93	92
corn	65	48	78	92	90
micro-avg (top 10)	82	65	82	88	92
micro-avg-D (118 classes)	75	62	n/a	n/a	87

Bu afsuski, matn tasnifida turli natijalarni solishtirganda sodir bo'ladigan hodisalarga xosdir. Ko'pincha eksperimental o'rnatish yoki natijalarni sharhlashni qiyinlashtiradigan baholashda farqlar mavjud.

Bu va boshqa natijalar shuni ko'rsatdiki, NB ning o'rtacha samaradorligi mustaqil va bir xil taqsimlangan (i.i.d.) ma'lumotlardir, ya'ni statistik tanlamaning barcha yaxshi xususiyatlariga ega bir xil ma'lumotlar bo'yicha o'qitilgan va sinovdan o'tkazilganda ***SVM*** kabi tasniflagichlar bilan raqobatbardosh emas. Biroq, bu farqlar ko'pincha ko'rinmas yoki hatto teskari bo'lishi mumkin, bu yerda, odatda, o'quv namunasi tasniflagich qo'llaniladigan ma'lumotlarning kichik to'plamidan olinadi, ma'lumotlarning tabiati vaqt o'tishi bilan o'zgaradi. Statsionar ma'lumotlarda xatolar bo'lishi mumkin (boshqa muammolar qatorida). Ko'pgina amaliyotchilar doimiy ravishda NB dan yaxshiroq ishlaydigan ma'lum bir muammo uchun ajoyib tasniflagichni qura olmaganlik tajribasiga ega. 2.9-jadvaldagi natijalardan xulosa shundan iboratki, ko'pchilik tadqiqotchilar ***SVM kNN*** dan yaxshiroq va ***kNN NB*** dan yaxshiroq deb hisoblashsa ham, klassifikatorlarning reytingi oxir-oqibat sinfga, hujjatlar to'plamiga va eksperimental o'rnatishga bog'liq. Matnni tasniflashda faqat qaysi mashinani o'rganish algoritmi ishlatilganligini bilish muhim hisoblanadi.

2.9- jadvaldagi kabi baholashlarni amalga oshirayotganda, mashg'ulotlar to'plami va testlar to'plami o'rtasida qat'iy ajratishni saqlash muhimdir. Testlar to'plamidan olingan ma'lumotlardan foydalanib, testlar to'plami bo'yicha to'g'ri tasniflash qarorlarini osongina qabul qilish mumkin masalan, ma'lum bir atama testlar to'plamida yaxshi bashoratchi ekanligi (garchi bu o'quv to'plamida bunday bo'lmasa ham) ni keltirish mumkin. Test to'plami haqidagi bilimlardan foydalanishning yanada nozik misoli - bu parametrning ko'p sonli qiymatlarini sinab ko'rish (masalan, tanlangan xususiyatlar soni) va test to'plami uchun eng yaxshi qiymatni tanlashdir. Qoidaga ko'ra, yangi ma'lumotlarning aniqligi - klassifikatorni ilovada ishlatganda duch keladigan ma'lumotlar turlaridir – bular klassifikator sozlangan test to'plamidagi aniqlikdan ancha past bo'ladi. Xuddi shu muammoni 1-qismning 8.1- bo'limda maxsus qidiruvda ko'rib chiqildi. Matnni statistik tasniflash bo'yicha toza statistik tajribada matnni tasniflash tizimini ishlab chiqishda hech qachon hech qanday dasturni ishga tushirmaslik yoki hatto test to'plamiga qaramaslik kerak. Buning o'rniga, o'z uslubingizni ishlab chiqishda sinov uchun ishlab chiqish to'plamini ajratib qo'ying. Agar

bunday to'plam parametr uchun yaxshi qiymatni topishning asosiy maqsadiga xizmat qilsa, masalan, tanlangan xususiyatlar soni, u holda u ***ushlab turilgan ma'lumotlar*** deb ham ataladi. Ta'lim to'plamining qolgan qismiga klassifikatorni turli parametr qiymatlari bilan o'rgating, so'ngra o'quv to'plamining ushlab turilgan qismida eng yaxshi natijalarni beradigan qiymatni tanlang. Ideal holda, eng oxirida, barcha parametrlar o'rnatilgandan va usul to'liq ko'rsatilgandan so'ng, test to'plamida bitta yakuniy tajribani o'tkazish va natijalarni nashr qilish mumkin. Klassifikatorni ishlab chiqishda testlar to'plami haqida hech qanday ma'lumot ishlatilmaganligi sababli, ushbu tajriba natijalari amaldagi haqiqiy ishlashni ko'rsatishi kerak.

2.10- jadval. **Parametrlarni baholash mashqlari uchun ma'lumotlar**

	docID	words in document	in c = *China*?
training set	1	Taipei Taiwan	yes
	2	Macao Taiwan Shanghai	yes
	3	Japan Sapporo	no
	4	Sapporo Osaka Taiwan	no
test set	5	Taiwan Taiwan Sapporo	?

Ko'pincha bu idealga erishib bo'lmaydi. Tadqiqotchilar bir necha yil davomida bir xil test majmuasida bir nechta tizimlarni baholashi mumkin. Ammo shunga qaramay, test ma'lumotlariga qaramaslik va tizimlarni imkon qadar tejamkorlik bilan ishlatish juda muhimdir. Yangi boshlanuvchilar ko'pincha bu qoidani buzadilar va ularning natijalari o'z kuchini yo'qotadi chunki ular ko'plab variantli tizimlarni ishga tushirish va test to'plamida eng yaxshi ishlaydigan tizimga o'zgartirishlar kiritish orqali o'z tizimini test ma'lumotlariga sozlaydi.

Misol. Vaziyatni tasavvur qiling-a, testlar to'plamidagi har bir hujjatga aynan bitta sinf berilgan va klassifikator ham har bir hujjatga aynan bitta sinf tayinlagan. Ushbu o'rnatish bitta tasnif deb ataladi (3.5-bo'lim). Bir tasnifda (i) noto'g'ri ijobiy qarorlarning umumiy soni noto'g'ri salbiy qarorlarning umumiy soniga teng ekanligini va (ii) mikroo'rtacha $\mathbf{F_1}$ va aniqlik bir xil ekanligini ko'rsating.

Misol. 2.2- rasmdagi sinf ustunlari sinfdagi tokenlar ulushidan farqli ravishda sinfdagi hujjatlarning ulushi sifatida hisoblanadi. Nega? Qanday amalga oshirilishini ko'rsating.

Misol. 2.2- rasmdagi funksiya vaqt murakkabligi $D(L_n+|C|L_n)$ ga ega. Funksiyani vaqt murakkabligi $D(L_n+|C|M_n)$ bo'lishi uchun qanday o'zgartirgan bo'lardingiz? Qanday amalga oshirilishini ko'rsating.

Misol. 2.10- jadvaldagi ma'lumotlarga asoslanib, (i) ko'p nomli *Naive Bayes* klassifikatorini baholang, (ii) klassifikatorni test hujjatiga qo'llang, (iii) Bernoulli NB klassifikatorini baholang, (iv) klassifikatorni test hujjatiga qo'llang. Sinov hujjatini tasniflash uchun kerak bo'lmagan parametrlarni hisoblashingiz shart emas.

Misol. So'zlarni ingliz yoki ingliz emas ekanlini aniqlashni tasniflashdir. So'zlar quyidagi tarqatish bilan manba tomonidan yaratilgan:

event	word	English?	probability
1	ozb	no	4/9
2	uzu	no	4/9
3	zoo	yes	1/18
4	bun	yes	1/18

(i) xususiyat sifatida b, n, o, u va z harflaridan foydalanadigan ko'p nomli NB klassifikatorining parametrlari hisoblanadi. Manbaning ehtimollik taqsimotini mukammal aks ettiruvchi o'quv majmuasini tasavvur qiling. Odatda matn tasnifi uchun xususiyatlar sifatida atamalardan foydalanadigan ***multinomial klassifikator*** uchun qilingan mustaqillik haqidagi taxminlarni hosil qiling. Nolga teng bo'lgan hisoblangan ehtimollar 0,01 ehtimolga tenglashtirilgan va nolga teng bo'lmagan hisoblangan ehtimolliklarga tegmaydigan silliqlash yordamida parametrlarni hisoblash kerak. (Bu soddalashtirilgan tekislash **P(A) + P(A) > 1** ga olib kelishi mumkin. Buni tuzatish uchun yechimlar talab qilinmaydi. (ii) Klassifikator hayvonot bog'i so'zini qanday tasniflaydi? (iii) hayvonot bog'i so'zini ***multinomial klassifikator*** yordamida (i) qismdagi kabi tasniflang, lekin pozitsion mustaqillik farazini yaratmang. Ya'ni, so'zdagi har bir pozitsiya uchun alohida parametrlarni baholang. Faqat hayvonot bog'ini tasniflash uchun kerakli parametrlarni hisoblash kerak.

Misol. Agar atama va sinf butunlay mustaqil bo'lsa, $I(U_t; C_c)$ va $X^2(ID, t, c)$ ning qiymatlari qanday? Agar ular to'liq bog'liq bo'lsa, qiymatlar qanday bo'ladi?

Misol. (2.16) tenglamadagi xususiyatni tanlash usuli ***Bernulli modeli*** uchun eng mos keladi. Nega? Uni ***multinomial model*** uchun qanday o'zgartirish mumkin?

Misol. Xususiyatlar, shuningdek, ma'lumot olish (IG) ga qarab tanlanishi mumkin, bu quyidagicha aniqlanadi:

$$IG(D,t,c) = H(p_D) - \sum_{x\in\{D_t,\neg D_t\}} \frac{|x|}{|D|} H(p_x)$$

Bu yerda **H** - entropiya, **D** - o'quv to'plami va D_t + va ID_t - mos ravishda **t** hadli **D** to'plami va **t** hadsiz **D** to'plamidir. p_A - A (kichik) to'plamdagi sinf taqsimoti, masalan, $p_A(c) = 0{,}25$, $p_A(c) = 0{,}75$, agar **A** dagi hujjatlarning chorak qismi **c** sinfida bo'lsa. O'zaro ma'lumot va ma'lumot olish teng ekanligini ko'rsating.

Misol. Ikkita X^2 formulasi ((2.18) va (2.19) tenglamalar) ekvivalent ekanligini ko'rsating.

Misol. X^2 misolida $|N_{11} - E_{11}| = |N_{10} - E_{10}| = |N_{01} - E_{01}| = |N_{00} - E_{00}|$. Bu umuman amal qilishini ko'rsating.

Misol. X^2 va o'zaro ma'lumotlar ijobiy va salbiy korrelyatsiya qiluvchi xususiyatlarni ajratmaydi. Ko'pgina yaxshi matn tasniflash xususiyatlari ijobiy o'zaro bog'liq bo'lganligi sababli (ya'ni, ular **c** dan ko'ra tez-tez uchraydi), salbiy ko'rsatkichlarni tanlashni aniq istisno qilish kerak. Buni qanday qilgan bo'lardingiz?

Mavzuga doir adabiyotlar va ularning tahlili

Statistik tasniflash va mashinalarni o'rganish bo'yicha umumiy kirishlarni (***Hastie va boshq. 2001***), (***Mitchell 1997***) va (***Duda va boshq. 2000***), shu jumladan, ko'plab muhim usullarni (qarorlar daraxtlari va kuchaytirish) topish mumkin. Matnni tasniflash usullari va natijalarini har tomonlama ko'rib chiqish (***Sebastiani 2002***). ***Manning*** va ***Schutze*** (1999, 5-bob) qarorlar daraxti, perseptronlar va maksimal entropiya modellarini qamrab olgan holda matn tasnifiga kirish mumkin. Naive Bayesga qaraganda aniqroq bo'lgan o'rganish usullarining ***superlinear*** vaqt murakkabligi haqida ko'proq ma'lumotni (***Perkins*** va ***boshq. 2003***) va (***Joachims 2006***) topish mumkin.

Maron va ***Kuhns*** (1960) birinchi NB matn klassifikatorlaridan birini tasvirlab bergan. ***Lyuis*** (1998) asosiy e'tiborni NB tasnifi tarixiga qaratadi. Bernoulli va multinomial modellar va ularning turli to'plamlar uchun aniqligi ***McCallum*** va ***Nigam*** (1998) tomonidan muhokama qilinadi.

Eyheramendy va boshqalar(2003) qo'shimcha NB modellarini taqdim etadi.

Domingos va *Pazzani*(1997), *Fridman* (1997) va *Xand* va *Yu*(2001) NBning ehtimollik baholari yomon bo'lsa-da, nima uchun yaxshi ishlashini tahlil qiladi. Birinchi maqolada, shuningdek, mustaqillik haqidagi taxminlar ma'lumotlarga to'g'ri kelganda NBning optimalligi muhokama qilinadi.

Pavlov va boshqalar. (2004) mustaqillik haqidagi taxminlarning nomaqbulligini qisman hal qiluvchi o'zgartirilgan hujjat taqdimotini taklif qiladi. *Bennett* (2000) NB ehtimollik taxminlarining 0 yoki 1 ga yaqin bo'lish tendentsiyasini hujjat uzunligi ta'siri bilan bog'laydi. *Ng* va *Jordan* (2001) shuni ko'rsatadiki, NB ba'zan (kamdan-kam hollarda) diskriminatsion usullardan ustundir chunki u o'zining optimal xato darajasiga tezroq erishadi.

Ushbu bobda keltirilgan asosiy NB modeli yanada samaraliroq yoritilishi uchun (*Rennie* va boshq. 2003, *Kolcz* and *Yih* 2007). Kontseptsiyaning o'zgarishi muammosi va zamonaviy klassifikatorlarning amalda har doim ham ustun bo'lmasligining boshqa sabablari *Forman* (2006) va *Hand*(2006) tomonidan muhokama qilinadi.

Matnni tasniflashda o'zaro ma'lumot va X^2 dan xususiyatni tanlash uchun dastlabki foydalanish *Lyuis* va *Ringuette*(1994) va *Schutze* va boshqalar(1995) mos ravishda *Yang* va *Pedersen*(1997) xususiyatlarni tanlash usullarini va ularning tasniflash samaradorligiga ta'sirini ko'rib chiqadi.

Ular o'zaro ma'lumotlarning boshqa usullar bilan raqobatbardosh emasligini aniqlaydi. *Yang* va *Pedersen* kutilgan o'zaro ma'lumotga (Tenglama (2.16)) ma'lumot olish sifatida murojaat qiladi. (*Snedecor* va *Cochran* 1989) statistikada X^2 testi uchun yaxshi ma'lumotnoma, shu jumladan 2×2 jadvallar uchun *Yates*ning uzluksizligi uchun tuzatish kiritiladi.

*Dun*ning(1993) hisoblar kichik bo'lganda X^2 testining muammolarini muhokama qiladi. Navbatsiz xususiyatini tanlash usullari *Hastie* va boshqalar tomonidan tasvirlangan (2001)

. **Koen**(1995) ko'p ahamiyatlilik testlari va ulardan qochish usullarini qo'llashning tuzoqlarini muhokama qiladi.

Forman(2004) bir nechta tasniflagichlar uchun xususiyat tanlashning turli usullarini baholaydi.

Devid D., Lyuis ModApte bo'linishini www.daviddlewis.com/resources/ testcollections/reuters215 da *Apte* va boshqalarga asoslanib belgilaydi(1994).

Yang va *Liu*(1999) matnlarni tasniflash usullarini baholashda ahamiyatlilik testlaridan foydalanadi. *Lyuis* va boshqalar(2004) SVM (4-bob) ReutersRCV1 da kNN va Rocchio (3-bob) ga qaraganda yaxshiroq ishlashini aniqladi.

2- bob bo'yicha foydalanilgan adabiyotlar

Bar-Ilan, Judit, and Tatyana Gutman.
2005.
How do search engines respond to some non-English queries?
Journal of Information Science 31 (1): 13-28.

Bar-Yossef, Ziv, and Maxim Gurevich.
2006.
Random sampling from a search engine's index.
In *Proc. WWW*, pp. 367-376. ACM Press.
DOI: *doi.acm.org/10.1145/1135777.1135833*.

Barroso, Luiz André, Jeffrey Dean, and Urs Hölzle.
2003.
Web search for a planet: The Google cluster architecture.
IEEE Micro 23 (2): 22-28.
DOI: *dx.doi.org/10.1109/MM.2003.1196112*.

Bartell, Brian Theodore.
1994.
Optimizing ranking functions: A connectionist approach to adaptive information retrieval.
PhD thesis, University of California at San Diego, La Jolla, CA.

Hughes, Baden, Timothy Baldwin, Steven Bird, Jeremy Nicholson, and Andrew MacKinlay.
2006.
Reconsidering language identification for written language resources.
In *Proc. International Conference on Language Resources and Evaluation*, pp. 485-488.

Hull, David.
1993.
Using statistical testing in the evaluation of retrieval performance.
In *Proc. SIGIR*, pp. 329-338. ACM Press.

Hull, David.

1996.
Stemming algorithms - A case study for detailed evaluation.
JASIS 47 (1): 70-84.
Indyk, Piotr.
2004.
Nearest neighbors in high-dimensional spaces.
In J. E. Goodman and J. O'Rourkeeds.), *Handbook of Discrete and Computational Geometry*, 2nd edition, pp. 877-892. Chapman and Hall/CRC Press.
Ingwersen, Peter, and Kalervo Järvelin.
2005.
The Turn: Integration of Information Seeking and Retrieval in Context.
Springer.

2- bob bo'yicha nazariy va amaliy test savollari

1. Avtomatik indeksatsiya – nima?
A) Faqat rasmiy protseduralarni hisoblash uchun faqat rasmiy protseduralar yordamida hisoblash orqali amalga oshiriladigan hujjat yoki so'rovni qayta ishlash uchun
B) Texnologiyalar hisoblash texnologiyasi yordamida amalga oshirilayotgan rasmiy protseduralardan foydalanishni ko'zda tutadigan va qidiruv imidjini tuzish bo'yicha asosiy qarorlar qabul qilishda intellektual protseduralardan foydalanishni o'z ichiga olishi mumkin
C) Hujjat yoki so'rovning matnidagi so'z yoki ibora, unda muhim semantik yukni olib boradi
D) To'g'ri javob yo'q

2. Faqat rasmiy protseduralarni hisoblash uchun faqat rasmiy protseduralar yordamida hisoblash orqali amalga oshiriladigan hujjat yoki so'rovni qayta ishlash uchun foydalanadigan jarayon bu nima?
A) Avtomatik indeksatsiya
B) To'g'ri javob yo'q
C) Informatsion so'z
E) Avtomatlashtirilgan indekslash

3. Informatsion so'z – bu nima?
A) Hujjat yoki so'rovning matnidagi so'z yoki ibora, unda muhim semantik yukni olib boradi

B) Faqat rasmiy protseduralarni hisoblash uchun faqat rasmiy protseduralar yordamida hisoblash orqali amalga oshiriladigan hujjat yoki so'rovni qayta ishlash uchun
C) Texnologiyalar hisoblash texnologiyasi yordamida amalga oshirilayotgan rasmiy protseduralardan foydalanishni ko'zda tutadigan va qidiruv imidjini tuzish bo'yicha asosiy qarorlar qabul qilishda intellektual protseduralardan foydalanishni o'z ichiga olishi mumkin
D) To'g'ri javob yo'q

4. Quyidagilarning qaysi biri hujjat yoki so'rovning matnidagi so'z yoki ibora, unda muhim semantik yukni olib boradi?
A) Informatsion so'z
B) Avtomatlashtirilgan indekslash
C) Avtomatik indeksatsiya
D) To'g'ri javob yo'q

5. Bepul indekslash – bu nima?
A) M-atnning ma'lumotli so'zlarini indekslashning maxsus lug'ati tavsiyalariga muvofiq almashtirishni ta'minlamaydi
B) Hujjat yoki so'rovning matnidagi so'z yoki ibora, unda muhim semantik yukni olib boradi
C) Faqat rasmiy protseduralarni hisoblash uchun faqat rasmiy protseduralar yordamida hisoblash orqali amalga oshiriladigan hujjat yoki so'rovni qayta ishlash uchun qo'llaniladi
D) To'g'ri javob yo'q

6. Indekslashning qaysi texnologiyasi matnning ma'lumotli so'zlarini indekslashning maxsus lug'ati tavsiyalariga muvofiq almashtirishni ta'minlamaydi?
A) Bepul indekslash
B) Informatsion so'z
C) Avtomatlashtirilgan indekslash
D) Avtomatik indeksatsiya

7. Leksik birlik (LE) IPI – bu nima?
A) Belgilar ketma-ketligi, so'z, iboralar, so'zma bo'lak yoki belgi, ma'lum bir kontseptsiya, obyekti yoki parametrning qiymati uchun hujjatlar yoki so'rovlarni ifodalash uchun ishlatiladigan so'z yoki belgi sifatida
B) matnning ma'lumotli so'zlarini indekslashning maxsus lug'ati tavsiyalariga muvofiq almashtirishni ta'minlamaydi

C) Hujjat yoki so'rovning matnidagi so'z yoki ibora, unda muhim semantik yukni olib boradi

D) Faqat rasmiy protseduralarni hisoblash uchun faqat rasmiy protseduralar yordamida hisoblash orqali amalga oshiriladigan hujjat yoki so'rovni qayta ishlash uchun qo'llaniladi

8. Belgilar ketma-ketligi, so'z, iboralar, so'zma bo'lak yoki belgi, ma'lum bir kontseptsiya, ob'ekti yoki parametrning qiymati uchun hujjatlar yoki so'rovlarni ifodalash uchun ishlatiladigan so'z yoki belgi nima deb ataladi?

A) Leksik birlik
B) Bepul indekslash
C) Informatsion so'z
D) Avtomatlashtirilgan indekslash

9. To'liqlikni indekslashga to'g'ri ta'rifni toping?

A) Hujjat miqdori va (yoki) so'rovning qidiruv shakli va (yoki) qidiruv shakli, shuningdek, qidiruv tizimiga kiritilgan aniq atamalar va matnda mavjud bo'lgan ma'lumotlar soniga kiritilgan hujjat yoki so'rov

B) Izlash sifati va sifatli ma'lumotlarning sonini qidirish mexanizmidagi ma'lum shartlar soniga nisbati bilan

C) Faqat rasmiy protseduralarni hisoblash uchun faqat rasmiy protseduralar yordamida hisoblash orqali amalga oshiriladigan hujjat yoki so'rovni qayta ishlash uchun qo'llaniladi

D) To'g'ri javob yo'q

10. Hujjat miqdori va (yoki) so'rovning qidiruv shakli va (yoki) qidiruv shakli, shuningdek, qidiruv tizimiga kiritilgan aniq atamalar va matnda mavjud bo'lgan ma'lumotlar soniga kiritilgan hujjat yoki so'rov qanday nomlanadi?

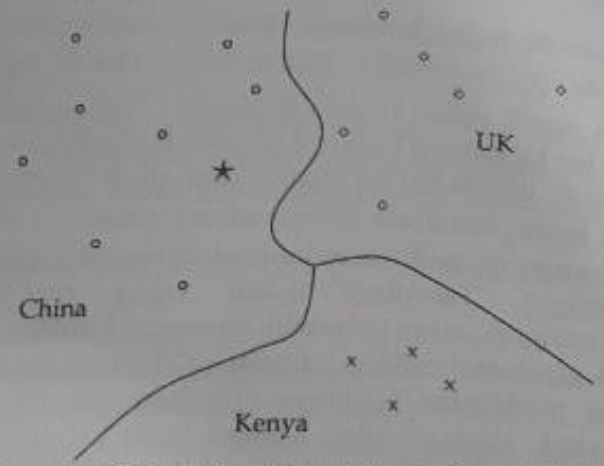
A) To'liqlikni indekslash
B) Indeksning o'ziga xos xususiyati
C) Bepul indekslash
D) To'g'ri javob yo'q

Naive Bayes-da hujjat ko'rinishi atamalar ketma-ketligi yoki e_1 ikkilik vektoridir $\langle e_1, \dots, e_{|t|} \rangle \in \{0,1\}^{|t|}$. Ushbu bobda ular 6- bobda ishlab chiqilgan matn tasnifi uchun boshqa ko'rinishni, vektor fazo modelini qabul qiladi. U har bir hujjatni bitta real qiymatli komponentga ega vektor sifatida ifodalaydi, odatda har bir atama uchun ***tf-idf*** og'irligi. Shunday qilib, **X** xujjat fazosi, tasniflash funksiyasining sohasi $\gamma, is\ \mathbb{R}^{|t|}$ mavjud. Ushbu bobda haqiqiy qiymatli vektorlarda ishlaydigan bir qator tasniflash usullari keltirilgan. Tasniflash uchun vektor fazo modelidan foydalanishda asosiy gipoteza qo'shnilik gipotezasi hisoblanadi.

Qo'shnilik gipotezasi sinfdagi hujjatlar qo'shni hududni tashkil qiladi va turli sinflar mintaqalari bir-biriga mos kelmaydi. Ko'p tasniflash vazifalari mavjud, xususan, ular 2-bobda duch kelgan matn tasnifi turlaridir, bu yerda sinflarni so'zni aniqlash bilan ajratish mumkin. Misol uchun, Xitoy sinfidagi hujjatlar odatda Xitoy, Pekin va Mao kabi o'lchamlar bo'yicha yuqori qiymatlarga ega bo'lsa, Buyuk Britaniya sinfidagi hujjatlar London, Britaniya va Qirolicha uchun yuqori qiymatlarga ega. Shunday qilib, ikki sinfning hujjatlari 3.1- rasmda ko'rsatilganidek, alohida qo'shni hududlarni tashkil qiladi va ularni ajratib turadigan chegaralarni chizish va yangi hujjatlarni tasniflash mumkin. Bu qanday aniq amalga oshirilganligi - bu bobning mavzusi hisoblanadi. Hujjatlar to'plamining qo'shni hududga ko'rsatilishi yoki ko'rsatilmasligi, hujjat ko'rsatish uchun qilgan aniq tanlovga bog'liq. Bu baholash turi, to'xtash ro'yxati va h.k. ga bog'liq.

Hujjatni ko'rsatish muhimligini bilish uchun guruh tomonidan yozilgan ikkita sinfni ko'rib chiqish va boshqalar bir kishi tomonidan yozilganligini ko'rish mumkin. 1 shaxs olmoshining tez-tez kelishi bir shaxs sinfiga dalildir. Ammo agar to'xtash ro'yxatidan foydalanilsa bu ma'lumot hujjat ko'rinishidan o'chiriladi. Agar tanlangan hujjat ko'rinishi noqulay bo'lsa tutashuv gipotezasi bajarilmaydi va vektor fazosini muvaffaqiyatli tasniflash mumkin emas. 1- qismning 6 va 7-boblarda ularni vaznli tasvirlarini xususan, uzunlik bo'yicha normallashtirilgan ***tf-idf*** ko'rinishlarini afzal ko'rishga olib kelgan bir xil mulohazalar ham qo'llaniladi. Misol uchun, hujjatda 5 marta takrorlangan atama bir martali atamaga qaraganda yuqoriroq vaznga ega bo'lishi kerak, ammo 5 marta kattaroq og'irlik atamaga juda katta urg'u beradi. Vektor

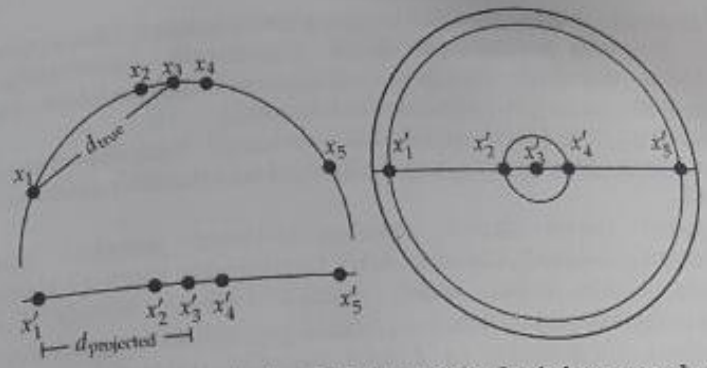
fazosini tasniflashda vaznsiz va normallashtirilmagan hisoblashlardan foydalanmaslik kerak.



3.1- rasm. **Vektor fazosini uchta sinfga ajratish**

Ushbu bobda ikkita vektor fazosini tasniflash usullarini kiritamiz, Rocchio va kNN. Rokkio tasnifi (3.2- bo'lim) vektor maydonini markazlar yoki prototiplarga asoslangan hududlarga ajratadi. Har bir sinf uchun bittadan sinfdagi barcha hujjatlarning massa markazi sifatida hisoblanadi. ***Rokkio*** tasnifi oddiy va samarali, ammo agar sinflar taxminan o'xshash radiusli sharlar bo'lmasa, noto'g'ri hisoblanadi. ***kNN*** yoki **k** eng yaqin qo'shni tasnifi (3.3- bo'lim) test hujjatiga **k** eng yaqin qo'shnilarning ko'pchilik sinfini belgilaydi. ***kNN*** aniq treningni talab qilmaydi va qayta ishlanmagan o'quv to'plamidan to'g'ridan-to'g'ri tasniflashda foydalanishi mumkin. Hujjatlarni tasniflashda boshqa tasniflash usullariga qaraganda unchalik samarali emas. Agar o'quv majmuasi katta bo'lsa, u holda ***kNN*** sferik bo'lmagan va boshqa murakkab sinflarni Rocchio'dan yaxshiroq boshqara oladi.

Ko'p sonli matn tasniflagichlari chiziqli tasniflagichlar sifatida ko'rib chiqilishi mumkin – bu xususiyatlarning oddiy chiziqli birikmasi asosida tasniflovchi tasniflagichlardir(3.4- bo'lim). Bunday klassifikatorlar xususiyatlar makonini quyida batafsil bayon qilinadigan tarzda chiziqli qaror giperplanlari bilan ajratilgan hududlarga ajratadi. Noto'g'ri kelishmovchilik tufayli (3.6- bo'lim) yanada murakkab chiziqli bo'lmagan modellar chiziqli modellarga qaraganda tizimli ravishda yaxshiroq emas.



3.2- rasm. **Birlik sferasining kichik maydonlarining proyeksiyalari**

Chapda: 2D yarim doiraning 1D ga proyeksiyasi. x koordinatalari -0.9,-0.2,0,0.2,0.9 da x_1, x_2, x_3, x_4, x_5, nuqtalar uchun masofa $|x_2x_3|$ 0,201 faqat | dan 0,5% ga farq qiladi $|x_2x_3|=0,2$; $|x_1x_5|/|x'_1x'_5| = d_{true}/d_{projected} \approx 1.06/0.9 = 1.18$ katta maydonni proyeksiyalashda katta buzilish (18%) misolidir. O'ngda: 3D yarim sharning 2D ga mos keladigan proyeksiyasi.

Chiziqli bo'lmagan modellar cheklangan miqdordagi o'quv ma'lumotlariga mos keladigan ko'proq parametrlarga ega, kichik va shovqinli ma'lumotlar to'plamlari uchun xato qilish ehtimoli ko'proq. Ikki toifali klassifikatorlarni ikkitadan ortiq sinflarga ega bo'lgan masalalarga qo'llashda bitta vazifalar mavjud - hujjat bir-birini istisno qiladigan bir nechta sinflardan biriga tayinlanishi kerak - va har qanday vazifalar - hujjat istalgan raqamga tayinlanishi mumkin. 3.5- bo'limda tushuntirib beriladigan sinflar. Ikki toifali klassifikatorlar har qanday muammolarni hal qiladi va muammolarni hal qilish uchun bittaga birlashtirilishi mumkin.

3.1. Hujjat ko'rinishlari va vektor fazolardagi bog'liqlik o'lchovlari

1- qismning 6- bobda bo'lgani kabi, hujjatlarni **R|V|** da vektor sifatida ko'rsatish mumkin. Vektor tasnifidagi hujjat vektorlarining xossalarini ko'rsatish uchun bu vektorlarni 3.1- rasmdagi misoldagidek tekislikdagi nuqtalar sifatida ko'rsatiladi.

Haqiqatda hujjat vektorlari ***gipersfera*** yuzasiga ishora qiluvchi uzunlik bo'yicha normalangan birlik vektorlaridir. Rasmlardagi 2D tekisliklarni quyidagi rasmda ko'rsatilganidek ***(giper-)sfera*** yuzasi tekisligiga proyeksiyalar sifatida ko'rish mumkin.

3.2- rasmda sirtning kichik joylari bilan cheklanib, tegishli proyeksiyani tanlansa, shar yuzasi va proyeksiya tekisligidagi masofalar taxminan bir xil bo'ladi.

Ko'pgina vektor fazosi tasniflagichlarining qarorlari masofa tushunchasiga asoslanadi, masalan, ***kNN*** tasnifida eng yaqin qo'shnilarni hisoblashda. Ushbu bobda asosiy masofa o'lchovi sifatida ***Evklid*** masofasidan foydalaniladi. Avvalroq uzunlik normallashtirilgan vektorlar uchun kosinus o'xshashligi va ***Evklid*** masofasi o'rtasida to'g'ridan-to'g'ri mos kelishini kuzatilgan edi. Vektor fazosini tasniflashda ikkita hujjatning o'zaro bog'liqligi o'xshashlik yoki masofa bilan ifodalanishi kamdan-kam ahamiyatga ega. Biroq, vektor fazosini tasniflashda hujjatlardan tashqari vektorlarning ***centroidlari*** yoki o'rtacha ko'rsatkichlari ham muhim rol o'ynaydi. ***Sentroidlar*** uzunligi normallashtirilmagan. Normallashtirilmagan vektorlar uchun nuqta natijasi, kosinus o'xshashligi va Evklid masofasi umuman boshqacha harakatga ega. Hujjat va markaz o'rtasidagi o'xshashlikni hisoblashda asosan kichik mahalliy hududlar bilan bog'liq bo'ladi va mintaqa qanchalik kichik bo'lsa, uchta o'lchovning xatti-harakati shunchalik o'xshash bo'ladi.

Misol. Kichkina maydonlar uchun ***gipersfera*** sirtidagi masofalar uning proyeksiyasidagi masofalar bilan yaxshi yaqinlashtiriladi (3.2-rasm), chunki kichik burchaklar uchun a ≈ sin(a). Qaysi o'lchamdagi burchak uchun buzilish mavjud (a/ sin(a) (i) 1,01, (ii) 1,05 va (iii) 1,1)?

3.2. Rokkio tasnifi

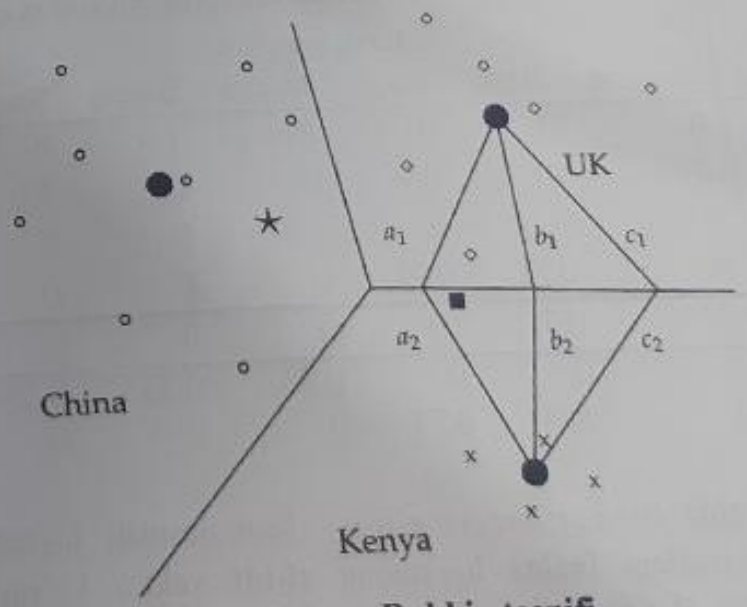
3.1-rasmda ikki o'lchovli (2D) fazoda uchta sinf, Xitoy, Buyuk Britaniya va Keniya ko'rsatilgan. Hujjatlar doiralar, olmoslar va X belgilari sifatida ko'rsatilgan. Qaror chegaralari deb ataladigan rasmdagi chegaralar uchta sinfni ajratish uchun tanlangan, ammo boshqacha tarzda to'g'ridan to'g'ri amalga oshiriladi. Rasmda yulduz sifatida tasvirlangan yangi hujjatni tasniflash uchun u joylashgan hududni aniqlanadi va unga ushbu mintaqaning sinfini - bu holda Xitoy belgilanadi. Vektor fazosini

tasniflashda ularning vazifasi yaxshi chegaralarni hisoblaydigan algoritmlarni ishlab chiqishdir, bunda "yaxshi" ta'lim jarayonida ko'rinmaydigan ma'lumotlarning yuqori tasnifi aniqligini anglatadi. Ehtimol, yaxshi sinf chegaralarini hisoblashning eng mashhur usuli bu chegaralarni aniqlash uchun markazlardan foydalanadigan Rokkio tasnifidir. **C** sinfining ***centroidi*** vektor o'rtacha yoki uni a'zolarining massa markazi sifatida hisoblanadi:

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d) \qquad (3.1)$$

bu yerda $\mathbf{D_c}$ - sinfi **c** bo'lgan **D** dagi hujjatlar to'plamidir. $\mathbf{D_c = \{d:h_d, c_i \in D\}}$. d ning normalangan vektorini $\vec{v}(d)$ bilan belgilanadi (1- qism, (6.11) tenglama). 3.3- rasmda uchta misol ***centroid*** qattiq aylana shaklida ko'rsatilgan. ***Rokkio*** tasnifidagi ikkita sinf o'rtasidagi chegara ikkita markazdan teng masofada joylashgan nuqtalar to'plamidir. Masalan, $|\mathbf{a_1}| = |\mathbf{a_2}|$, $|\mathbf{b_1}| = |\mathbf{b_2}|$, va $|\mathbf{c_1}| = |\mathbf{c_2}|$ rasmda. Ushbu nuqtalar to'plami har doim chiziqdir.

M o'lchamli fazoda chiziqning umumlashtirilishi ***giperplandir***, uni qondiradigan $\vec{x}$ nuqtalar to'plami sifatida aniqlanadi: $\vec{w}^T \vec{x} = b$



3.3-rasm. **Rokkio tasnifi**

bu yerda $\vec{w}$ - giper tekislikning M o'lchamli normal vektori 1 va b doimiy. Giper tekisliklarning bu ta'rifi chiziqlarni (2D dagi istalgan chiziq $w_1x_1+w_2+x_2=b$) bilan aniqlanishi mumkin) va 2 o'lchovli tekisliklarni (3D dagi har qanday tekislik $w_1x_1+w_2x_2+w_3x_3=b$ bilan aniqlanishi mumkin) o'z ichiga oladi. $w_3x_3 = b$. Chiziq tekislikni ikkiga, tekislik 3 o'lchovli fazoni ikkiga, giper tekisliklar esa yuqori o'lchamli bo'shliqlarni ikkiga ajratadi. Shunday qilib, Rocchio tasnifidagi sinf mintaqalarining chegaralari giperplanlardir. Rokkiodagi tasniflash qoidasi nuqtani u tushadigan mintaqaga qarab tasniflashdir. Ekvivalent tarzda nuqta eng yaqin bo'lgan **centroid** $\vec{m}(c)$ ni aniqlaymiz va keyin uni **c** bilan belgilanadi. Misol tariqasida 3.3- rasmdagi yulduz ko'rib chiqiladi. U koinotning Xitoy hududida joylashgan va shuning uchun ***Rokkio*** uni Xitoyga tayinlaydi. Rokkio algoritmini psevdokodda 3.4- rasmda ko'rish mumkin.

$$\vec{w}^T\vec{x} = b \qquad (3.2)$$

Asosiy chiziqli algebradan $\vec{v}\cdot\vec{w} = \vec{v}^T\cdot\vec{w}$, ya'ni $\vec{v}$ va $\vec{w}$ ning nuqta ko'paytmasi $\vec{v}$ va $\vec{w}$ ning ***transpozitsiya***sini matritsali ko'paytirish natijasida hosil bo'lganiga teng ekanligini eslang.

3.1- jadval. **2.1- jadvaldagi ma'lumotlar uchun vektorlar va sinf markazlari**

	term weights					
vector	Chinese	Japan	Tokyo	Macao	Beijing	Shanghai
$\vec{d}_1$	0	0	0	0	1.0	0
$\vec{d}_2$	0	0	0	0	0	1.0
$\vec{d}_3$	0	0	0	1.0	0	0
$\vec{d}_4$	0	0.71	0.71	0	0	0
$\vec{d}_5$	0	0.71	0.71	0	0	0
$\vec{\mu}_c$	0	0	0	0.33	0.33	0.33
$\vec{\mu}_{\bar{c}}$	0	0.71	0.71	0	0	0

3.1- jadvalda $(1+\log_{10} tf_{t,d})\log_{10}(4/df_t)$ if $tf_{t,d}$ formulasidan foydalangan holda 2.1- jadvaldagi beshta hujjatning **tf-idf** vektor ko'rinishlari hisoblanadi agar $\mathbf{tf}_{t,d}>0$ bo'lsa (1- qism (6.14) tenglama). Ikki sinf

centroidlari $\mu_c = 1/3\cdot(\vec{d}_1+\vec{d}_2+\vec{d}_3)$ va $\mu_{\bar{c}} = 1/3\cdot(\vec{d}_4)$. Sinov hujjatining markazdan uzoqlari $|\mu_c-\vec{d}_5|\approx 1.15$ *and* $|\mu_{\bar{c}}-\vec{d}_5|\approx 0.0$. Shunday qilib, ***Rocchio*** **ds** ni **c** ga belgilaydi. Bu holda ajratuvchi ***giperplane*** quyidagi parametrlarga ega bo'ladi:

$$\vec{w} \approx (0 - 0.71 - 0.71\ 1/3\ 1/3\ 1/3)^T$$

$$b = -1/3$$

$\vec{w}$ va **b** ni hisoblash uchun quyida misollar keltirilgan. Ushbu ***giperplane*** hujjatlarni kerakli tarzda ajratishini osongina tekshirish mumkin: $\vec{w}^T\vec{d}_1 \approx 0\cdot 0+-0.71\cdot 0+-0.71\cdot 0+1/3\cdot 1.0+1/3\cdot 0=1/3>b$ shunga o'xshash, $\vec{w}^T\vec{d}_i > b$ *for* $i=2$ *and* $i=3$ va $\vec{w}^T\vec{d}_5 = -1<b$. Shunday qilib, **c** dagi hujjatlar giperplanning ustida joylashgan $(\vec{w}^T\vec{d}_i > b)$ va **c** dagi hujjatlar giperplandan pastda joylashgan $(\vec{w}^T\vec{d}_i < b)$. 3.4- rasmdagi tayinlash mezoni ***Evklid*** masofasi (**APPLYROCCHIO**, 1-qator). Muqobil kosinus o'xshashligi: **d** ni **c** sinfiga belgilash kerak. $c = \arg\max_{c'} \cos(\vec{\mu}(c'), \vec{v}(d))$ 3.1- bo'limda muhokama qilinganidek, ikkita tayinlash mezoni ba'zan turli tasniflash qarorlarini qabul qiladi. Bu yerda **Rokkio** tasnifining ***Evklid*** masofasi variantini taqdim etiladi chunki u ***Rokkioning K-vositalarni klasterlashiga*** yaqin mosligini ta'kidlaydi (5.4- bo'lim). ***Rokkio tasnifi*** - bu Rocchio bilan bog'liq fikr-mulohazalarning bir shakli. Tegishli teskari aloqada Rocchio vektorining eng muhim tarkibiy qismiga mos keladigan tegishli hujjatlarning o'rtacha ko'rsatkichi tegishli hujjatlarning "sinfi" ning markaziy ko'rsatkichidir. Rokkio tasnifida Rokkio formulasining so'rov komponentini o'tkazib yuborish kerak chunki matn tasnifida so'rov yo'q. Rokkio tasnifi $J > 2$ sinflariga nisbatan qo'llanilishi mumkin, Rokkioning tegishliligi haqidagi fikr-mulohazasi esa faqat ikkita sinfni, tegishli va tegishli emasligini ajratish uchun mo'ljallangan. Rokkio tasnifidagi sinflar ishlash usuli yaqinligidan tashqari, radiuslari o'xshash bo'lgan taxminiy sharlar bo'lishi kerak. 3.3- rasmda Buyuk Britaniya va Keniya o'rtasidagi chegara ostidagi tekis kvadrat Buyuk Britaniya sinfiga ko'proq mos keladi chunki Buyuk Britaniya Keniyaga qaraganda ko'proq tarqalgan. Ammo Rokkio uni Keniyaga tayinlaydi chunki u sinfdagi ballarni taqsimlash tafsilotlarini e'tiborsiz qoldiradi va tasniflash uchun faqat markazdagi masofadan foydalanadi.

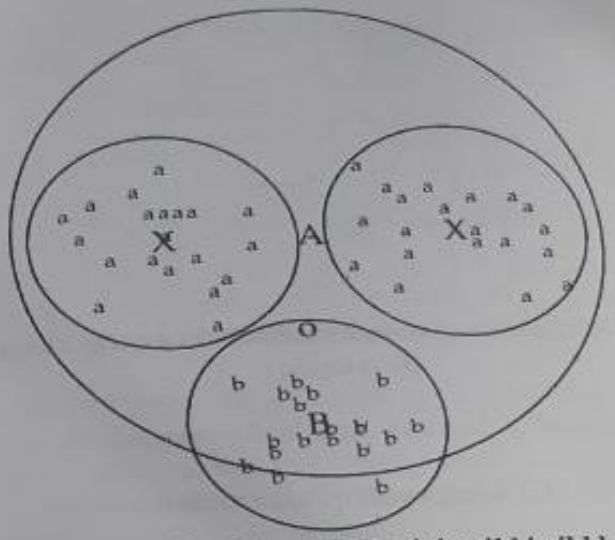
TRAINROCCHIO($\mathbb{C}, \mathbb{D}$)
1 **for each** $c_j \in \mathbb{C}$
2 **do** $D_j \leftarrow \{d : \langle d, c_j \rangle \in \mathbb{D}\}$
3 $\vec{\mu}_j \leftarrow \frac{1}{|D_j|} \sum_{d \in D_j} \vec{v}(d)$
4 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}$

APPLYROCCHIO($\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}, d$)
1 **return** $\arg\min_j |\vec{\mu}_j - \vec{v}(d)|$

3.4- rasm. **Rocchio tasnifi: Trening va sinov**

Sharsimonlik haqidagi taxmin 2.5- rasmda ham amal qilmaydi. "a" sinfini bitta prototip bilan yaxshi ifoda qilish qiyin chunki u ikkita klasterga ega. Rokkio ko'pincha bu turdagi ***multimodal sinfni*** noto'g'ri tasniflaydi. Multimodallik uchun matn tasnifiga misol 1989 yilda o'z nomini ***Myanmaga*** o'zgartirgan ***Birma*** kabi mamlakatdir. Nom o'zgarishidan oldin va keyin ikkita klaster kosmosda bir-biriga yaqin bo'lishi shart emas. Shuningdek, tegishli fikr-mulohazalarda multimodallik muammosiga duch kelish mumkin.

Ikki sinfli tasnif - bu sinflar kamdan-kam hollarda o'xshash radiusli sharlar kabi taqsimlanadigan yana bir holat. Ko'pgina ikki toifali tasniflagichlar Xitoy kabi fazoning kichik hududini egallagan sinfni va uning keng tarqalgan to'ldiruvchisini ajratib turadi. Teng radiuslarni qabul qilish ko'p sonli noto'g'ri musbatlarga olib keladi. Shuning uchun ikki toifali tasniflash muammolarining aksariyati o'zgartirilgan qaror qoidasini talab qiladi:
d sinfga tayinlang c *iff* $|\vec{\mu}(c) - \vec{v}(d)| < |\vec{\mu}(\bar{c}) - \vec{v}(d)| - b$ musbat doimiy uchun **b**. Rokkioning tegishli fikr-mulohazalarida bo'lgani kabi, salbiy hujjatlarning markaziy qismi ko'pincha umuman ishlatilmaydi. Shuning uchun qaror mezonini soddalashtiradi $|\vec{\mu}(c) - \vec{v}(d)| <$ b musbat doimiy **b'** uchun.



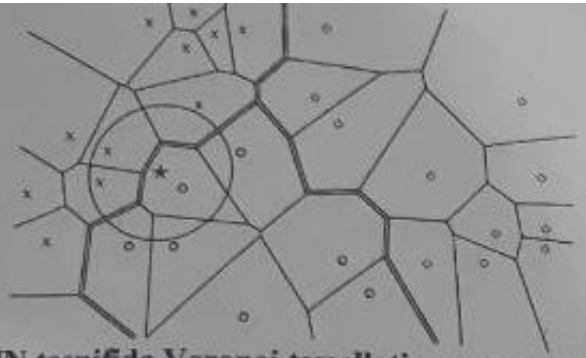
3.5- rasm. **"a" multimodal klassining ikki xil klasteri**

"a" multimodal klassi ikki xil klasterdan iborat (markazi X harflarida joylashgan kichik yuqori doiralar). Rokkio tasnifi "**o**" ni "**a**" deb noto'g'ri tasniflaydi chunki u "**b**" sinfining markaziy **B** markaziga qaraganda "**a**" sinfining **A** markaziga yaqinroqdir.

3.2- jadvalda ***Rokkio*** tasnifining vaqt murakkabligi berilgan. Barcha hujjatlarni ularning tegishli (normallashtirilmagan) centroidiga qo'shish **D(|D|L$_{ave}$)** (**D(|D||V|)** dan farqli o'laroq) chunki faqat nolga teng bo'lmagan yozuvlarni ko'rib chiqish kerak. Markazni hisoblash uchun har bir vektor yig'indisini o'z sinfining kattaligiga bo'lish **D(|V|)** ga teng. Umuman olganda, mashg'ulot vaqti to'plam hajmida chiziqli hisoblanadi. Shunday qilib, ***Rocchio tasnifi*** va *Naive Bayes* bir xil chiziqli mashg'ulot vaqti murakkabligiga ega. Keyingi bo'limda ***sferik bo'lmagan***, uzilgan yoki boshqa tartibsiz shakllarga ega bo'lgan sinflar bilan yaxshiroq shug'ullanadigan yana bir vektor fazosini tasniflash usuli va **kNN** bilan tanishtiriladi.

Misol. Rocchio tasnifi hujjatga o'quv majmuasi yorlig'idan farqli yorliq belgilashi mumkinligini ko'rsating.

D(T) uchun **D(|D|L$_{ave}$)** yozish mumkin va test hujjatlari uzunligi chegaralangan deb faraz qilinadi.



3.6-rasm. **1NN tasnifida Voronoi tessellation va qaror chegaralari (ikki chiziq)**

Bu rasmda uchta sinf: X, doira va olmos sinflari mavjud.

3.3. k eng yaqin qo'shni modeli

Rokkiodan farqli o'laroq, k eng yaqin qo'shni yoki **kNN** tasnifi mahalliy darajada qaror chegarasini belgilaydi. **1NN** uchun har bir hujjatni eng yaqin qo'shnisining sinfiga ajratish mumkin. kNN uchun har bir hujjatni uning k eng yaqin qo'shnilarining ko'pchilik sinfiga tayinlanadi, bu yerda **k** parametrdir. KNN tasnifining asosi shundan iboratki, qo'shnilik gipotezasiga asoslanib, **d** test hujjati **d** atrofidagi mahalliy mintaqada joylashgan o'quv hujjatlari bilan bir xil belgiga ega bo'lishi mumkin.

1NNdagi qaror chegaralari 3.6- rasmda ko'rsatilganidek, ***Voronoi mozaikasi***ning birlashtirilgan segmentlari. Obyektlar to'plamining Voronoi tesselatsiyasi bo'shliqni Voronoi hujayralariga taqsimlaydi. Bu yerda har bir obyektning hujayrasi boshqa obyektlarga qaraganda obyektga yaqinroq bo'lgan barcha nuqtalardan iborat. Ularning holatlarida obyektlar hujjatlardir. ***Voronoy mozaikasi*** keyinchalik samolyotni **|D|** ga bo'ladi qavariq ko'pburchaklar, ularning har biri o'zining tegishli hujjatini o'z ichiga oladi. 3.6- rasmda ko'rsatilganidek, bu yerda qavariq ko'pburchak chiziqlar bilan chegaralangan 2 o'lchovli fazodagi qavariq mintaqadir. **kNN**da umumiy **k ∈ N** uchun eng yaqin ***k qo'shnilar to'plami*** bir xil bo'lgan fazodagi mintaqani ko'rib chiqiladi. Bu yana qavariq ko'pburchak bo'lib, fazo qavariq ko'pburchaklarga

bo'linadi va ularning har birida ***k eng yaqin qo'shnilar*** to'plami o'zgarmas bo'ladi.

```
TRAIN-KNN(C, D)
1  D' ← PREPROCESS(D)
2  k ← SELECT-K(C, D')
3  return D', k

APPLY-KNN(C, D', k, d)
1  S_k ← COMPUTENEARESTNEIGHBORS(D', k, d)
2  for each c_j ∈ C
3  do p_j ← |S_k ∩ c_j|/k
4  return arg max_j p_j
```

2.7- rasm. **kNN ta'lim va sinovdan o'tkazish**

p_j - $P(c_j|S_k) = P(c_j|d)$ uchun taxmin. **c_j** sinfidagi barcha hujjatlar to'plamini bildiradi.

1NN unchalik mustahkam emas. Har bir test hujjatining tasnifi qarori noto'g'ri etiketlangan yoki xatoligi mavjud bo'lishi mumkin bo'lgan yagona o'quv hujjatining sinfiga tayanadi. **k > 1** uchun **kNN** yanada mustahkamroq hisoblanadi. U hujjatlarni o'zlarining eng yaqin qo'shnilarining ko'pchilik sinfiga topshiradi va tasodifiy ravishda bajaralidi. Ushbu **kNN** tasniflash algoritmining ehtimolli versiyasi mavjud. Uni **c** sinfiga a'zo bo'lish ehtimolini nisbati sifatida baholash mumkin.

k eng yaqin qo'shnilar **c** 3.6- rasmda **k = 3** ga misol keltirilgan. Yulduzning sinfga a'zoligi uchun taxminiy taxminlar $\hat{P}$(doira klassi|yulduz) = 1/3, $\hat{P}$(X sinf|yulduz) = 2/3 va $\hat{P}$(olmos sinfi|yulduz) = 0. 3 nn bahosi ($\hat{P}_1$(doira klassi|yulduz) = 1/3) va 1nn bahosi ($\hat{P}_1$(doira klassi|yulduz) = 1) **X** klassini afzal ko'rgan 3nn va afzal ko'rgan 1nn bilan farqlanadi. Doira sinfi **kNN**dagi k parametri ko'pincha tasniflash muammosi bo'yicha tajriba yoki bilimga asoslangan holda tanlanadi. Bog'lanish ehtimolini kamaytirish uchun **k** ni qiymati boshqacha bo'lishi ma'qul. k = 3 va k = 5 keng tarqalgan tanlovdir lekin 50 va 100 orasidagi ancha katta qiymatlar ham ishlatiladi. Parametrni o'rnatishning muqobil usuli - bu o'quv majmuasining ushlab turgan qismida eng yaxshi natijalarni beradigan **k** ni tanlashdir.

Ko'pburchakni yuqori o'lchamlarga umumlashtirish politopdir. ***Politop*** - **M** o'lchovli fazodagi **(M - 1)** o'lchovli gipertekisliklar bilan chegaralangan mintaqadir. **M** o'lchamlarida **kNN** uchun qaror chegaralari **(M-1)** o'lchovli giperplanlarning segmentlaridan iborat bo'lib, ular

hujjatlar to'plamini o'qitish uchun *Voronoi mozaikasini* qavariq politoplarga aylantiradi. Hujjatni uning **k** *eng yaqin qo'shnilar*ining ko'pchilik sinfiga tayinlashning qaror mezoni **M=2** (ko'pburchaklarga mo'ljallangan mozaikalar) va **M > 2** (politoplarga mo'ljallangan mozaikalar) uchun teng qo'llaniladi.

Shuningdek, ***k eng yaqin qo'shnilarning*** "*ovozlarini*" ularning kosinus o'xshashligi bo'yicha baholash mumkin. Ushbu sxemada sinfning bali quyidagicha hisoblanadi:

3.3- jadval. **kNN tasnifi uchun o'qitish va sinov vaqtlari**

kNN with preprocessing of training set	
training	$\Theta(\|\mathbb{D}\|L_{ave})$
testing	$\Theta(L_a + \|\mathbb{D}\|M_{ave}M_a) = \Theta(\|\mathbb{D}\|M_{ave}M_a)$
kNN without preprocessing of training set	
training	$\Theta(1)$
testing	$\Theta(L_a + \|\mathbb{D}\|L_{ave}M_a) = \Theta(\|\mathbb{D}\|L_{ave}M_a)$

$\mathbf{M_{ave}}$ - to'plamdagi hujjatlar lug'atining o'rtacha hajmi.

Bu yerda $\mathbf{S_k(d)}$ **d** ning ***k eng yaqin qo'shnilari*** to'plami va $\mathbf{I_c(d') = 1}$ **iff d' c** sinfida bo'lsa, aks holda 0 ga teng bo'ladi. Keyin hujjatni eng yuqori ball olgan sinfga topshiramiz. O'xshashlik bo'yicha tortish ko'pincha oddiy ovoz berishdan ko'ra aniqroqdir. Misol uchun, agar ikkita sinfning eng yuqori **k** qismida bir xil miqdordagi qo'shnilari bo'lsa, qo'shnilari o'xshash bo'lgan sinf g'alaba qozonadi. 3.7- rasmda ***kNN algoritmi*** umumlashtirilgan.

$$score(c,d) = \sum_{d' \in S_k(d)} I_c(d')\cos(\vec{v}(d'),\vec{v}(d))$$

Misol. 3.1- jadvaldagi to'rtta o'quv hujjatidan test hujjatining masofalari

$|\vec{d}_1 - \vec{d}_5| = |\vec{d}_2 - \vec{d}_5| = |\vec{d}_3 - \vec{d}_5| = 1.41$ *and* $|\vec{d}_4 - \vec{d}_5| = 0.0$ **d5** ning eng yaqin qo'shnisi hisoblanadi. Shuning uchun **d4** va 1NN **d5** ni **d4** ning sinfiga belgilaydi.

3.3.1. KNN ning vaqt murakkabligi va optimalligi

3.3- jadvalda kNN ning vaqt murakkabligi berilgan. kNN boshqa tasniflash algoritmlaridan ancha farq qiladigan xususiyatlarga ega. KNN klassifikatorini o'rgatish oddiygina **k** ni aniqlash va hujjatlarni oldindan qayta ishlashdan iborat. Haqiqatan ham, agar **k** uchun qiymatni oldindan tanlansa va oldindan ishlov berilmasa, u holda kNN umuman treningni

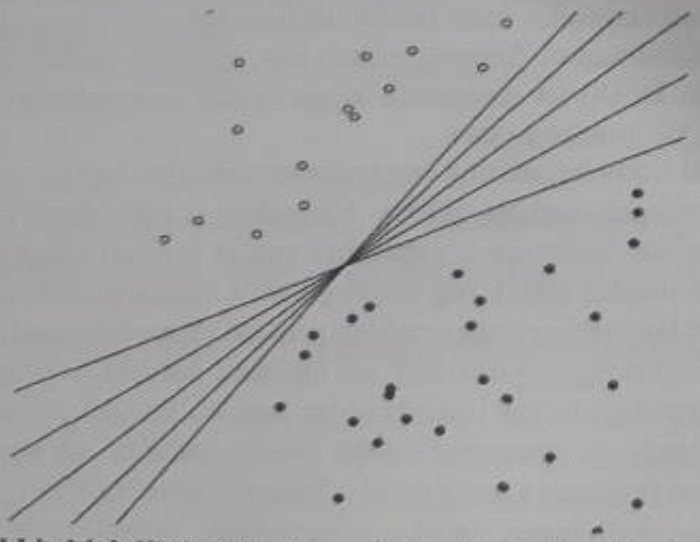
talab qilmaydi. Amalda, ***tokenlash*** kabi oldindan ishlov berish bosqichlarini bajarish kerak. Har safar yangi sinov hujjatini tasniflaganda qayta-qayta emas, balki o'quv bosqichining bir qismi sifatida o'quv hujjatlarini bir marta qayta ishlash maqtiqiyroq hisoblanadi. Sinov vaqti kNN uchun $\Theta(|\mathbb{D}|M_{ave}Ma)$. Ta'lim to'plamining o'lchamida chiziqli bo'ladi chunki har bir o'quv hujjatining sinov hujjatidan masofasini hisoblash kerak. Sinov vaqti **J** sinflar soniga bog'liq emas, shuning uchun **kNN** katta **J** bilan bog'liq muammolar uchun potentsial afzalliklarga ega. **kNN** va ***Rocchio*** tasnifida (centroidlar) yoki ***Naive Bayesda*** (avvalgi) bo'lgani kabi parametrlarni hech qanday baholash amalga oshirilmaydi va shartli ehtimollar bajariladi). kNN oddiygina treningdagi barcha misollarni yod oladi, o'rnatadi va keyin ular bilan test hujjatini solishtiradi. Shu sababli, kNN xotiraga asoslangan o'rganish yoki misolga asoslangan o'rganish deb ham ataladi.

Odatda mashinali o'qitishda imkon qadar ko'proq o'quv ma'lumotlariga ega bo'lish maqsadga muvofiqdir. Ammo kNNda katta o'quv majmualari tasniflashda jiddiy samaradorlik natijasi bilan birga keladi. KNN testini $\mathbf{D(|D|M_{ave}Ma)}$ ga qaraganda samaraliroq qilish mumkinmi yoki hujjatlar uzunligini hisobga olmaganda $\mathbf{D(|D|)}$ dan samaraliroq bo'lishi mumkinmi? Kichik o'lchamli **M** uchun tezkor kNN algoritmlari mavjud. Muayyan samaradorlikni oshirish uchun xato chegaralarini beradigan katta **M** uchun taxminlar ham mavjud (3.7- bo'lim). Ushbu taxminlar matn tasniflash ilovalari uchun keng qamrovli sinovdan o'tkazilmagan. Shuning uchun ular aniqlikni sezilarli darajada yo'qotmasdan $\mathbf{D(|D|)}$ ga qaraganda ancha yaxshi samaradorlikka erisha oladimi yoki yo'qmi aniq emas.

O'quvchi sinov hujjatining eng yaqin qo'shnilarini topish muammosi va ular qidiradigan maxsus qidiruv o'rtasidagi o'xshashlikni payqagan bo'lishi mumkin. So'rovga o'xshashligi yuqori bo'lgan hujjatlar uchun aslida, ikkala masala ham ***k eng yaqin qo'shni*** muammolari bo'lib, faqat kNNdagi test hujjatining nisbiy zichligi (vektori) bilan farqlanadi (10 yoki 100 ta nol bo'lmagan yozuvlar) va ularning kamligi (vektori) bilan maxsus qidiruvdagi so'rov (odatda nolga teng bo'lmagan 10 tadan kam) inobatga olinadi. 1- qismning 1.1-bo'limida samarali **ad hoc** qidirish uchun ***teskari indeks***ni kiritilgan edi. ***Invert indeks*** ham samarali *kNN* uchun yechimmi? ***Invert indeks*** so'rov bilan kamida bitta umumiy atamaga ega bo'lgan hujjatlarni qidirishni cheklaydi. Shunday qilib, kNN kontekstida agar test hujjatida ko'p sonli o'quv hujjatlari bilan atamalar

mos kelmasa ***teskari indeks*** samarali bo'ladi. Buni shunday bo'lishi tasniflash muammosiga bog'liq. Agar hujjatlar uzun bo'lsa va to'xtash ro'yxati ishlatilmasa unda kamroq vaqt tejaladi. Ammo qisqa hujjatlar va katta to'xtash ro'yxati bilan teskari indeks o'rtacha sinov vaqtini 10 yoki undan ko'p marta qisqartirishi mumkin. Invert indeksdagi qidiruv vaqti so'rovdagi atamalar ro'yxatining uzunligiga bog'liq. E'lonlar ro'yxati sublinear ravishda o'sadi to'plam uzunligi, chunki so'z boyligi mos ravishda ortadi.

Heaps qonuni - agar ba'zi atamalarning paydo bo'lish ehtimoli oshsa boshqalarning paydo bo'lish ehtimoli kamayishi kerak. Biroq, ko'pchilik yangi atamalar kamdan-kam uchraydi. Shuning uchun teskari indekslarni qidirishning murakkabligi **D(T)** deb olinadi va hujjatning o'rtacha uzunligi vaqt o'tishi bilan o'zgarmasligini hisobga olinsa **D(T) = D(|D|)** bo'ladi. Keyingi bobda kNN ning samaradorligi matnlarni tasniflashda eng aniq o'rganish usullariga yaqinligi ko'rib chiqiladi. Ta'lim usuli sifatining o'lchovi uning ***Bayes xatolik darajasi***, muayyan muammo uchun u tomonidan o'rganilgan tasniflagichlarning o'rtacha xato darajasidir. kNN nolga teng bo'lmagan ***Bayes xatolik*** darajasi bilan bog'liq muammolar uchun optimal emas - ya'ni eng yaxshi mumkin bo'lgan tasniflagich ham nolga teng bo'lmagan tasniflash xatosiga ega bo'lgan muammolar uchun. ***INN xatosi*** asimptotik (o'quv to'plamining ortishi bilan) bilan chegaralanadi.



3.8-rasm. **Ikki chiziqli ajratiladigan sinflarni ajratuvchi cheksiz ko'p gipertekisliklar**

Bayesning xatolik darajasi ikki baravar yuqori. Ya'ni, agar optimal klassifikatorning xatolik darajasi **x** bo'lsa u holda **1NN *asimptotik xato* darajasi 2x** dan kam bo'ladi. Bu shovqinning ta'siri bilan bog'liq - 2.5-bo'limda shovqinli xususiyatlar ko'rinishidagi shovqinning bir misoli allaqachon ko'rilgan edi, ammo shovqin boshqa shakllarni ham olishi mumkin, chunki ular keyingi bo'limda muhokama qilinadi. Shovqin kNN ning ikkita komponentiga ta'sir qiladi: ***sinov hujjati*** va ***eng yaqin o'quv hujjati***. Shovqinning ikkita manbasi qo'shimcha hisoblanadi shuning uchun 1NN ning umumiy xatosi optimal xato tezligidan ikki baravar yuqori. Bayes xatolik darajasi 0 bilan bog'liq muammolar uchun 1NN xato darajasi 0 ga yaqinlashadi chunki treninglar to'plamining hajmi oshadi.

Misol. Nima uchun kNN ***multimodal*** sinflarni ***Rokkio***ga qaraganda yaxshiroq boshqarishini tushuntiring.

3.4. Chiziqli va chiziqli bo'lmagan tasniflagichlar

Ushbu bo'limda ***Naive Bayes*** va ***Rocchio*** o'rganishning ikkita usuli chiziqli tasniflagichlarning namunalari, ehtimol matn tasniflagichlarining eng muhim guruhi ekanligini ko'rsatiladi va ular chiziqli bo'lmagan tasniflagichlar bilan taqqoslanadi. Munozarani soddalashtirish uchun ushbu bo'limda faqat ikkita klassifikator ko'rib chiqiladi va chiziqli klassifikator xususiyatlarning chiziqli birikmasini polga solishtirish orqali sinfga a'zolikni hal qiladigan ikki klassli tasniflagich sifatida aniqlanadi. Ikki o'lchovda chiziqli tasniflagich oddiy chiziqdir. Beshta misol 3.8-rasmda keltirilgan. Bu chiziqlar $w_1x_1 + w_2x_2 = b$ funksional ko'rinishga ega. Chiziqli klassifikatorning tasniflash qoidasi, agar $w_1x_1 + w_2x_2 > b$ bo'lsa, **c** ga hujjat va $w_1x_1 + w_2x_2 \leq b$ bo'lsa **c** ga belgilanishi kerak. Bu yerda, $(x_1, x_2)^T$ - hujjatning ikki o'lchovli vektor tasviri va $(x_1, x_2)^T$ - parametr vektori.

APPLYLINEAR CLASSIFIER $(\vec{w}, b, \vec{x})$

1 $score \leftarrow \sum_{i=1}^{M} w_i x_i$
2 **if** $score > b$
3 **then return** 1
4 **else return** 0

3.9-rasm. **Chiziqli tasniflash algoritmi**

Bu (**b** bilan birga) qaror chegarasini belgilaydi. Chiziqli klassifikatorning muqobil geometrik talqini 4.7-rasmda keltirilgan. Ushbu 2D chiziqli klassifikatorni (3.3) tenglamada takrorlangan (3.2) tenglamada bo'lgani kabi ***giper tekislik***ni aniqlash orqali yuqori o'lchamlarga umumlashtirish mumkin.

$$\vec{w}^T\vec{x} = b \tag{3.2}$$

Tayinlash mezoni quyidagicha: agar **w~ c** ga tayinlang $\vec{w}^T\vec{x} \le b$, agar $\vec{w}^T\vec{x} > b$. Chiziqli klassifikator sifatida ishlatadigan ***giper tekislik***ni ***qaror giperplaniyasi*** deb ataladi. **M** o'lchamdagi chiziqli tasniflashning tegishli algoritmi 3.9-rasmda ko'rsatilgan. Chiziqli tasnif dastlab ushbu algoritmning soddaligini hisobga olgan holda ahamiyatsiz ko'rinadi. Biroq, qiyinchilik chiziqli klassifikatorni o'rgatishda, ya'ni **w~** parametrlarini aniqlashda va o'quv majmuasiga asoslanadi.

Umuman olganda, ba'zi o'rganish usullari boshqalarga qaraganda ancha yaxshi parametrlarni hisoblab chiqadi.

Bunda o'rganish usuli sifatini baholash mezoni yangi ma'lumotlar bo'yicha o'rganilgan chiziqli tasniflagichning samaradorligi hisoblanadi. Endi ular ***Rocchio*** va ***Naive Bayes*** chiziqli tasniflagichlar ekanligini ko'rsatiladi.

Buni Rokkioda ko'rish uchun $\vec{x}$ vektori qaror chegarasida bo'lgan bo'lsa, e'tibor bering ikkita sinf markazlariga teng masofa:

$$|\vec{\mu}(c_1) - \vec{x}| = \vec{\mu}(c_2) - \vec{x}| \tag{3.4}$$

Ba'zi bir asosiy arifmetik amallar ashuni ko'rsatadiki, ***bunormal vector*** $\vec{w} = \vec{\mu}(c_1) - \vec{\mu}(c_2) va\ b = 0.5 * (|\vec{\mu}(c_1)|^2 - |\vec{\mu}(c_2)|^2)$ bo'lgan chiziqli tasniflagichga mos keladi. $\hat{P}(c|d)$ bilan **c** toifasini tanlaydigan qaror qoidasidan ***Naive Bayes***ning chiziqliligini olish mumkin (2.2-rasm) bu yerda:

$$\hat{P}(c|d) \propto \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c)$$

$\mathbf{n_d}$ - hujjatdagi lug'at tarkibiga kiruvchi tokenlar sonidir. To'ldiruvchi turkumni **c** deb belgilab, **logorifm** koeffitsientlari uchun quyidagilar olinadi:

$$\log \frac{\hat{P}(c|d)}{\hat{P}(\bar{c}|d)} = \log \frac{\hat{P}(c)}{\hat{P}(\bar{c})} + \sum_{1 \le k \le n_d} \log \frac{\hat{P}(t_k|c)}{\hat{P}(t_k|\bar{c})} \tag{3.5}$$

3.4- jadval. **Chiziqli tasniflagich**

t_i	w_i	d_{1i}	d_{2i}	t_i	w_i	d_{1i}	d_{2i}
prime	0.70	0	1	dlrs	-0.71	1	1
rate	0.67	1	0	world	-0.35	1	0
interest	0.63	0	0	sees	-0.33	0	0
rates	0.60	0	0	year	-0.25	0	0
discount	0.46	1	0	group	-0.24	0	0
bundesbank	0.43	0	0	dlr	-0.24	0	0

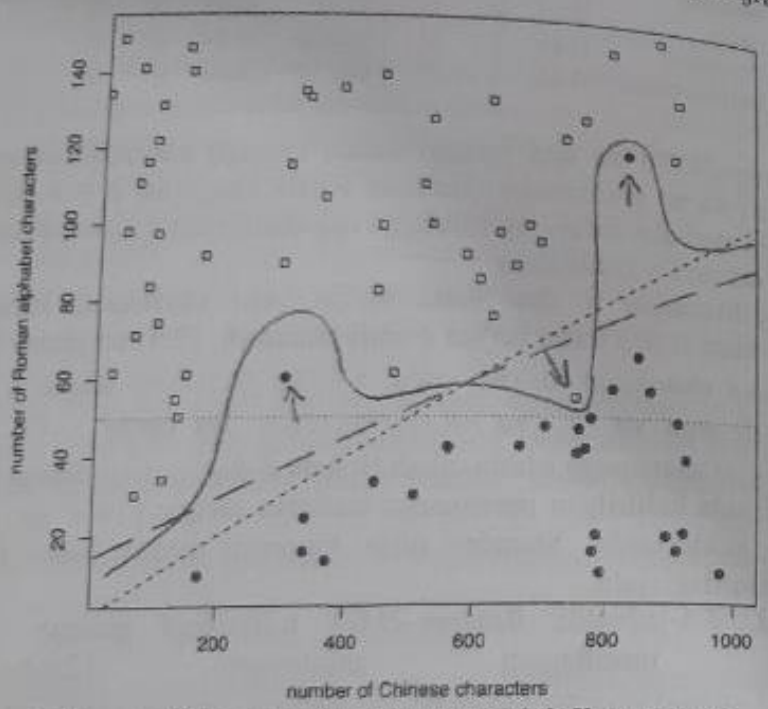
Reuters-21578 da sinf foizlari uchun chiziqli klassifikatorning t_i o'lchamlari va w_i parametrlari beriladi. Kirish chegarasi $\mathbf{b = 0}$. $\mathbf{D_{ii}}$ va dunyo kabi atamalar salbiy og'irliklarga ega chunki ular raqobatdosh sinf valyutasi uchun ko'rsatkichdir.

Agar koeffitsientlar 1 dan katta bo'lsa yoki ekvivalent logorifm koeffitsientlari 0 dan katta bo'lsa **c** sinfi tanlanadi. (3.5) tenglama (3.3) uchun misol ekanligini ko'rish oson. $b = -\log[\hat{P}(c)/\hat{P}(\bar{c})]$ ichida t_i ning takrorlanish soni va. $w_i = \log[\hat{P}(t_i|c)/\hat{P}(t_i|\bar{c})], x_i$ Bu yerda $i, 1 \le i \le M$, indeksi lug'at shartlariga ishora qiladi (**k** kabi **d** dagi pozitsiyalarga emas 2.4.1-bo'limda keltirilgan paremetrlar inobatga olingan) va $\vec{x}$ va $\vec{w}$ M o'lchamli vektorlardir. Shunday qilib, logorifm fazoda ***Naive Bayes*** chiziqli tasniflagichdir.

Misol. 3.4-jadvalda ***Reuters-21578*** toifasidagi qiziqish uchun chiziqli tasniflagich aniqlangan (2.6-bo'lim). $\vec{w}^T\vec{d}_1 = 0.67 + 0.46 \cdot 1 + (-0.71) \cdot 1 + (-0.35) \cdot 1 =$ bo'lgani uchun ular $\vec{d}_1$ "Stavka chegirmasi $\mathbf{d_{ii}}$ dunyo" hujjati foizga belgilanadi. $\vec{w}^T\vec{d}_2 = -0.01 \le b$. bo'lgani uchun to'ldiruvchi sinfga $\vec{d}_2$ "asosiy $\mathbf{d_{ii}}$" belgilanadi. Oddiylik uchun ushbu misolda oddiy ikkilik vektor tasvirini qabul qilinadi. Sodir bo'lgan shartlar uchun 1, sodir bo'lmagan shartlar uchun 0 qabul qilinadi.

3.10-rasm chiziqli muammoning grafik misoli bo'lib, ikkita sinfning asosiy taqsimotlari **P(d|c)** va **P(d|c)** chiziq bilan ajratilganligini anglatish uchun aniqlanadi. Bu ajratuvchi chiziq sinf chegarasi deb ataladi. Bu ikkita sinfning "***haqiqiy***" chegarasi va uni o'rganish usuli sinf chegarasini taxmin qilish uchun hisoblaydigan qaror chegarasidan ajratib turadi. Matn tasnifida odatdagidek, 3.10-rasmda (strelkalar bilan belgilangan) Ba'zi shovqinli hujjatlar mavjud bo'lib, ular sinflarning umumiy taqsimotiga yaxshi mos kelmaydi. 2.5-bo'limda shovqin xususiyatini noto'g'ri xususiyat sifatida aniqlangan edi. Bu hujjat taqdimotiga kiritilganda

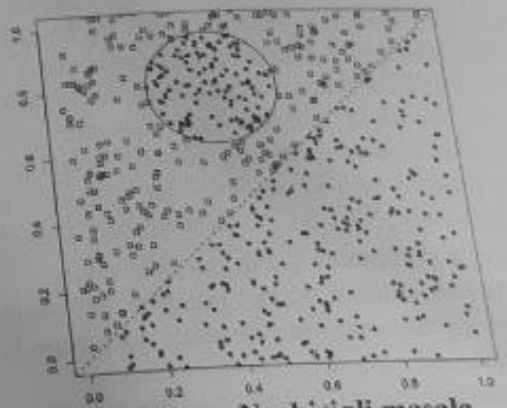
o'rtacha tasniflash xatosini oshiradi. Shunga o'xshab, shovqin hujjati - bu o'quv majmuasiga kiritilganda o'rganish usulini chalg'itadigan va tasniflash xatosini oshiradigan hujjat hisoblanadi. Intuitiv ravishda asosiy tarqatish vakillik maydonini asosan bir xil bo'lgan hududlarga ajratadi.



3.10-rasm. **Shovqin bilan bog'liq chiziqli muammo**

Ushbu faraziy veb-sahifalarni tasniflash stsenariysida faqat xitoy tilidagi veb-sahifalar qattiq doiralar va aralash xitoycha-inglizcha veb-sahifalar kvadratchalardir. Ikki sinf chiziqli sinf chegarasi bilan ajratilgan, uchta shovqin hujjatidan tashqari (strelkalar bilan belgilangan) bir jinsli sinf topshiriqlaridir. O'z hududida hukmron sinfga mos kelmaydigan hujjat shovqin hujjatidir. Shovqin hujjatlari chiziqli tasniflagichni o'qitish qiyin bo'lishining sabablaridan biridir. Agar klassifikatorning qaror giperplanini tanlashda shovqin hujjatlariga juda ko'p e'tibor qaratilsa u yangi ma'lumotlar bo'yicha noto'g'ri hisoblanadi. Asosiysi, qaysi hujjatlar shovqinli hujjatlar ekanligini va shuning uchun potentsial noto'g'ri ekanligini aniqlash odatda qiyin. Agar ikkita sinfni mukammal ajratib turadigan giperplan mavjud bo'lsa ikkita sinf chiziqli ravishda

ajratiladigan deb ataladi. Haqiqatdan ham, agar chiziqli ajralish o'rinli bo'lsa 3.8-rasmda ko'rsatilgandek cheksiz sonli chiziqli ajratgichlar (3.4-mashq) mavjud bo'lib, bu yerda mumkin bo'lgan ajratuvchi giperplanlar soni cheksiz bo'ladi. 3.8-rasmda chiziqli klassifikatorni o'rgatishdagi yana bir qiyinchilik ko'rsatilgan. Agar chiziqli bo'linadigan muammo bilan shug'ullanish kerak bo'ladigan bo'lsa unda o'quv ma'lumotlarini mukammal ajratadigan barcha qarorlar giperplanlari orasidan tanlash mezoniga muhtojlik zesiladi. Umuman olganda ushbu giperplanlarning ba'zilari yangi ma'lumotlarda yaxshi ishlaydi, ba'zilari esa yo'q.



3.11-rasm. **Nochiziqli masala**

Nochiziqli klassifikatorga kNN misol bo'la oladi. KNN ning nochiziqliligi 3.6-rasmga o'xshash misollarni ko'rib chiqayotganda intuitiv ravishda aniq bo'ladi. KNN ning qaror chegaralari (3.6-rasmdagi qo'sh chiziqlar) mahalliy chiziqli segmentlardir, lekin umuman olganda, 2D chiziqqa yoki yuqori o'lchamdagi giperplanga teng bo'lmagan murakkab shaklga ega. 3.11-rasmda nochiziqli muammoning yana bir misoli keltirilgan. Grafikning yuqori chap qismidagi doiraviy "anklav" tufayli **P(d|c)** va **P(d|c)** taqsimotlari orasida yaxshi chiziqli ajratuvchi mavjud emas. Chiziqli klassifikatorlar anklavni noto'g'ri tasniflaydi, kNN kabi chiziqli bo'lmagan tasniflagich esa, agar o'quv majmuasi yetarlicha katta bo'lsa ushbu turdagi muammolar uchun juda aniq bo'ladi. Agar muammo nochiziqli bo'lsa va uning sinf chegaralarini chiziqli giperplanlar bilan yaxshi yaqinlashtirish mumkin bo'lmasa unda chiziqli klassifikatorlarga qaraganda chiziqli bo'lmagan tasniflagichlar ko'pincha

aniqroq bo'ladi. Agar muammo chiziqli bo'lsa, oddiyroq chiziqli tasniflagichdan foydalanish yaxshidir.

3.5. Ikkitadan ortiq sinflar bilan tasniflash

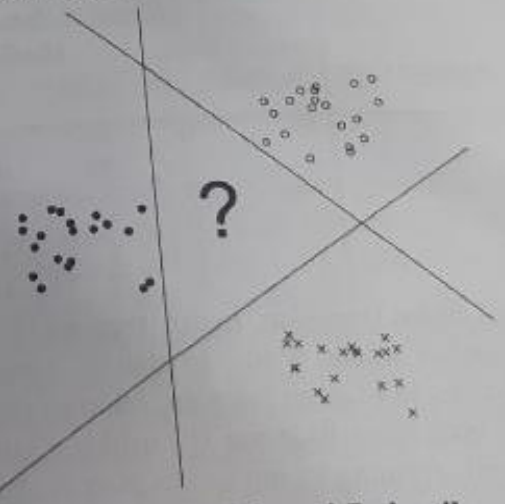
Ikki klassli chiziqli klassifikatorlarni **J > 2** sinfga kengaytirish mumkin. Foydalanish usuli sinflar bir-birini istisno qiladimi yoki yo'qligiga bog'liq. Bir-birini istisno qilmaydigan sinflar uchun tasniflash har qanday, ko'p etiketli yoki ko'p qiymatli tasnif deb ataladi. Bunday holda hujjat bir vaqtning o'zida bir nechta sinfga yoki bitta sinfga tegishli bo'lishi mumkin yoki hech bir sinfga tegishli bo'lmasligi mumkin. Bir sinf bo'yicha qaror boshqalari uchun barcha variantlarni ochiq qoldiradi. Ba'zida sinflar bir-biridan mustaqil deb aytiladi, lekin bu noto'g'ri chunki sinflar aniqlangan ma'noda kamdan-kam statistik mustaqildir. (2.1) tenglamadagi tasniflash muammosining rasmiy ta'rifi nuqtai nazaridan har qanday tasniflashda **J** turli klassifikatorlar o'rganiladi, ularning har biri $\mathbf{c}_j$ yoki $\mathbf{c}_j$ ni qaytaradi: $\gamma_j(d) \in \{c_j, \bar{c}_j\}$. Har qanday tasniflash vazifasini chiziqli klassifikatorlar yordamida hal qilish juda oddiy:

1. Har bir sinf uchun klassifikator tuzing, bunda o'quv majmuasi sinfdagi hujjatlar to'plamidan (ijobiy belgilar) va uning to'ldiruvchisidan (salbiy belgilar) iborat bo'lsin.
2. Sinov hujjatini hisobga olgan holda, har bir tasniflagichni alohida qo'llang.

Bir klassifikatorning qarori boshqa tasniflagichning qaroriga ta'sir qilmaydi. Ikkitadan ortiq sinfga ega bo'lgan ikkinchi turdagi tasniflash bitta tasnifdir. Bu yerda darslar bir-biridan farq qiladi. Har bir hujjat to'liq sinflardan biriga tegishli bo'lishi kerak. Bir klassifikatsiya ko'p nomli, ko'p sinf yoki bir belgili tasnif deb ham ataladi. Rasmiy ravishda bir tasniflashda yagona tasniflash funksiyasi **g** mavjud, uning diapazoni **C**, ya'ni $\gamma_j(d) \in \{c_1, \ldots, c_J\}$. **kNN** klassifikatorning biri hisoblanadi. Haqiqiy muammolardan biri matnda kamroq uchraydi, har qanday muammolardan ko'ra tasniflash qiyin. Buyuk Britaniya, Xitoy, parranda go'shti yoki qahva kabi darslar bilan hujjat bir vaqtning o'zida ko'plab mavzularga tegishli bo'lishi mumkin - Buyuk Britaniya bosh vaziri Xitoyga qahva va parranda savdosi haqida gapirish uchun tashrif buyurganida. Shunga qaramay, 3.1-rasmda keltirirlganidek, sinflar haqiqatan ham bir-birini istisno qilmasa ham, ko'pincha bitta taxminni beradi. Hujjat tilini

identifikatsiyalash bo'yicha tasniflash muammosi uchun bitta taxmin yaxshi yaqinlikdir chunki aksariyat matnlar faqat bitta tilda yozilgan. Bunday hollarda, bitta cheklov qo'yish klassifikatorning samaradorligini oshirishi mumkin chunki klassifikatorlarning birortasi hujjatni hech qanday sinfga yoki bir nechta sinfga tayinlaganligi bilan bog'liq xatolar bartaraf etiladi. **J** giper tekisliklari **R|V|** ni ajratmaydi 3.12-rasmda ko'rsatilganidek, **J** alohida mintaqalarga ajratiladi. Shunday qilib, bitta klassifikatsiya uchun ikki sinfli chiziqli klassifikatorlardan foydalanganda kombinatsiya usulidan foydalanish kerak ekan.

Eng oddiy usul - ***torank*** sinflari va keyin yuqori o'rinli sinfni tanlang. Geometrik jihatdan reyting **J** chiziqli ajratgichlardan masofalarga nisbatan bo'lishi mumkin. Sinf ajratuvchisiga yaqin bo'lgan hujjatlar noto'g'ri tasniflanadi, shuning uchun ajratuvchidan qanchalik uzoq bo'lsa, ijobiy tasniflash qarori to'g'ri ekanligi shunchalik ishonchli bo'ladi. Shu bilan bir qatorda, sinflarni tartiblash uchun to'g'ridan-to'g'ri ishonch o'lchovidan foydalanish mumkin masalan, sinfga a'zolik ehtimoli. Chiziqli klassifikatorlar bilan bitta klassifikatsiya uchun ushbu algoritmni quyidagicha ifodalash mumkin:



3.12- rasm. **J giper tekisliklari fazoni J ajratilgan hududlarga ajratmaydi.**

1. Har bir sinf uchun klassifikator tuzing, bunda o'quv majmuasi sinfdagi hujjatlar to'plamidan (ijobiy belgilar) va uning to'ldiruvchisidan (salbiy belgilar) iborat.

2. Sinov hujjatini hisobga olgan holda har bir tasniflagichni alohida qo'llash.

3. Hujjatni sinfga joylashtirish:
- maksimal ball;
- maksimal ishonch qiymati;
- yoki maksimal ehtimollik.

J > 2 sinflar uchun klassifikatorning ishlashini tahlil qilish uchun muhim vosita chalkashlik matritsasi hisoblanadi. Chalkashlik matritsasi h_{c1}, c_{2i} sinflarining har bir juftligi uchun c_1 dan nechta hujjat borligini ko'rsatadi. c_2 ga noto'g'ri tayinlanganligini anglatadi. 3.5-jadvalda klassifikator uchta moliyaviy toifani - ***pul-fx***, ***savdo*** va ***foiz***larni bug'doy, makkajo'xori va donning uchta qishloq xo'jaligi sinfidan ajrata oladi, lekin bu ikki guruh ichida ko'plab xatolarga yo'l qo'yiladi. Chalkashlik matritsasi tizimning aniqligini oshirish imkoniyatlarini aniqlashga yordam beradi. Masalan, 3.5-jadvaldagi ikkinchi eng katta xatoni hal qilish uchun bug'doy hujjatlarini don hujjatlaridan ajratib turadigan xususiyatlarni kiritishga harakat qilish mumkin.

3.5-jadval. **Reuters-21578 uchun chalkashlik matritsasi.**

true class \ assigned class	*money-fx*	*trade*	*interest*	*wheat*	*corn*	*grain*
money-fx	95	0	10	0	0	0
trade	1	1	90	0	1	0
interest	13	0	0	0	0	0
wheat	0	0	1	34	3	7
corn	1	0	2	13	26	5
grain	0	0	2	14	5	10

Misol. Uch xil tildan (masalan, ingliz, frantsuz, ispan) har biri 100 tadan 300 ta hujjatdan iborat o'quv to'plamini yarating. Xuddi shu protsedura bo'yicha test to'plamini yarating, lekin to'rtinchi tildan 100 ta hujjat qo'shing. (i) bitta klassifikatorni (ii) ushbu o'quv to'plamida har qanday klassifikatorni o'rgating va uni test to'plamida baholang. (iii) Ikki klassifikatorning ushbu vazifada o'zini tutishida qiziqarli farqlar bormi?

3.6. Noto'g'ri o'zgaruvchanlik almashinuvi

Noto'g'ri chiziqli klassifikatorlar chiziqli tasniflagichlarga qaraganda kuchliroqdir. Ba'zi muammolar uchun nolinchi tasniflash

xatosi bo'lgan chiziqli bo'lmagan tasniflagich mavjud, ammo bunday chiziqli klassifikator yo'q. Bu har doim chiziqli bo'lmagandan foydalanishimiz kerakligini anglatadimi? Statistik matnlarni tasniflashda optimal samaradorlik uchun tasniflagichlar mavjudmi?

Bu savollarga javob berish uchun ushbu bo'limda mashinali o'qitishdagi eng muhim tushunchalardan biri bo'lgan qarama-qarshilik almashinuvi kiritiladi. Savdo nima uchun universal optimal o'rganish usuli yo'qligini tushuntirishga yordam beradi? Tegishli o'rganish usulini tanlash shuning uchun matnni tasniflash muammosini hal qilishning muqarrar qismidir. Ushbu bo'lim davomida chiziqli va chiziqli bo'lmagan tasniflagichlardan mos ravishda "kamroq kuchli" va "kuchliroq" o'rganishning prototip namunalari sifatida foydalaniladi. Bu bir necha sabablarga ko'ra soddalashtirilgan. Birinchidan, ko'pgina chiziqli bo'lmagan modellar chiziqli modellarni alohida holat sifatida qabul qiladi. Masalan, kNN kabi chiziqli bo'lmagan o'rganish usuli ba'zi hollarda chiziqli tasniflagichni ishlab chiqaradi. Ikkinchidan, chiziqli modellarga qaraganda kamroq murakkab bo'lgan chiziqli bo'lmagan modellar mavjud. Masalan, ikkita parametrli kvadratik polinom 10000 o'lchovli chiziqli klassifikatordan kamroq kuchga ega. Uchinchidan, o'rganishning murakkabligi klassifikatorning xususiyati emas chunki o'rganishning ko'plab jihatlari (masalan, xususiyat tanlash, tartibga solish va 4-bobda chegarani maksimallashtirish kabi cheklovlar mavjud) mavjud.

O'rganishning yakuniy natijasi bo'lgan klassifikator turiga ta'sir qilmasdan kuchliroq yoki kamroq kuchli ta'lim usuli - bu tasniflagich chiziqli yoki chiziqli bo'lmaganligidan qat'i nazar. Ular o'quvchini 3.7-bo'limda sanab o'tilgan nashrlarga havola qiladi. Ushbu bo'limda chiziqli va chiziqli bo'lmagan tasniflagichlar matnni tasniflashda zaifroq va kuchliroq o'rganish usullari uchun proksi sifatida xizmat qiladi. Avvalo matn tasnifidagi maqsadimizni aniqroq bayon qilishimiz kerak. 2.1-bo'limda test to'plamidagi tasniflash xatosini minimallashtirishni mumkinligi aytilgan edi. Yashirin taxmin shundan iboratki, o'quv hujjatlari va test hujjatlari bir xil asosiy taqsimot bo'yicha ishlab chiqariladi. Ushbu taqsimotni $\mathbf{P}(\mathbf{h_d},\mathbf{c_i})$ bilan belgilanadi, bunda **d** - hujjat va **c** - uning yorlig'i yoki sinfi. 2.4 va 2.5-rasmlar $\mathbf{P}(\mathbf{h_d},\mathbf{c_i})$ ni $\mathbf{P}(\mathbf{c})$ va $\mathbf{P}(\mathbf{d}|\mathbf{c})$ hosilasiga taqsimlaydigan generativ modellarning namunalari edi. 3.10 va 3.11-rasmlarda $\mathbf{d} \in \mathbf{R}_2$ va $\mathbf{c} \in$ {kvadrat, yaxlit doira} bilan $\mathbf{h_d}$, $\mathbf{c_i}$ uchun generativ modellar tasvirlangan. Ushbu bo'limda baholash o'lchovi

sifatida to'g'ri tasniflangan test hujjatlari sonini (yoki teng ravishda, test hujjatlaridagi xatolik darajasini) ishlatish o'rniga, etiketkaning o'ziga xos noaniqligiga javob beradigan baholash o'lchovi qabul qilinadi. Ko'pgina matnlarni tasniflash muammolarida berilgan hujjat ko'rinishi turli sinflarga tegishli hujjatlardan kelib chiqishi mumkin. Chunki turli sinflarga mansub hujjatlar bir xil hujjat ko'rinishida ko'rsatilishi mumkin. Misol uchun, bir jumladan iborat hujjatlar Xitoy Fransiyani sudga beradi va Fransiya Xitoyni sudga beradi, xuddi shu hujjatning tasviri **d'**={Xitoy, Fransiya, sudga tortiladi} so'zlar to'plamida tasvirlangan. Ammo faqat oxirgi hujjat **c'** = Fransiya tomonidan olib borilgan huquqiy harakatlar sinfiga tegishlidir. Ushbu bo'limdagi hisob-kitoblarni soddalashtirish uchun klassifikatorni baholashda testlar to'plamidagi xatolar soni hisoblanadi, balki klassifikator hujjatning sinfda bo'lishining shartli ehtimolligi **P(c|d)** ni qanchalik to'g'ri baholashiga qaraladi. Yuqoridagi misolda ularda **P(c'|d') = 0,5** bo'lishi mumkin. Matnni tasniflashda ularning maqsadi **g** klassifikatorni topishdan iborat bo'lib, **d** hujjatlar bo'yicha o'rtacha olinganda **g(d)** haqiqiy ehtimollik **P(c|d)** ga imkon qadar yaqin bo'ladi. Bu o'rtacha kvadrat xato yordamida o'lchanadi:

$$MSE\ (\gamma) = E_d[\gamma(d) - P(c \mid d)]^2. \tag{3.7}$$

bu yerda $\mathbf{E_d}$ - **P(d)** ga nisbatan kutish. O'rtacha kvadratik xato atamasi **g** tomonidan to'g'ri bo'lmasa yaqin bo'lgan qarorlar uchun qisman kredit beradi.

$$\begin{aligned} E[x-\alpha]^2 &= Ex^2 - 2Ex\,\alpha + \alpha^2 \\ &= (Ex)^2 - 2Ex\,\alpha + \alpha^2 \\ &\quad + Ex^2 - 2(Ex)^2 + (Ex)^2 \\ &= [Ex-\alpha]^2 \\ &\quad + Ex^2 - E2x(Ex) + E(Ex)^2 \\ &= [Ex-\alpha]^2 + E[x-Ex]^2 \end{aligned} \tag{3.8}$$

$$\begin{aligned} E_D E_d[\Gamma_D(d) - P(c \mid d)]^2 &= E_d E_D[\Gamma_D(d) - P(c \mid d)]^2 \\ &= E_d[\,[E_D\Gamma_D(d) - P(c \mid d)]^2 \\ &\quad + E_D[\Gamma_D(d) - E_D\Gamma_D(d)]^2\,] \end{aligned} \tag{3.9}$$

3.13-rasm. **Noaniq-variantli dekompozitsiya uchun arifmetik o'zgarishlar**

(3.9) tenglamani chiqarish uchun (3.8) tenglamada **a** = **P(c|d)** va **x** = **DD(d)** ni o'rnatish mumkin.

Ular **g** klassifikatorni $\mathbf{P(h_d,c_i)}$ taqsimoti uchun optimal deb belgilanadi, agar u MSE(g) ni minimallashtirsa. MSEni minimallashtirish klassifikatorlar uchun desideratumdir. Ularga o'rganish usullari uchun ham mezon kerak. Eslatib o'tamiz, ular **D** o'rganish usulini **D** yorliqli o'quv to'plamini kirish sifatida qabul qiladigan va **g** klassifikatorini qaytaruvchi funksiya sifatida belgilagan edik. O'rganish usullari uchun ular o'quv to'plamlari bo'yicha o'rtacha hisoblangan, minimal MSE bilan **g** tasniflagichlarini o'rganadigan **D** ni topishni maqsad qilib olamiz. Buni o'rganish xatosini minimallashtirish sifatida rasmiylashtirishimiz mumkin:

$$learning\ -error\ (\Gamma) = E_D[MSE\ (\Gamma(D))] \tag{3.7}$$

bu yerda $\mathbf{E_D}$ - belgilangan o'quv majmualaridan kutish. Vaziyatni soddalashtirish uchun o'quv to'plamlari qat'iy belgilangan o'lchamga ega deb taxmin qilish mumkin - $\mathbf{P(h_d,c_i)}$ taqsimoti keyin o'quv to'plamlari bo'yicha **P(D)** taqsimotini belgilaydi. Statistik matnlarni tasniflashda o'rganish usulini tanlash mezoni sifatida o'rganish xatosidan foydalanish mumkin. O'rganish usuli **D**, agar u o'rganish xatosini minimallashtirsa **P(D)** taqsimoti uchun optimal hisoblanadi. Yaxshiroq o'qilishi uchun **D(D)** uchun **DD** yozilsa, (3.7) tenglamani quyidagicha o'zgartirish mumkin:

$$\begin{aligned} learning\ -error\ (\Gamma) &= E_D[MSE\ (\Gamma_D)] && (1.15)\\ &= E_D E_d[\Gamma_D(d) - P(c \mid d)]^2 && (1.16)\\ &= E_d[bias\ (\Gamma, d) + variance\ (\Gamma, d)] \end{aligned}$$

bu yerda (3.10) va (3.11) tenglamalar orasidagi ekvivalentlik 3.13-rasmdagi (3.9) tenglamada ko'rsatilgan. E'tibor bering, **d** va **D** bir-biridan mustaqil. Umuman olganda, tasodifiy **d** hujjati va **D** tasodifiy o'quv to'plami uchun **D** yorliqli **d** nusxasini o'z ichiga olmaydi. Bias - bu **P(c|d)**, **d** ning **c** dagi haqiqiy shartli ehtimolligi va **DD(d)** o'rganilgan klassifikatorning bashorati, o'quv to'plamlari bo'yicha o'rtacha hisoblangan kvadrat farqi. Agar o'rganish usuli doimiy ravishda noto'g'ri bo'lgan tasniflagichlarni ishlab chiqarsa og'ishlar kata bo'ladi. Agar (i) klassifikatorlar doimiy ravishda to'g'ri bo'lsa yoki (ii) turli o'quv to'plamlari turli hujjatlarda xatoliklarga sabab bo'lsa yoki (iii) turli o'quv to'plamlari bir xil hujjatlarda ijobiy va salbiy xatolarga sabab bo'lsa ham

bu o'rtacha 0 ga yaqin bo'ladi. Agar ushbu uchta shartdan biri bajarilsa u holda $E_D G_D(d)$, barcha treninglar = to'plamlar bo'yicha kutish $P(c|d)$ ga yaqin. ***Rokkio*** va ***Naive Bayes*** kabi chiziqli usullar chiziqli bo'lmagan muammolar uchun yuqori moyillikka ega chunki ular faqat bitta turdagi sinf chegarasini, chiziqli giperplanni modellashtirishi mumkin. Agar generativ model $P(h_d,c_i)$ murakkab chiziqli bo'lmagan sinf chegarasiga ega bo'lsa (3.11) tenglamadagi egilish muddati yuqori bo'ladi chunki ko'p sonli nuqtalar doimiy ravishda noto'g'ri tasniflanadi. Misol uchun, 3.11-rasmdagi aylana anklav chiziqli modelga mos kelmaydi va chiziqli tasniflagichlar tomonidan izchil ravishda noto'g'ri tasniflanadi. Klassifikatorga kiritgan domen bilimi (yoki uning yetishmasligi) natijasida og'ish haqida o'ylash mumkin. Agar ikkita sinf o'rtasidagi haqiqiy chegara chiziqli ekanligini bilsak, unda chiziqli tasniflagichlarni ishlab chiqaradigan o'rganish usuli chiziqli bo'lmagan usuldan ko'ra ko'proq muvaffaqiyat qozonadi. Ammo agar haqiqiy sinf chegarasi chiziqli bo'lmasa va klassifikatorni chiziqli bo'lishiga noto'g'ri yondoshsak u holda tasniflash aniqligi o'rtacha past bo'ladi. **kNN** kabi chiziqli bo'lmagan usullar past moyillikka ega bo'ladi. 3.6-rasmda kNN ning qaror chegaralari o'zgaruvchanligini ko'rishimiz mumkin - o'quv majmuasida hujjatlarning taqsimlanishiga qarab, o'rganilgan qaror chegaralari juda katta farq qilishi mumkin. Natijada, har bir hujjat ba'zi o'quv to'plamlari uchun to'g'ri tasniflash imkoniyatiga ega. Shuning uchun o'rtacha bashorat $E_D G_D(d)$ **P (c|d)** ga yaqinroq va chiziqli o'rganish usuliga qaraganda kichikroq. Dispersiya - o'rganilgan klassifikatorlar bashoratining o'zgarishi: $D_D(d)$ va uning o'rtacha $E_D D_D(d)$ o'rtasidagi o'rtacha kvadratchalar farqi. Turli xil o'quv to'plamlari **D** juda turli xil D_D tasniflagichlarini keltirib chiqarsa dispersiya katta bo'ladi. Agar o'quv to'plami D_D qabul qiladigan tasniflash qarorlariga to'g'ri yoki noto'g'ri bo'lishidan qat'i nazar ozgina ta'sir qilsa bu kichikdir. Variantlar qanchalik mos kelmasligi qarorlar to'g'ri yoki noto'g'ri emasligiga mos ravishda o'lchaydi. Chiziqli o'rganish usullari kam dispersiyaga ega chunki tasodifiy chizilgan o'quv majmualarining ko'pchiligi shunga o'xshash qaror giperplanlarini ishlab chiqaradi. 3.10 va 3.11-rasmlardagi chiziqli o'rganish usullari bilan ishlab chiqarilgan qaror chiziqlari o'quv majmuasiga qarab asosiy sinf chegaralaridan biroz chetga chiqadi lekin hujjatlarning katta qismi uchun sinf tayinlanishi (asosiy chegaraga yaqin bo'lganlar bundan mustasno)

ta'sir qilmaydi. 3.11-rasmdagi dumaloq anklav doimiy ravishda noto'g'ri tasniflanadi.

kNN kabi chiziqli bo'lmagan usullar yuqori dispersiyaga ega. 3.6-rasmdan ko'rinib turibdiki, kNN ikkita sinf o'rtasidagi juda murakkab chegaralarni modellashtirishi mumkin. Shuning uchun u 3.10-rasmda tasvirlangan turdagi shovqin hujjatlariga sezgir. Natijada (3.11) tenglamadagi dispersiya atamasi kNN uchun katta: Test hujjatlari ba'zan noto'g'ri tasniflanadi - agar ular o'quv to'plamidagi shovqin hujjatiga yaqin bo'lsa - va ba'zan to'g'ri tasniflanadi - shovqin hujjatlari bo'lmasa ularning yonidagi mashg'ulotlarda bo'lsa. Bu mashg'ulot to'plamidan o'quv majmuasiga qadar yuqori o'zgarishlarga olib keladi. Yuqori tafovutli o'rganish usullari o'quv ma'lumotlarini haddan tashqari moslashtirishga moyil.

Tasniflashdan maqsad, $P(h_d,c_i)$ ning asosiy taqsimotini haqiqiy xususiyatlarini qo'lga kiritadigan darajada o'quv ma'lumotlarini moslashtirishdir. Haddan tashqari moslashishda o'rganish usuli shovqindan ham o'rganadi. Haddan tashqari moslashish MSEni oshiradi va ko'pincha yuqori variatsiyali o'rganish usullari uchun muammo hisoblanadi. Shuningdek, dispersiyani o'rganish usulining model murakkabligi yoki shunga o'xshash tarzda xotira sig'imi deb hisoblashimiz mumkin - o'quv to'plamining tavsifi qanchalik batafsil eslab qolishi va keyin yangi ma'lumotlarga amal qilinadi. Ushbu quvvat o'quv majmuasiga mos keladigan mustaqil parametrlar soniga mos keladi. Har bir kNN mahallasi S_k mustaqil tasniflash qarorini qabul qiladi. Bu holatda parametr 3.7-rasmdagi $P^{\wedge}(c|S_k)$ bahosi hisoblanadi. Shunday qilib, kNN ning imkoniyatlari faqat bilan cheklangan o'quv majmuasining hajmidir. U katta o'quv to'plamlarini yodlashi mumkin. Bundan farqli o'laroq, ***Rocchio*** parametrlari soni qat'iy belgilangan - har bir o'lcham uchun **J** parametr, har bir markaz uchun bitta - o'quv majmuasining o'lchamidan qat'iy nazar. Rokkio klassifikatori (uni belgilaydigan markazlar shaklida) nozik taneli tafsilotlarni "eslab qololmaydi" hamda o'quv majmuasida hujjatlarni taqsimlaydi. (3.7) tenglamaga ko'ra, o'rganish usulini tanlashda ularning maqsadi o'rganish xatosini minimallashtirishdir. Tenglama (3.11) tomonidan qisqacha ifodalanishi mumkin bo'lgan asosiy tushuncha shundan iboratki, o'rganish xatosi ikki komponentga ega: o'rganish xatosi, bir vaqtning o'zida minimallashtirish mumkin emas. Ikkita D_1 va D_2 o'rganish usullarini solishtirganda, ko'p hollarda taqqoslash bir usulda yuqori

og'ish va past dispersiyaga, ikkinchisi esa past og'ishlikka va yuqori dispersiyaga ega bo'ladi. Bitta o'rganish usuli va boshqasiga nisbatan qaror faqat o'quv majmualarida ishonchli tarzda yaxshi klassifikatorlarni ishlab chiqaradigan (kichik dispersiya) yoki juda qiyin qaror chegaralari (kichik tarafkashlik) bilan tasniflash muammolarini o'rganishi mumkin bo'lgan birini tanlash masalasi emas. Buning o'rniga, arizadagi nohaqlik va tafovutning tegishli afzalliklarini baholash va shunga mos ravishda tanlash kerak. Bu ayirboshlash og'ish -variant almashinuvi deb ataladi.

3.10-rasmda biroz o'ylab topilgan illyustratsiya berilgan, ammo kelishuvga misol sifatida foydali bo'ladi. Ba'zi Xitoy matnlarida ***CPU***, ***ONLINE*** va ***GPS*** kabi Rim alifbosida yozilgan inglizcha so'zlar mavjud. Faqat xitoy tilidagi veb-sahifalarni aralash xitoycha-inglizcha veb-sahifalardan ajratish vazifasini ko'rib chiqish mukn. Qidiruv tizimi ingliz tilini bilmaydigan (lekin **CPU** kabi so'zlarni tushunadigan) xitoylik foydalanuvchilarga aralash sahifalarni filtrlash imkoniyatini taklif qilishi mumkin. Ushbu tasniflash vazifasi uchun ikkita xususiyatdan foydalaniladi: Rim alifbosi belgilari soni va veb-sahifadagi xitoycha belgilar soni. Yuqorida aytib o'tilganidek, generativ modelning $\mathbf{P(h_0,c_1)}$ taqsimoti qisqa chiziqdan yuqori (mos ravishda, pastda) eng aralash (mos ravishda xitoy) hujjatlarni yaratadi, biroq bir nechta shovqin hujjatlari mavjud bo'ladi.

3.10-rasmda uchta tasniflagichni ko'rish mumkin:

• Bir xususiyatli tasniflagich. Nuqtali gorizontal chiziq sifatida ko'rsatilgan. Bu klassifikator faqat bitta xususiyatdan, rim alifbosi belgilari sonidan foydalanadi. O'quv majmuasidagi noto'g'ri tasniflar sonini minimallashtiradigan o'rganish usulini nazarda tutsak, gorizontal qaror chegarasining pozitsiyasi o'quv majmuasidagi farqlardan (masalan, shovqin hujjatlari) katta ta'sir ko'rsatmaydi. Shunday qilib, o'rganish usuli bu turdagi tasniflagichni ishlab chiqarish kam dispersiyaga ega ekanligini ko'rsatadi. Ammo uning noto'g'riligi yuqori chunki u pastki chap burchakdagi kvadratlarni va 50 dan ortiq Rim belgilaridan iborat "qattiq doira" hujjatlarini noto'g'ri tasniflaydi.

• Chiziqli klassifikator. Uzoq chiziqlar bilan kesilgan chiziq sifatida ko'rsatilgan. Chiziqli klassifikatorlarni o'rganish kamroq og'ishishga ega chunki faqat shovqinli hujjatlar va ehtimol ikki sinf o'rtasidagi chegaraga yaqin bir nechta hujjatlar noto'g'ri tasniflanadi. Dispersiya bir xususiyatli klassifikatorlarga qaraganda yuqori lekin baribir kichik: uzun tire bilan kesilgan chiziq ikki sinf o'rtasidagi haqiqiy chegaradan bir oz og'adi,

shuning uchun o'quv majmualaridan o'rganilgan deyarli barcha chiziqli qaror chegaralaridir. Shunday qilib, juda oz sonli hujjatlar (sinf chegarasiga yaqin hujjatlar) nomuvofiq tarzda tasniflanadi.

• "***Fit-trening-set-perfectly***" tasniflagichi. Qattiq chiziq sifatida ko'rsatilgan. Bu yerda o'rganish usuli o'quv majmuasidagi sinflarni mukammal ajratib turadigan qaror chegarasini quradi. Bu usul eng past og'ishishga ega chunki doimiy ravishda noto'g'ri tasniflanadigan hujjat yo'q - klassifikatorlar ba'zan hatto test to'plamida shovqinli hujjatlarni ham olishadi. Ammo bu o'rganish usulining farqi yuqori. Shovqin hujjatlari qaror chegarasini o'zboshimchalik bilan siljitishi mumkinligi sababli o'quv majmuasidagi shovqin hujjatlariga yaqin bo'lgan test hujjatlari noto'g'ri tasniflanadi - chiziqli o'rganish usuli buni amalga oshirishi dargumon.

Eng mashhur matn tasniflash algoritmlarining ko'pchiligi chiziqli bo'lishi ajablanarli. Ushbu usullarning Ba'zilari, xususan, chiziqli SVMlar, muntazam logistik regressiya va muntazam chiziqli regressiya eng samarali ma'lum usullardan biridir. Noto'g'ri kelishmovchiliklar almashinuvi ularning muvaffaqiyati haqida tushuncha beradi. Matnlarni tasniflashdagi tipik sinflar murakkab va ular chiziqli tarzda yaxshi modellashtirilishi dargumon. Biroq, bu sezgi odatda matn ilovalarida duch keladigan yuqori o'lchamli bo'shliqlar uchun noto'g'ri. O'lchovlilikning oshishi bilan chiziqli bo'linish ehtimoli tez oshadi. Shunday qilib, yuqori o'lchamli fazolardagi chiziqli modellar, ularning chiziqliligiga qaramay juda kuchli hisoblanadi. Hatto kuchliroq nochiziqli o'rganish usullari murakkabroq qaror chegaralarini modellashtirishi mumkin.

Giperplandan ko'ra o'quv ma'lumotlaridagi shovqinga ham sezgirdir. Nochiziqli ta'lim usullari ba'zan o'quv majmuasi katta bo'lsa yaxshiroq ishlaydi, lekin har qanday holatda emas.

3- bob bo'yicha foydalanilgan adabiyotlar

Ferragina, Paolo, and Rossano Venturini.
2007.
Compressed permuterm indexes.
In *Proc. SIGIR*. ACM Press.
Forman, George.

2004.
A pitfall and solution in multi-class feature selection for text classification.
In *Proc. ICML*.
Forman, George.
2006.
Tackling concept drift by temporal inductive transfer.
In *Proc. SIGIR*, pp. 252-259. ACM Press.
DOI: *doi.acm.org/10.1145/1148170.1148216*.
Forman, George, and Ira Cohen.
2004.
Learning from little: Comparison of classifiers given little training.
In *Proc. PKDD*, pp. 161-172.
Fowlkes, Edward B., and Colin L. Mallows.
1983.
A method for comparing two hierarchical clusterings.
Journal of the American Statistical Association 78 (383): 553-569.
URL: *www.jstor.org/view/01621459/di985957/98p0926l/0*.
Fox, Edward A., and Whay C. Lee.
1991.
FAST-INV: A fast algorithm for building large inverted files.
Technical report, Virginia Polytechnic Institute & State University, Blacksburg, VA, USA.
Kernighan, Mark D., Kenneth W. Church, and William A. Gale.
1990.
A spelling correction program based on a noisy channel model.
In *Proc. ACL*, pp. 205-210.
King, Benjamin.
1967.
Step-wise clustering procedures.
Journal of the American Statistical Association 69: 86-101.
Kishida, Kazuaki, Kuang-Hua Chen, Sukhoon Lee, Kazuko Kuriyama, Noriko Kando, Hsin-Hsi Chen, and Sung Hyon Myaeng.
2005.
Overview of CLIR task at the fifth NTCIR workshop.
In *NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*. National Institute of Informatics.

Klein, Dan, and Christopher D. Manning.
2002.
Conditional structure versus conditional estimation in NLP models.
In *Proc. Empirical Methods in Natural Language Processing*, pp. 9-16.
Kleinberg, Jon M.
1997.
Two algorithms for nearest-neighbor search in high dimensions.
In *Proc. ACM Symposium on Theory of Computing*, pp. 599-608. ACM Press.
DOI: *doi.acm.org/10.1145/258533.258653*.
Kleinberg, Jon M.
1999.
Authoritative sources in a hyperlinked environment.
JACM 46 (5): 604-632.
URL: *citeseer.ist.psu.edu/article/kleinberg98authoritative.html*.
Kleinberg, Jon M.
2002.
An impossibility theorem for clustering.
In *Proc. NIPS*.

3- bob bo‘yicha nazariy va amaliy test savollari

1. Indeksning o‘ziga xos xususiyati nimalar bilan belgilanadi?
A) Izlash sifati va sifatli ma’lumotlarning sonini qidirish mexanizmidagi ma’lum shartlar soniga nisbati bilan
B) Hujjat miqdori va (yoki) so‘rovning qidiruv shakli va (yoki) qidiruv shakli, shuningdek, qidiruv tizimiga kiritilgan aniq atamalar va matnda mavjud bo‘lgan ma’lumotlar soniga kiritilgan hujjat yoki so‘rov
C) Faqat rasmiy protseduralarni hisoblash uchun faqat rasmiy protseduralar yordamida hisoblash orqali amalga oshiriladigan hujjat yoki so‘rovni qayta ishlash uchun qo‘llaniladi
D) To‘g‘ri javob yo‘q
2. Izlash sifati va sifatli ma’lumotlarning sonini qidirish mexanizmidagi ma’lum shartlar soniga nisbatideganda nimalar tushuniladi?
A) Indeksning o‘ziga xos xususiyati
B) To‘liqlikni indekslash
C) Bepul indekslash

D) To‘g‘ri javob yo‘q

3. Tez-tez o‘zgarib turadigan to‘plamlar va to‘plamda o‘zgarishlar bo‘lishi uchun qanday indekslash talab qilinadi?
   A) Dinamik
   B) To‘liqlikni indekslash
   C) Bepul indekslash
   D) To‘g‘ri javob yo‘q

4. Tez-tez ishlatiladigan disk ma’lumotlarini asosiy xotirada saqlash texnikasi qanday nomlanadi?
   A) Keshlash
   B) Qidiruv vaqti
   C) Buffer
   D) To‘g‘ri javob yo‘q

5. Diskni o‘qish yoki yozishni bajarayotganda, diskning ma’lumotlar joylashgan qismiga o‘tish biroz vaqt oladi. Bu jarayon qanday nomlanadi?
   A) Qidiruv vaqti
   B) Keshlash
   C) To‘liqlikni indekslash
   D) To‘g‘ri javob yo‘q

6. Qidiruv vaqtida odatdagi disklar uchun o‘rtacha necha ms ni tashkil qiladi?
   A) 5 ms
   B) 10 ms
   C) 2 ms
   D) 8 ms

7. Operatsion tizimlar odatda butun bloklarni o‘qiydi va yozadi. Bloklarning 8, 16, 32 va 64 kilobayt (KB) hajmi keng tarqalgan. Blokni o‘qish yoki yozish paytida asosiy xotiraning qaysi qismida bajariladi?
   A) Buffer
   B) Qidiruv vaqti
   C) Keshlash
   D) To‘g‘ri javob yo‘q

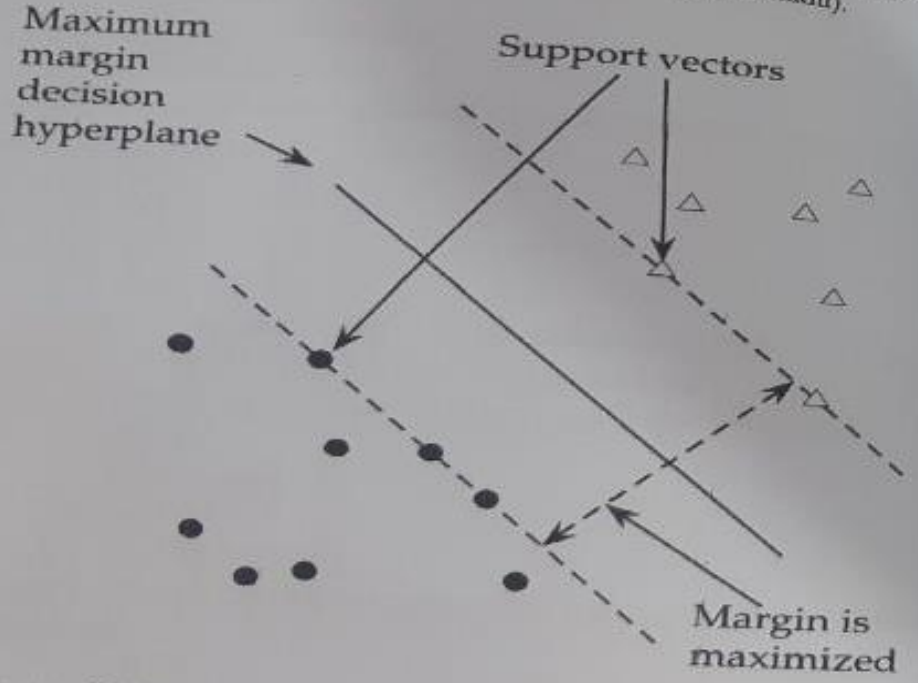
8. Indeks tuzishni samaraliroq qilish uchun atamalarni qanday usuldan foydalanamiz?
   A) termID
   B) Qidiruv vaqti
   C) Keshlash
   D) To‘g‘ri javob yo‘q

9. termID – nima?
   A) Indeks tuzish uchun atamalarni samaraliroq qilish usuli
   B) Blokni o‘qish yoki yozish bajarilish paytida asosiy xotiraning qismi
   C) Blokni o‘qish yoki yozish bajarilish paytida tarmoqning qismi
   D) To‘g‘ri javob yo‘q

10. Buffer – nima?
   A) Blokni o‘qish yoki yozish bajarilish paytida asosiy xotiraning qismi
   B) Butun to‘plam qayta ishlanmaguncha qayta-qayta chaqiriladigan tokenlash oqimi
   C) Indeks tuzish uchun atamalarni samaraliroq qilish usuli
   D) To‘g‘ri javob yo‘q

IV BOB. HUJJATLAR BILAN ISHLASHDA MASHINALI O‘QITISH VA VEKTORLI MASHINALARNI QO‘LLANILISHI

Tasniflagich samaradorligini oshirish so‘nggi yigirma yil ichida mashinali o‘qitish bo‘yicha intensiv tadqiqotlar sohasi bo‘ldi va bu - qo‘llab-quvvatlovchi vektor mashinalari, mustahkamlangan qarorlar daraxtlari, muntazam logistik regressiya kabi zamonaviy klassifikatorlarning yangi avlodiga olib keldi. Neyron tarmoqlarini vujudga keldi. Ushbu usullarning ko‘pchiligi shu jumladan, ushbu bobning asosiy mavzusi bo‘lgan vektorli mashinalarni qo‘llanilishi, axborotni qidirish muammolari xususan, matn tasnifi uchun muvaffaqiyatli qo‘llanilgan. **SVM** bu katta hajmdagi klassifikatorning bir turi: bu vektor fazosiga asoslangan mashinali o‘qitish usuli bo‘lib, maqsad ikkita sinf o‘rtasida o‘quv ma’lumotlarining istalgan nuqtasidan maksimal darajada uzoqda bo‘lgan qarorlar chegarasini topishdir (ehtimol, ba’zi nuqtalarni chetlab o‘tish deb hisoblash mumkin).

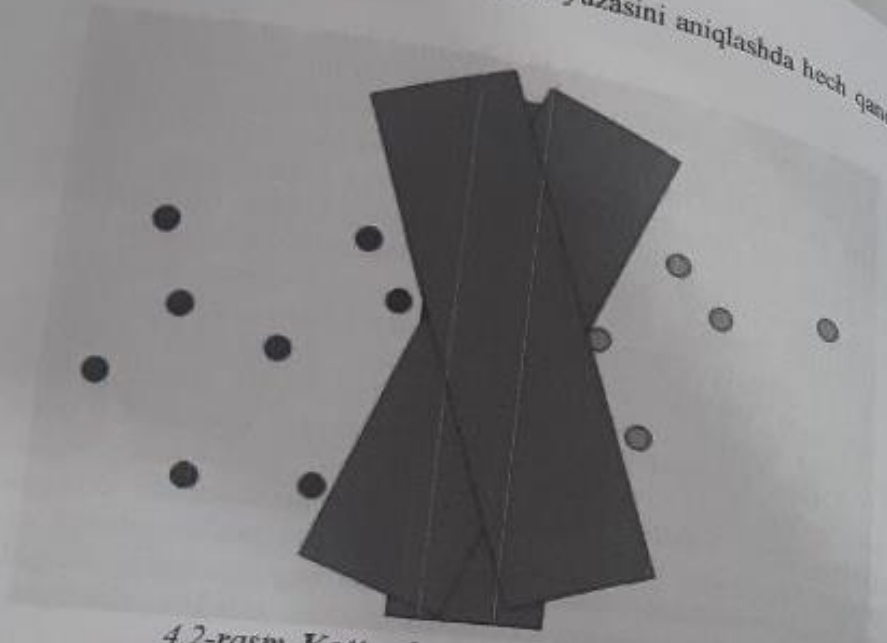


4.1- rasm. **Qo‘llab-quvvatlovchi vektorlar klassifikatorning chetidan 5 ball yuqorida joylashganlik grafigi**

4.1. Vektorli mashinalarni qo‘llanilishi

Dastlab chiziqli klassifikator (4.1-bo‘lim) bilan ajratiladigan ikki toifali ma’lumotlar to‘plamlari uchun SVM-lar ko‘rib chiqiladi va ishlab chiqiladi, so‘ngra 4.2-bo‘limdagi modelni ajratilmaydigan ma’lumotlar, ko‘p sinfli muammolar va chiziqli bo‘lmagan modellarga kengaytiriladi, shuningdek, SVM ishlashi bo‘yicha qo‘shimcha muhokamalar taqdim etiladi. 4.3-bo‘limda matn klassifikatorlarining amaliy qo‘llanilishini ko‘rib chiqishga o‘tiladi: qaysi turdagi tasniflagichlar qachon mos keladi va tasniflashda domenga xos matn xususiyatlaridan qanday foydalanish mumkin? Va nihoyat, matnni tasniflash uchun yaratgan mashinali o‘qitish texnologiyasini maxsus qidiruvda hujjatlarni tartiblashni o‘rganish muammosiga qanday qo‘llanilishi mumkinligi ko‘rib chiqiladi (4.4-bo‘lim). Ushbu vazifani bajarish uchun bir nechta mashinali o‘qitish usullari qo‘llanilgan bo‘lsa-da, SVM-lardan foydalanish muhim ahamiyatga ega. Vektor mashinalari boshqa mashinalarni o‘rgatish usullaridan yaxshiroq emas (ehtimol, kam ta’lim ma’lumotlariga ega bo‘lgan holatlar bundan mustasno), lekin ular eng zamonaviy darajada ishlaydi va juda dolzarb nazariy va empirik jihatdan juda katta ahamiyatga ega.Chiziqli bo‘linadigan holat bu - ikki sinfli, ajratiladigan o‘quv ma’lumotlar to‘plami uchun masalan, 4.8-rasmdagi ko‘plab mumkin bo‘lgan chiziqli ajratgichlar mavjud. Intuitiv ravishda, ikkita sinfning ma’lumot elementlari orasidagi bo‘shliqning o‘rtasida chizilgan qaror chegarasi bir yoki ikkala sinf misollariga juda yaqin bo‘lganidan ko‘ra yaxshiroq ko‘rinadi. ***Perseptron*** algoritmi kabi ba’zi o‘rgatish usullari (4.7-bo‘lim) har qanday chiziqli ajratgichni topsa, boshqalari, masalan, ***Naive Bayes*** ba’zi mezonlarga ko‘ra eng yaxshi chiziqli ajratuvchini qidiradi. SVM, xususan, har qanday ma’lumot nuqtasidan maksimal darajada uzoqda joylashgan qaror yuzasini izlash mezonini belgilaydi. Qaror yuzasidan eng yaqin ma’lumotlar nuqtasigacha bo‘lgan bu masofa tasniflagichning chegarasini aniqlaydi. Qurilishning ushbu usuli albatta, SVM uchun qaror funksiyasi ajratuvchining o‘rnini belgilaydigan ma’lumotlarning (odatda kichik) kichik to‘plami tomonidan to‘liq aniqlanganligini anglatadi. Bu nuqtalar ***qo‘llab-quvvatlovchi vektorlar*** deb ataladi (vektor fazosida nuqtani boshlang‘ich va bu nuqta orasidagi vektor sifatida ko‘rish mumkin). 4.1-rasmda misol muammosi uchun chegara va qo‘llab-quvvatlash vektorlari ko‘rsatilgan. Boshqa

...nuqtalari tanlangan qaror yuzasini aniqlashda hech qanday rol o'ynamaydi.



4.2-rasm. **Katta chegaralarni tasniflash**

Katta chegarada turib olish modelning imkoniyatlarini pasaytiradi: yoq qaror yuzasi joylashtirilishi mumkin bo'lgan burchaklar diapazoni qaror giperplaniga qaraganda kichikroq (3.8-rasm).

Chegarani maksimal darajada oshirish yaxshi ko'rinadi, chunki qaror yuzasiga yaqin nuqtalar juda noaniq tasniflash qarorlarini ifodalaydi. Klassifikatorning har qanday holatda ham qaror qabul qilish ehtimoli deyarli 50%. Katta chegaraga ega bo'lgan klassifikator past aniqlikdagi tasniflash qarorlarini qabul qilmaydi. Bu sizga tasniflashning xavfsizlik chegarasini beradi. O'lchovdagi yengil xato yoki hujjatning ozgina o'zgarishi noto'g'ri tasnifga olib kelmaydi. SVMlarni rag'batlantiruvchi yana bir sezgi 4.2-rasmda ko'rsatilgan. Qurilish bo'yicha, SVM klassifikatori qaror chegarasi atrofida katta chegarani talab qiladi. Qaror qabul qiluvchi giperplan bilan solishtirganda agar siz sinflar o'rtasida ma'lumot ajratish moslamasini joylashtirish kerak bo'lsa, uni qaerga qo'yish mumkinligi haqida kamroq tanlov mavjud. Natijada, modelning xotira sig'imi pasaygan va shuning uchun uning test ma'lumotlarini

to'g'ri umumlashtirish qobiliyatini oshirish kerak (3-bobdagi noaniqlik-variant almashinuvi muhokamasi). Keling, algebra bilan SVMni rasmiylashtiramiz. Qaror giperplaniyasini kesishish atamasi **b** va ***giper*** tekislikka perpendikulyar bo'lgan qaror ***giper tekisligi*** normal vektor $\vec{w}$ bilan aniqlash mumkin. Ushbu vektor odatda mashinali o'qitish adabiyotida ***og'irlik vektori*** deb ataladi.

Normal vektorga perpendikulyar bo'lgan barcha gipertekisliklar orasida **b** kesishma termini belgilanadi. Giper tekislik normal vektorga perpendikulyar bo'lganligi uchun giper tekislikdagi barcha $\vec{w}^T\vec{x} = -b$ ni qanoatlantiradi. Endi ularda o'quv ma'lumotlarning $D = \{(\vec{x}_i, y_i)\}$ nuqtalari to'plami bor deylik, bu yerda har bir a'zo bir juft $\vec{x}_i$ nuqta va unga mos keladigan y_i sinf yorlig'i. SVMlar uchun ikkita ma'lumot sinfi har doim +1 va −1 deb nomlanadi (1 va 0 o'rniga) va kesishish atamasi har doim aniq **b** sifatida xususiyati bo'yicha ifodalanadi (har doim qo'shimcha og'irlik vektoriga $\vec{w}$ ko'paytirish o'rniga). Agar hamma narsa shu tarzda qilinsa, matematika yanada aniq ishlaydi chunki u deyarli darhol funktsional chegara ta'rifida ko'riladi. Chiziqli klassifikator u holda:

$$f(\vec{x}) = sign\ (\vec{w}^T\vec{x} + b) \qquad (4.1)$$

−1 qiymati bir sinfni, +1 qiymati esa boshqa sinfni bildiradi.

Agar nuqta qaror chegarasidan uzoqda bo'lsa, uni tasniflashda ishonchimiz komil. Berilgan ma'lumotlar to'plami va qaror giperplaniyasi uchun i-misolning funksional chegarasini i^* gipertekislikka nisbatan $\vec{x}_i, y_i(\vec{w}^T\vec{x}_i + b)$ miqdori sifatida belgilanadi. Ma'lumotlar to'plamining qaror yuzasiga nisbatan funktsional chegarasi minimal funktsional chegaraga ega bo'lgan ma'lumotlar to'plamidagi har qanday nuqtaning funktsional chegarasidan ikki baravar ko'p bo'ladi (2 koeffitsienti chegaraning butun kengligi bo'ylab o'lchashdan kelib chiqadi, masalan: 4.3-rasm). Biroq bu ta'rifdan foydalanishda muammo bor: qiymat cheklangan chunki $\vec{w}$ va **b** ni oddiygina kattalashtirish orqali har doim funktsional chegarani xohlaganimizcha kattalashtira olamiz. Misol uchun, agar $\vec{w}$ ni 5 $\vec{w}$ va **b** ni **5b** ga almashtirsak, u holda funktsional chegara $y_i(5\vec{w}^T\vec{x}_i + 5b)$ besh marta katta bo'ladi. Bu $\vec{w}$ vektorining o'lchamiga Ba'zi cheklovlar qo'yishimiz kerakligini ko'rsatadi. Buni qanday qilishni tushunish uchun keling, haqiqiy geometriyani ko'rib chiqaylik. $\vec{x}'$ nuqtadan qaror chegarasigacha bo'lgan ***Yevklid masofasi*** qancha? 4.3-rasmda bu masofa **r** bilan belgilanadi. Bilamizki, nuqta va gipertekislik orasidagi eng qisqa masofa tekislikka perpendikulyar va

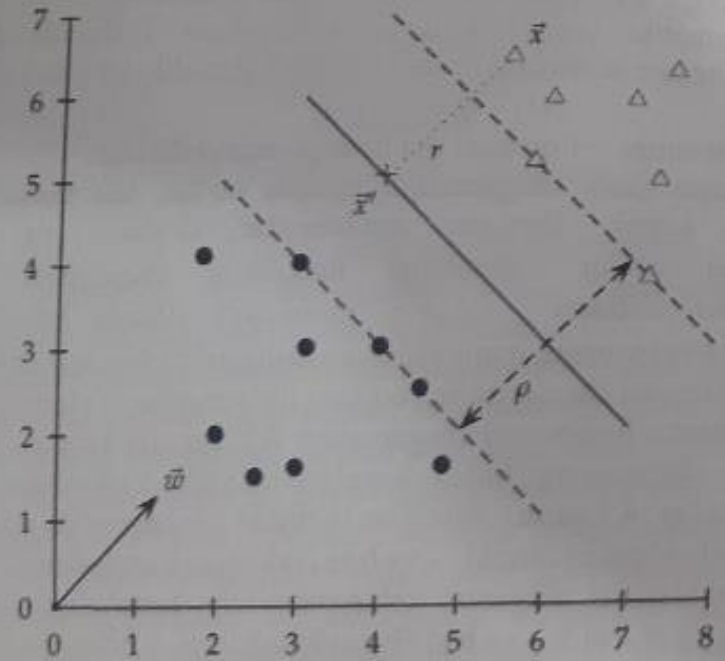
shuning uchun $\vec{w}$ ga parallel. Ushbu yo'nalishdagi birlik vektori $\vec{w}/|\vec{w}|$ hisoblanadi. Keyin diagrammadagi nuqta chiziq $r\vec{w}/|\vec{w}|$ vektorining tarjimasi. $\vec{x}'$ ***Giper tekislikdagi*** **~x** ga eng yaqin nuqtani $\vec{x}'$ deb belgilanadi. Keyin:

$$\vec{x}' = \vec{x} - yr\frac{\vec{w}}{|\vec{w}|}$$

deb belgilanadi. (4.2)

Bu yerda **y** ga ko'paytirish qaror yuzasining har ikki tomonida joylashgan $\vec{x}$ ning ikkita holati uchun belgini o'zgartiradi. Bundan tashqari, $\vec{x}'$ qaror chegarasida yotadi.

3.1-bo'limda muhokama qilinganidek, vektor fazodagi nuqtalarning umumiy holatini keltiramiz, lekin agar nuqtalar uzunligi normalangan hujjat vektorlari bo'lsa unda barcha harakat birlik shar yuzasida sodir bo'ladi va qaror yuzasi sharning sirtini kesib o'tadi.



4.3-rasm. **Nuqtaning geometrik chegarasi (r) va qaror chegarasi (r)**

Shuning uchun $\sim\mathbf{w}^T \sim\mathbf{x}' + \mathbf{b} = 0$ ni qanoatlantiradi. Demak:

$$\vec{w}^T(\vec{x} - yr\frac{\vec{w}}{|\vec{w}|}) + b = 0 \quad (4.3)$$

r uchun yechim beradi:

$$r = y\frac{\vec{w}^T\vec{x} + b}{|\vec{w}|} \quad (4.4)$$

Shunga qaramay, ajratuvchi giperplanega eng yaqin nuqtalar qo'llab-quvvatlovchi vektorlardir. Tasniflagichning geometrik chegarasi ikkita sinfning qo'llab-quvvatlovchi vektorlarini ajratib olish mumkin bo'lgan chiziqning maksimal kengligidir. Ya'ni, (4.4) tenglamada berilgan **r** uchun ma'lumotlar nuqtalari bo'yicha minimal qiymatdan ikki baravar ko'p yoki ekvivalent tarzda 4.2-rasmda ko'rsatilgan belgilar ajratgichlardan birining maksimal kengligi. Geometrik chegara parametrlarni masshtablashda aniq o'zgarmasdir: agar $\vec{w}$ ni 5 $\vec{w}$ va **b** ni **5b** ga almashtirsak, geometrik chegara bir xil bo'ladi chunki u tabiatan **~w** uzunligi bilan normallashtiriladi. Bu shuni anglatadiki, ular geometrik chekkaga ta'sir qilmasdan **~w** ga istalgan masshtab cheklovini qo'yishimiz mumkin. Boshqa tanlovlar qatorida 6-bobdagi kabi birlik vektorlaridan ham foydalanishimiz mumkin, buning uchun $\vec{w} = 1$. Bu geometrik chegarani funktsional chegara bilan bir xil qilish ta'siriga ega bo'ladi.

Buni eslang

$$|\vec{w}| = \sqrt{\vec{w}^T\vec{w}}$$

Funktsional chegarani xohlagancha o'lchashimiz mumkinligi sababli, katta ***SVMlarni*** yechishda qulaylik yaratish uchun keling, barcha ma'lumotlar nuqtalarining funktsional chegarasi kamida 1 bo'lishini va kamida bitta ma'lumot vektori uchun u 1 ga teng bo'lishini talab qilaylik. Ya'ni, ma'lumotlarning barcha elementlari uchun:

$$y_i(\vec{w}^T\vec{x} + b) \geq 1 \quad (4.5)$$

va tengsizlik tenglik bo'lgan qo'llab-quvvatlovchi vektorlar mavjud. Har bir misolning gipertekislikdan masofasi $r_i = y_i(\vec{w}^T\vec{x}_i + b)/|\vec{w}|$ geometrik chekka $\rho = 2/|\vec{w}|$. Bizning xohishimiz hali ham bu geometrik chegarani maksimal darajada oshirishdir. Ya'ni, $\vec{w}$ va **b** ni topmoqchimiz:

- $\rho = 2/|\vec{w}|$ maksimalla shtirlgan
- *Barcha* $(\vec{x}_i, y_i) \in D$ *uchun* , $y_i(\vec{w}^T\vec{x}_i + b) \geq 1$

Maksimal **2/|~w|** minimallashtirish bilan bir xildir **|~w|/2**. Bu minimallashtirish muammosi sifatida SVMning yakuniy standart formulasini beradi: (4.6) **~w** va **b** toping:

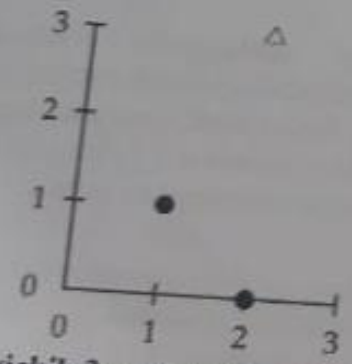
(4.6)

- $\frac{1}{2}\vec{w}^T\vec{w}$ minimallas htirilgan
- *Barcha* $\{(\vec{x}_i, y_i)\}$ *uchun* , $y_i(\vec{w}^T\vec{x}_i + b) \geq 1$

Endi ular chiziqli cheklovlarga bog'liq kvadratik funksiyani optimallashtirmoqdamiz. Kvadrat optimallashtirish masalalari matematikaviy optimallashtirish masalalarining standart, taniqli sinfi bo'lib, ularni yechish uchun ko'plab algoritmlar mavjud. Printsipial jihatdan standart kvadratik dasturlash (QP) kutubxonalari yordamida *SVM* ni qurishimiz mumkin edi ammo bu sohada **SVM** dan paydo bo'ladigan **QP** turining tuzilishidan foydalanishga qaratilgan ko'plab tadqiqotlar olib borildi. Natijada, deyarli hamma modellarni yaratishda foydalanadigan **SVM**-larni yaratish uchun yanada murakkablashdi ammo tezroq va kengaytiriladigan kutubxonalar mavjud. Bu yerda bunday algoritmlarning tafsilotlarini keltirmaymiz. Biroq bunday optimallashtirish muammosini hal qilish shaklini tushunish uchun quyidagi narsalar foydali bo'ladi. Yechim ikkilamchi masalani tuzishni o'z ichiga oladi, bunda ***Lagrange*** ko'paytmasi a_i boshlang'ich masaladagi har bir $y_i(\vec{w}^T\vec{x}_i + b) \geq 1$ cheklov bilan bog'langan:

- $\sum_i a_i y_i = 0$
- $a_i \geq 0$ barcha $1 \leq i \leq N$ uchun

Yechim quyidagi shaklda bo'ladi: (4.7)



4.4-rasm. **SVM uchun kichik 3 ma'lumot nuqtasining o'quv to'plami**

$$\vec{w} = \sum_i a_i y_i \vec{x}_i$$

$$b = y_k - \vec{w}^T\vec{x}_k \text{ barbir } \vec{x}_k \text{ uchun } a_k \neq 0 \quad (4.8)$$

Yechimda a_i ning ko'p qismi nolga teng. Har bir nolga teng bo'lmagan a_i mos keladigan $\vec{x}_i$ i qo'llab-quvvatlovchi vektor ekanligini ko'rsatadi. Tasniflash funksiyasi quyidagicha hisoblanadi:

$$f(\vec{x}) = sign\left(\sum_i a_i y_i \vec{x}_i^T \vec{x} + b\right) \quad (4.9)$$

Ikkilik masalada maksimallashtiriladigan atama ham, tasniflash funksiyasi ham nuqtalar juftligi ($\vec{x}$ *va* $\vec{x}_i$ *yoki* $\vec{x}_i$ *va* $\vec{x}_1$)) orasidagi nuqta mahsulotini o'z ichiga oladi va bu ma'lumotlardan foydalanishning yagona usulidir. Buning ahamiyati haqida keyinroq to'xtalib o'tamiz. Xulosa qilish uchun trening ma'lumotlar to'plamidan boshlaymiz. Ma'lumotlar to'plami eng yaxshi ajratuvchi giperplanni noyob tarzda belgilaydi va ushbu tekislikni topish uchun ma'lumotlarni kvadratik optimallashtirish protsedurasi orqali yuboramiz. Tasniflash uchun yangi $\vec{x}$ nuqtasi berilgan bo'lsa, (4.1) yoki (4.9) tenglamadagi $\mathbf{f}(\vec{x})$tasniflash funksiyasi nuqtaning giperplanet normaliga proyeksiyasini hisoblaydi. Ushbu funksiyaning belgisi nuqtaga tayinlanadigan sinfni belgilaydi. Agar nuqta tasniflagichning chegarasida bo'lsa (yoki tasniflash xatolarini minimallashtirishga qaror qilgan bo'lishimiz mumkin bo'lgan boshqa ishonch chegarasi **t**) bo'lsa, tasniflagich ikkita sinfdan birini emas, balki "bilmayman" ni qaytarishi mumkin. $\mathbf{f}(\vec{x})$qiymati ham tasniflash ehtimoliga aylantirilishi mumkin. Qiymatlarni o'zgartirish uchun sigmasimon o'rnatish standart hisoblanadi. Bundan tashqari, ***marj*** doimiy bo'lgani uchun agar model turli manbalardan olingan o'lchamlarni o'z ichiga olsa, Ba'zi o'lchamlarni ehtiyotkorlik bilan qayta o'zgartirish talab qilinishi mumkin. Biroq, ularning hujjatlarimiz (nuqtalarimiz) birlik ***gipersferasida*** bo'lsa, bu muammo emas.

Misol. Ko'rsatilgan (juda kam) ma'lumotlar to'plamida **SVM** yaratishni ko'rib chiqing.

Misol. Geometrik tarzda ishlaganda shunga o'xshash misol uchun maksimal chegara og'irlik vektori ikki sinfning eng qisqa bog'lovchi nuqtalariga ya'ni (1.1) va (2.3) orasidagi chiziqqa parallel bo'lib, og'irlik vektorini beradi. Optimal qaror yuzasi bu chiziqqa ortogonal bo'lib, uni yarmida kesib o'tadi.

Shuning uchun u (1.5, 2) orqali o'tadi. Shunday qilib, SVM qaror chegarasi:

$$y = x_1 + 2x_2 - 5.5$$

$y_i(\vec{w}^T\vec{x}_i + b) \geq 1$ni belgilovchi standart cheklov bilan algebraik tarzda ishlagan holda, ular $|\vec{w}|$bilan belgilanadi. Bu cheklov ikkita qo'llab-quvvatlovchi vektor tomonidan tenglik bilan qondirilganda sodir bo'ladi. Bundan tashqari, Ba'zi **a** uchun yechim $\vec{w} = (a, 2a)$ ekanligini bilamiz. Shunday qilib, ularda shunday holatlar bor:

$$a + 2a + b = -1$$

$$2a + 6a + b = 1$$

Shuning uchun **a = 2/5** va **b = -11/5**. Shunday qilib, optimal ***giperplane*** $\vec{w}$ - (2/5, 4/5) va **b = -11/5** bilan berilgan.

r chegarasi $2/|\vec{w}| = 2/\sqrt{4/25 + 16/25} = 2/(2\sqrt{5}/5) = \sqrt{5}$ Bu javobni geometrik jihatdan 4.4-rasmni tekshirish orqali tasdiqlash mumkin.

Misol. Ma'lumotlar to'plami uchun (har bir sinf misollarini o'z ichiga olgan) bo'lishi mumkin bo'lgan qo'llab-quvvatlash vektorlarining minimal soni qancha?

Misol. ***SVMlarda*** yadrolardan foydalana olishning asosi (4.2.3-bo'limga qarang) tasniflash funksiyasini (4.9) tenglama ko'rinishida yozish mumkin (bu yerda katta muammolar uchun a_i ning ko'pchiligi 0 ga teng). Yuqoridagi misoldagi ma'lumotlar to'plami uchun tasniflash funksiyasini ushbu shaklda qanday yozish mumkinligini aniq ko'rsating. Ya'ni, ma'lumotlar nuqtalari paydo bo'ladigan va yagona o'zgaruvchi $\sim x$ bo'lgan funksiya sifatida **f** ni yozing.

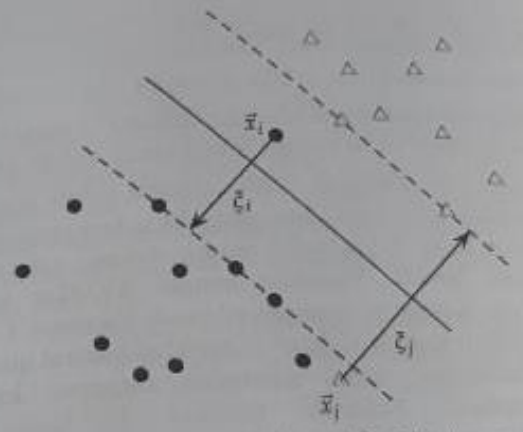
Misol. **SVMlight** (http://svmlight.joachims.org/) kabi SVM paketini o'rnating va 4.1-misolda muhokama qilingan ma'lumotlar to'plami uchun SVM yarating. Dastur matn bilan bir xil yechimni berishiga ishonch hosil qiling. SVMlight yoki bir xil o'quv ma'lumotlari formatini qabul qiladigan boshqa paket uchun o'quv fayli quyidagicha bo'ladi:

```
+1 1:2 2:3
-1 1:2 2:0
-1 1:1 2:1
```

Keyin SVMlight uchun trening buyrug'i quyidagicha bo'ladi:

svm_learn -c 1 - a alphas.dat train.dat model.dat

-c opsiyasi 4.2.1-bo'limda muhokama qiladigan bo'sh o'zgaruvchilardan foydalanishni o'chirish uchun kerak. Og'irlik vektorining normasi 4.1-misolda topilgan narsaga mos kelishini tekshiring. a_i qiymatlarini o'z ichiga olgan **alphas.dat** faylini ko'rib chiqing.



4.5-rasm. **Sust o'zgaruvchilar bilan katta marj tasnifi**

4.2. SVM modeli uchun kengaytmalar

4.2.1. Marja tasnifi

Matnni tasniflashda keng tarqalgan juda yuqori o'lchamli muammolar uchun Ba'zida ma'lumotlar chiziqli ravishda ajratilishi mumkin. Ammo umumiy holatda ular bunday emas va ular bunday bo'lsa ham bir nechta g'alati shovqin hujjatlarini e'tiborsiz qoldirib, ma'lumotlarning asosiy qismini yaxshiroq ajratib turadigan yechimni afzal ko'rishimiz mumkin. Agar **D** o'quv to'plami chiziqli ravishda ajratilmasa, standart yondashuv model qarorining chegarasi bir nechta xatolarga yo'l qo'yishdir (Ba'zi nuqtalar - chetga chiqish yoki shovqinli misollar - chegaraning ichida yoki noto'g'ri tomonida). Keyin ular har bir noto'g'ri tasniflangan misol uchun to'lovni to'laymiz, bu (4.5) tenglamada berilgan marja talabini qondirishdan qanchalik uzoqligiga bog'liq. Buni amalga oshirish uchun ***slack*** o'zgaruvchilarni kiritamiz p_i. ξ_i uchun nolga teng bo'lmagan qiymat x_i qiymatiga proportsional xarajat bo'yicha marj talabini qondirmaslikka imkon beradi. 4.5-rasmga qarang. Bo'sh o'zgaruvchilar bilan SVM optimallashtirish muammosining formulasi quyidagicha hisoblanadi:

Find $\vec{w}, b,$ *and* $\xi_i \geq 0$ *such that* :

- $\frac{1}{2}\vec{w}^T\vec{w} + C\sum_i \xi_i$ *is* min *imized*
- *and for all* $\{(\vec{x}_i, y_i)\}$, $y_i(\vec{w}^T\vec{x}_i + b) \geq 1 - \xi_i$ (4.10)

Optimallashtirish muammosi shundan iboratki, u ***marj*** ni qanchalik to'liq qilishi mumkinligi va bu marjga ruxsat berish uchun qancha ballni siljitish kerakligi ξ_i 0 ni o'rnatish orqali $\vec{x}_i$ nuqtasi uchun chegara 1 dan kam bo'lishi mumkin, lekin keyin buni qilganlik uchun minimallashtirish uchun C_i jarima to'laydi. l_i yig'indisi mashg'ulotdagi xatolar sonining yuqori chegarasini beradi. Yumshoq marjali SVM'lar ***marj*** bilan almashtirilgan o'quv xatolarini minimallashtiradi. Parametr **C** - tartibga solish atamasi bo'lib, u haddan tashqari o'rnatishni nazorat qilish usulini ta'minlaydi. Kattalashgani sayin, geometrik chegarani kamaytirish hisobiga ma'lumotlarni o'chirishi mumkin. U kichik bo'lsa, bo'shashmasdan o'zgaruvchilardan foydalangan holda Ba'zi ma'lumotlar nuqtalarini hisobga olish va ma'lumotlarning asosiy qismini modellashtirish uchun to'liq chegarani joylashtirish oson. Yumshoq ***marj***ni tasniflash uchun ikkita muammo mavjud:

Find $\alpha_1, \ldots \alpha_N$ *such that* $\sum \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j$ (4.11)

- $\sum_i \alpha_i y_i = 0$
- $0 \leq \alpha_i \leq C$ *for all* $1 \leq i \leq N$

Ikkilamchi masalada na bo'sh o'zgaruvchilar p_i, na ular uchun ***Lagrange multiplikatorlari*** ko'rinmaydi. Qo'llab-quvvatlovchi vektor ma'lumotlar nuqtalari uchun ***Lagrange multiplikatorlari***ning mumkin bo'lgan o'lchamini chegaralovchi doimiy **C** qoladi. Avvalgidek, nol bo'lmagan a_i bilan $\vec{x}_i$ qo'llab-quvvatlovchi vektorlar bo'ladi. Ikkilamchi masalaning yechimi quyidagi ko'rinishda bo'ladi:

$$\vec{w} = \sum \alpha y_i \vec{x}_i \qquad (4.12)$$

$$b = y_k(1 - \xi_k) - \vec{w}^T\vec{x}_k \quad \text{for } k = \arg\max_k \alpha_k$$

Tasniflash uchun yana $\vec{w}$ aniqlash kerak emas, buni (4.9) tenglamada bo'lgani kabi ma'lumotlar nuqtalari bo'lgan nuqta mahsulotlari nuqtai nazaridan amalga oshirish mumkin. Odatda, qo'llab-quvvatlash vektorlari o'quv ma'lumotlarining kichik bir qismi bo'ladi. Biroq, agar muammo ajratib bo'lmaydigan yoki kichik chegara bilan bo'lsa, noto'g'ri tasniflangan yoki chegara ichidagi har bir ma'lumot nuqtasi nolga teng

bo'lmagan a_i ga ega bo'ladi. Agar ushbu nuqtalar to'plami katta bo'lib qolsa, 4.2.3-bo'limda ko'rib chiqiladigan nochiziqli holat uchun bu sinov vaqtida **SVM** lardan foydalanishda katta pasayish bo'lishi mumkin. ***Chiziqli SVMlar*** bilan o'qitish va sinovdan o'tkazishning murakkabligi 4.1-jadvalda ko'rsatilgan. ***SVMni o'qitish*** vaqti asosiy **QP**ni yechish vaqtiga bog'liq va shuning uchun nazariy va empirik murakkablik uni hal qilish uchun ishlatiladigan usulga qarab o'zgaradi. QPni yechishning standart natijasi shundan iboratki, ma'lumotlar to'plamining o'lchamida kubik vaqt talab etiladi. SVM o'rgatish bo'yicha barcha so'nggi ishlar bu murakkablikni kamaytirishga harakat qildi, ko'pincha taxminiy yechimlar bilan qoniqish hosil qildi.

4.1-jadval. **Turli tasniflagichlarni o'qitish va sinovdan o'tkazishning murakkabligi, shu jumladan SVM**

Classifier	Mode	Method	Time complexity
NB	training		$\Theta(\|\mathbf{D}\|L_{ave} + \|\mathbf{C}\|\|V\|)$
NB	testing		$\Theta(\|\mathbf{C}\|M_a)$
Rocchio	training		$\Theta(\|\mathbf{D}\|L_{ave} + \|\mathbf{C}\|\|V\|)$
Rocchio	testing		$\Theta(\|\mathbf{C}\|M_a)$
kNN	training	preprocessing	$\Theta(\|\mathbf{D}\|L_{ave})$
kNN	testing	preprocessing	$\Theta(\|\mathbf{D}\|M_{ave}M_a)$
kNN	training	no preprocessing	$\Theta(1)$
kNN	testing	no preprocessing	$\Theta(\|\mathbf{D}\|L_{ave}M_a)$
SVM	training	conventional	$O(\|\mathbf{C}\|\|\mathbf{D}\|^3 M_{ave})$; $\approx O(\|\mathbf{C}\|\|\mathbf{D}\|^{1.7} M_{ave})$, empirically
SVM	training	cutting planes	$O(\|\mathbf{C}\|\|\mathbf{D}\|M_{ave})$
SVM	testing		$O(\|\mathbf{C}\|M_a)$

Trening - bu o'rganish usuli **D** ga nisbatan tasniflagichni o'rganish uchun ketadigan vaqt, test esa bitta hujjatni tasniflash uchun klassifikatorni oladi. SVMlar uchun ko'p sinfli tasniflash |**C**| to'plami tomonidan amalga oshiriladi deb taxmin qilinadi bir-ikki dam olish klassifikatorlari. $\mathbf{L_{ave}}$ - har bir hujjatdagi tokenlarning o'rtacha soni, **Mave** esa hujjatning o'rtacha lug'ati (nolga teng bo'lmagan xususiyatlar soni). $\mathbf{L_a}$ va $\mathbf{M_a}$ test hujjatidagi mos ravishda tokenlar va turlarning raqamlari.

Odatda, empirik murakkablik $\mathbf{O(|D|1.7)}$ ga teng. Shunga qaramay, an'anaviy SVM algoritmlarining o'ta chiziqli o'qitish vaqti ularni juda katta o'quv ma'lumotlar to'plamlarida ishlatishni qiyinlashtiradi yoki imkonsiz qiladi. O'quv misollari soni bo'yicha chiziqli bo'lgan muqobil an'anaviy **SVM** yechim algoritmlari matn muammolarining yana bir

standart atributi bo'lgan ko'plab xususiyatlar mavjud. Biroq, tekislikni kesish texnikasiga asoslangan yangi o'qitish algoritmi bu masalaga istiqbolli javob beradi chunki o'quv misollari soni va misollardagi nolga teng bo'lmagan xususiyatlar soni bo'yicha chiziqli ish vaqtiga ega. Shunga qaramay, kvadratik optimallashtirishni amalga oshirishning haqiqiy tezligi *Naive Bayes* modelidagi kabi oddiygina atamalarni hisoblashdan ancha sekinroq bo'lib qolmoqda. Keyingi bo'limda bo'lgani kabi SVM algoritmlarini chiziqli bo'lmagan **SVM** larga kengaytirish standart ravishda o'qitish murakkabligini $|\mathbf{D}|$ faktoriga oshiradi. Amalda yuqori darajali xususiyatlarni amalga oshirish va chiziqli **SVM.4** ni o'rgatish ko'pincha arzonroq bo'lishi mumkin.

4.2.2. Ko'p sinfli SVMlar

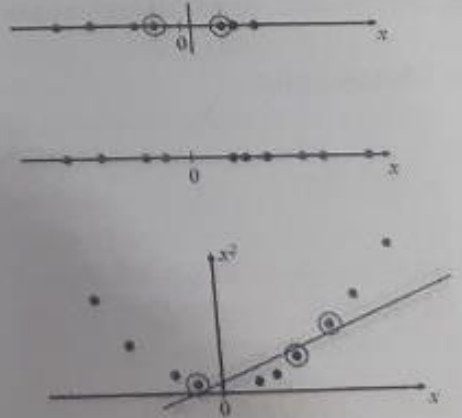
SVMlar tabiatan ikki toifali tasniflagichlardir. SVMlar bilan ko'p sinflarni tasniflashning an'anaviy usuli 3.5-bo'limda muhokama qilingan usullardan birini qo'llashdir. Xususan, amaliyotda eng keng tarqalgan texnika $|\mathbf{C}|$ qurish bo'lgan birga qarshi klassifikatorlar (odatda "birga qarshi hamma" yoki OVA tasnifi deb ataladi) va test ma'lumotlarini eng katta chegara bilan tasniflaydigan sinfni tanlashdir. Yana bir strategiya to'plamni yaratishdir hamda birga qarshi klassifikatorlar va eng ko'p tasniflagichlar tomonidan tanlangan sinfni tanlashdir. Bu $|\mathbf{C}|(|\mathbf{C}|-1)/2$ klassifikatorlarini yaratish bilan bog'liq bo'lsa-da, klassifikatorlarni o'qitish vaqti haqiqatda qisqarishi mumkin chunki har bir klassifikator uchun o'quv ma'lumotlari to'plami ancha kichik. Biroq, bu ko'p sinfli muammolarni hal qilish uchun juda oqlangan yondashuvlar emas. Yaxshiroq alternativa ko'p sinfli SVMlarni qurish orqali taqdim etiladi, bunda kirish xususiyatlaridan va ma'lumotlar sinfidan tashkil topgan juftlikdan olingan $\Phi(\vec{x},y)$ xususiyat vektori ustidan ikki klassli klassifikator quramiz. Sinov vaqtida tasniflagich $\vec{w}^T\Phi(\vec{x},y')$ sinfini tanlaydi. Mashg'ulot vaqtidagi chegara to'g'ri sinf va eng yaqin boshqa sinf uchun bu qiymat orasidagi bo'shliqdir va shuning uchun kvadratik dasturni shakllantirish, $\forall i\ \forall y \neq y_i\ \vec{w}^T\Phi(\vec{x}_i,y_i)-\vec{w}^T\Phi(\vec{x}_i,y) \geq 1-\xi_i$ bo'lishini talab qiladi. Ushbu umumiy usul har xil turdagi chiziqli klassifikatorlarning ko'p sinfli formulasini berish uchun kengaytirilishi mumkin. Bu, shuningdek, tasnifni umumlashtirishning oddiy misolidir, bunda sinflar faqat mustaqil, toifali belgilar to'plami emas, balki ular o'rtasida

aniqlangan munosabatlarga ega bo'lgan ixtiyoriy tuzilgan obyektlar bo'lishi mumkin. SVM dunyosida bunday ish tizimli SVMlar yorlig'i ostida keladi. Ularni 4.4.2-bo'limda yana bir bor eslatib o'tamiz.

4.2.3. Nochiziqli SVMlar

Ular hozirgacha taqdim etgan narsalarga ko'ra, chiziqli ravishda ajratiladigan ma'lumotlar to'plamlari (ehtimol, bir nechta istisnolar yoki ba'zi shovqinlar bilan) yaxshi ishlov beriladi. Ammo agar ma'lumotlar to'plami chiziqli klassifikator tomonidan tasniflashga imkon bermasa, nima qilinadi? Keling, bir o'lchovli ishni ko'rib chiqaylik. 4.6-rasmdagi eng yuqori ma'lumotlar to'plami to'g'ridan-to'g'ri chiziqli tasniflagich bilan tasniflanadi, ammo o'rta ma'lumotlar to'plami emas. Buning o'rniga intervalni tanlay olishimiz kerak. Ushbu muammoni hal qilishning usullaridan biri ma'lumotlarni yuqori o'lchamli fazoga joylashtirish va undan keyin yuqori o'lchamli fazoda chiziqli klassifikatordan foydalanishdir. Masalan, rasmning pastki qismi shuni ko'rsatadiki, agar ma'lumotlarni ikki o'lchamga solishtirish uchun kvadratik funksiyadan foydalansak (qutb koordinatalarini proyeksiya qilish boshqa imkoniyat bo'ladi) chiziqli ajratuvchi ma'lumotlarni osongina tasniflashi mumkin.

Xususiyatlarni moddiylashtirish yuqori tartib va o'zaro ta'sir shartlarini to'g'ridan-to'g'ri hisoblashni va keyin ularni chiziqli modelga qo'yishni anglatadi.



4.6-rasm. **Chiziqli bo'linmaydigan ma'lumotlarni yuqori o'lchamli fazoga proyeksiya qilish**

Chiziqli bo'linmaydigan ma'lumotlarni yuqori o'lchamli fazoga proyeksiya qilish uni chiziqli ravishda ajratish mumkin bo'lishi mumkin.

Umumiy g'oya asl xususiyat maydonini o'quv to'plamini ajratish mumkin bo'lgan yuqori o'lchamli xususiyat maydoniga solishtirishdir. Albatta, buni ma'lumotlar nuqtalari o'rtasidagi bog'liqlikning tegishli o'lchamlarini saqlaydigan usullar bilan qilishni xohlaymiz, natijada klassifikator hali ham yaxshi umumlashtirilishi kerak. SVMlar va boshqa qator chiziqli klassifikatorlar "yadro hiylasi" deb ataladigan yuqori o'lchamli bo'shliqqa ushbu xaritalashni amalga oshirishning oson va samarali usulini ta'minlaydi. Bu haqiqatan ham hiyla emas: u ular ko'rgan matematikadan foydalanadi. SVM chiziqli tasniflagichi ma'lumotlar nuqtasi vektorlari orasidagi nuqta mahsulotiga tayanadi $(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \vec{x}_j$ bo'lsin. Keyin ular hozirgacha ko'rgan tasniflagich:

$$f(\vec{x}) = \text{sign}\left(\sum_i \alpha_i y_i K(\vec{x}_i, \vec{x}) + b\right) \tag{4.13}$$

Endi har bir ma'lumot nuqtasini qandaydir o'zgartirish orqali yuqori o'lchamli fazoga solishga qaror qilaylik. $\Phi: \vec{x} \mapsto \phi(\vec{x})$ Keyin nuqta mahsulotga aylanadi. $\phi(\vec{x}_i)^T \phi(\vec{x}_j)$ Agar ushbu nuqta mahsulotini (bu shunchaki haqiqiy raqam) dastlabki ma'lumotlar nuqtalari nuqtai nazaridan sodda va samarali hisoblash mumkinligi aniqlansa, aslida xaritalashimiz shart emas edi $\vec{x} \mapsto \phi(\vec{x})$. Aksincha, shunchaki miqdorni hisoblashimiz va keyin (4.13) tenglamadagi funksiya qiymatidan foydalanishimiz mumkin $K(\vec{x}_i, v) = \phi(\vec{x}_i)^T \phi(\vec{x}_j)$. Yadro funksiyasi **K** - bu Ba'zi kengaytirilgan xususiyatlar maydonidagi nuqta mahsulotiga mos keladigan funksiya.

4.2-misol: Ikki o'lchamli yadro.

vectors $\vec{u} = (u_1 \quad u_2)$, $\vec{v} = (v_1 \quad v_2)$, consider $K(\vec{u}, \vec{v}) = (1 + \vec{u}^T \vec{v})^2$. We wish to show that this is a kernel, i.e., that $K(\vec{u}, \vec{v}) = \phi(\vec{u})^T \phi(\vec{v})$ for some ϕ. Consider $\phi(\vec{u}) = (1 \quad u_1^2 \quad \sqrt{2}u_1 u_2 \quad u_2^2 \quad \sqrt{2}u_1 \quad \sqrt{2}u_2)$. Then:

$$\begin{aligned} K(\vec{u}, \vec{v}) &= (1 + \vec{u}^T \vec{v})^2 \\ &= 1 + u_1^2 v_1^2 + 2u_1 v_1 u_2 v_2 + u_2^2 v_2^2 + 2u_1 v_1 + 2u_2 v_2 \\ &= (1 \quad u_1^2 \quad \sqrt{2}u_1 u_2 \quad u_2^2 \quad \sqrt{2}u_1 \quad \sqrt{2}u_2)^T (1 \quad v_1^2 \quad \sqrt{2}v_1 v_2 \quad v_2^2 \quad \sqrt{2}v_1 \quad \sqrt{2}v_2) \\ &= \phi(\vec{u})^T \phi(\vec{v}) \end{aligned} \tag{4.14}$$

$$K(\vec{u}, \vec{v}) = (1 + \vec{u}^T \vec{v})^2 = 1 + \phi(\vec{u})^T \phi(\vec{v})$$

Funksional tahlil tilida qanday yadro funksiyalari mavjud? Yadro funksiyalari ba'zan aniqroq ***Merser yadrolari*** deb ataladi chunki ular

Merser shartini qanoatlantirishi kerak: $\int g(\vec{x})^2 d\vec{x}$ chekli bo'lgan har qanday $g(\vec{x})$ uchun ularda quyidagilar bo'lishi kerak:

$$\int K(\vec{x}, \vec{z}) g(\vec{x}) g(\vec{z})^2 d\vec{x} d\vec{z} \geq 0 \tag{4.15}$$

Yadro funksiyasi **K** uzluksiz, simmetrik bo'lishi va musbat aniq grannm matritsaga ega bo'lishi kerak. Bunday **K**, takror ishlab chiqaruvchi yadroning ***Gilbert fazosining*** (Hilbert fazosi nuqta hosilalari ostida yopilgan vektor fazosi) xaritalash mavjudligini bildiradi, shunday qilib u yerdagi nuqta mahsuloti **K** funksiyasi bilan bir xil qiymatni beradi. Agar yadro ***Merser*** talablarini qoniqtirmasligi shart, keyin tegishli **QP** hech qanday yechimga ega bo'lmasligi mumkin. Agar siz ushbu masalalarni yaxshiroq tushunishni istasangiz, 4.5-bo'limda keltirilgan ***SVMlar*** haqidagi kitoblarga murojaat qilishingiz kerak. Aks holda, yadrolar bilan ishlashning 90 foizi quyida tavsiflangan va yaroqli yadrolarni belgilaydigan ikkita vektorning ikkita to'g'ridan-to'g'ri funksiyalar guruhidan birini ishlatishini bilish bilan kifoyalanishingiz mumkin. Yadrolarning keng tarqalgan ikkita oilasi ***ko'p nomli yadrolar*** va ***radial asosli funksiyalardir***. Ko'p nomli yadrolar **K(~x, ~z) = (1 +~x T~z) d** ko'rinishida bo'ladi. d = 1 holati chiziqli yadro bo'lib, ushbu bo'lim boshlanishidan oldin shunday bo'lgan edi. d = 2 holati kvadratik yadro beradi va juda tez-tez ishlaydi. 4.2-misolda kvadrat yadroni tasvirlab berdik. Radial funksiyaning eng keng tarqalgan shakli ***Gauss taqsimoti*** bo'lib, u quyidagicha hisoblanadi:

$$\int K(\vec{x}, \vec{z}) = e^{-(\vec{x}-\vec{z})^2/(2\sigma^2)} \tag{4.16}$$

Radial asosli funksiya (**rbf**) ma'lumotlarni cheksiz o'lchovli ***Gilbert fazosiga solishtirishga*** teng, shuning uchun kvadrat yadroni qilganimizdek, ***radial asos funksiyasini*** aniq tasvirlay olmaymiz. Ushbu ikki oiladan tashqari, boshqa yadrolarni ishlab chiqish bo'yicha qiziqarli ishlar ham bor edi, ularning Ba'zilari matn ilovalari uchun foydalidir. Xususan, matn yadrolari (4.5-bo'limga qarang). ***SVMlar*** dunyosi o'ziga xos til bilan birga keladi, bu esa mashinali o'rgatishda qo'llaniladigan tildan ancha farq qiladi. Terminologiya matematikada chuqur ildizlarga ega, ammo bu terminologiyadan qo'rqmaslik kerak. Darhaqiqat, juda oddiy narsalar haqida gapiramiz. Ko'phadli yadro ularga xususiyatli birikmalarni modellashtirish imkonini beradi (ko'phadning tartibiga qadar). Ya'ni, agar biz alohida so'zlar bilan emas, balki, ehtimol,

operatsion VA tizimi yoki etnik VA tozalash kabi mavzu tasnifi haqida o'ziga xos ma'lumot beradigan juft so'zlarning paydo bo'lishini modellashtirishni istasak unda kvadratikdan foydalanishimiz kerak. Agar so'zlarning uch marta kelishi aniq ma'lumot beradigan bo'lsa kub yadrosidan foydalanishimiz kerak. Bir vaqtning o'zida siz asosiy funksiyalarning vakolatlarini ham olasiz.

Ko'pgina matn ilovalari, ehtimol, bu foydali emas, lekin faqat matematika bilan birga keladi va umid qilamanki, zarar qilmaydi. ***Radial asosli funksiya*** sizga doiralarni (gipersferalarni) ajratib turadigan xususiyatlarga ega bo'lish imkonini beradi - garchi bir nechta bunday xususiyatlar o'zaro ta'sir qilganda qaror chegaralari ancha murakkablashadi. ***Matn yadrosi*** atamalarning belgilar qatorlari bo'lgan xususiyatlarga ega bo'lish imkonini beradi. Bularning barchasi to'g'ridan-to'g'ri tushunchalar bo'lib, ular boshqa ko'plab joylarda turli nomlar ostida ishlatilgan.

4.2.4. Eksperimental natijalar

2.6-bo'limda SVM juda samarali matn tasniflagichi ekanligini ko'rsatadigan natijalarni taqdim etilgan edi. ***Dumais*** va boshqalarning natijalari 2.9-jadvalda keltirilgan SVMlar eng yaxshi ko'rsatkichlarni aniq ko'rsatadi. Bu matn tasnifi bo'yicha SVMlarning kuchli obro'sini o'rnatgan bir qancha ishlardan biri edi. Matnlarni tasniflash uchun SVMlarni o'lchash va baholash bo'yicha yana bir ish Joachims tomonidan bajarilgan edi. Uing natijalarini 4.2-jadvalda keltiramiz.

4.2-jadval. **SVM klassifikatorining zararsizligi**

	NB	Roc-chio	Dec. Trees	kNN	linear SVM $C = 0.5$	linear SVM $C = 1.0$	rbf-SVM $\sigma \approx 7$
earn	96.0	96.1	96.1	97.8	98.0	98.2	98.1
acq	90.7	92.1	85.3	91.8	95.5	95.6	94.7
money-fx	59.6	67.6	69.4	75.4	78.8	78.5	74.3
grain	69.8	79.5	89.1	82.6	91.9	93.1	93.4
crude	81.2	81.5	75.5	85.8	89.4	89.4	88.7
trade	52.2	77.4	59.2	77.9	79.2	79.2	76.6
interest	57.6	72.5	49.1	76.7	75.6	74.8	69.1
ship	80.9	83.1	80.9	79.8	87.4	86.5	85.8
wheat	63.4	79.4	85.5	72.9	86.6	86.8	82.4
corn	45.2	62.2	87.7	71.4	87.5	87.8	84.6
microavg.	72.3	79.9	79.4	82.6	86.7	87.5	86.4

SVM klassifikatorining zararsizligi F_1 dan tortib, natijalar Reuters 21578 ma'lumotlar to'plamida 10 ta eng katta toifalar va barcha 90 toifadagi mikroo'rtacha ishlash uchun ko'rsatilgan.

Yoaxims Dumais boshqalardan farqli o'laroq, juda ko'p atama xususiyatlaridan foydalangan. U ***MI funksiyalarini*** tanlashdan (2.5.1-bo'lim) juda cheklangan miqdordagi xususiyatlar bilan tasniflagichlarni yaratish uchun foydalangan. Chiziqli SVM ning muvaffaqiyati ***Naive Bayes*** kabi boshqa chiziqli yondashuvlar bo'yicha 3.6-bo'limda muhokama qilingan natijalarni aks ettiradi. ***Oddiy atama funksiyalari*** bilan ishlash uzoq yo'lni olishi mumkin. Xuddi shu mashinali o'rgatish usullari bo'yicha turli hujjatlarning natijalari qanchalik farq qilishi yana bir bor sezilarli. Xususan, boshqa tadqiqotchilarning takrorlashlariga asoslanib, ***Naive Bayes*** natijalari juda zaif ko'rinadi va 2.9-jadvaldagi natijalar vakili sifatida qabul qilinishi kerak.

4.3. Matnli hujjatlarni tasniflash masalalari

Tijorat dunyosida matnlarni tasniflashning ko'plab ilovalari mavjud. Elektron pochta spamlarini filtrlash, ehtimol, hozir eng keng tarqalgan. Jekson va Muliner (2002) shunday deb yozadi: "Hujjatlarni mazmuni bo'yicha avtomatik ravishda tasniflash imkoniyatining tijorat qiymati haqida hech qanday savol yo'q. Korporativ intranetlar, davlat idoralari va internet-nashriyotlar uchun bunday imkoniyatlarning ko'plab potentsial ilovalari mavjud.

Ushbu natijalar zararsizlik F_1(7.4-bo'lim) da keltirilgan. Ko'pgina tadqiqotchilar matn tasnifini baholash uchun ushbu o'lchovni afzal ko'rmaydilar chunki uni hisoblash tizimning haqiqiy parametrlarini o'rnatishdan ko'ra interpolyatsiyani o'z ichiga olishi mumkin va nima uchun bu qiymat maksimal F_1 yoki aniqlik/eslab qolish egri chizig'idagi boshqa nuqta emas, balki xabar qilinishi kerakligi aniq emas balki topshirilgan vazifa bilan rag'batlantiriladi. Oldingi natijalar (Joachims 1998) yuqori tartibli polinom yoki rbf yadrolaridan foydalanish orqali ushbu vazifada sezilarli yutuqlarni taklif qilgan bo'lsa-da, bu qattiq chegarali SVMlar bilan sodir bo'ldi. Yumshoq chegarali SVMlar bilan standart C = 1 bilan oddiy chiziqli SVM eng yaxshi ishlaydi.

Tasniflash haqidagi munozaralarimizning aksariyati tasnifga tegishli matnli hujjatlarning o'ziga xos xususiyatlarini muhokama qilish o'rniga,

turli xil mashinali o'qitish usullarini joriy etishga qaratilgan. Ushbu noto'g'ri tushuncha darslik uchun mos keladi, lekin dastur ishlab chiqaruvchisi uchun noto'g'ri. Ko'pincha bitta mashinani o'rgatish usulidan boshqasiga o'tishdan ko'ra, domenga xos matn xususiyatlaridan foydalanish orqali ko'proq samaradorlikka erishish mumkin. Jekson va Muliner (2002) shuni ta'kidlaydi: "Ma'lumotlarni tushunish muvaffaqiyatli toifalashning kalitlaridan biridir, ammo bu toifalash vositalarining ko'p sotuvchilari juda zaif bo'lgan sohadir. Bozordagi "bir o'lcham hammaga mos" vositalarining aksariyati keng turdagi kontent turlarida sinovdan o'tkazilmagan." Ushbu bo'limda bir oz orqaga chekinishni va matn tasnifining ilovalarini, mumkin bo'lgan yechimlar maydonini va amaliy evristikaning foydaliligini ko'rib chiqmoqchimiz.

4.3.1. Klassifikatordan foydalanishlarni tanlash

Matn klassifikatorini yaratish zaruriyatiga duch kelganda, birinchi savol - hozirda qancha ta'lim ma'lumotlari mavjud? Yo'qmi? Juda kam? Juda ko'pmi? Yoki har kuni o'sib borayotgan katta miqdormi? Haqiqiy ilovalarda mashinali o'qitish klassifikatorini qo'llashda eng katta amaliy muammolardan biri bu yetarli darajada o'quv ma'lumotlarini yaratish yoki olishdir. Ko'pgina muammolar va algoritmlar uchun yuqori samarali tasniflagichni ishlab chiqarish uchun har bir sinfdan yuzlab yoki minglab misollar talab qilinadi va ko'plab real dunyo kontekstlari katta toifalar to'plamini o'z ichiga oladi. Dastlab klassifikator imkon qadar tezroq zarur deb hisoblanadi. Agar amalga oshirish uchun ko'p vaqt mavjud bo'lsa uning katta qismi ma'lumotlar resurslarini yig'ishga sarflanishi mumkin. Agar sizda yorliqli ta'lim ma'lumotlari bo'lmasa va ayniqsa ma'lumotlar sohasi haqida ma'lumotga ega bo'lgan xodimlar mavjud bo'lsa, qo'lda yozilgan qoidalardan foydalanishni hech qachon unutmasligingiz kerak. Ya'ni, siz 2-bobning boshida aytib o'tganimizdek, doimiy so'rovlarni yozasiz. Masalan:

AGAR (bug'doy OR don) AND NOT (butun or non) bo'lsa, u holda c = don

Amalda, qoidalar bundan kattaroq bo'ladi va ularni oddiy mantiqiy iboralardan ko'ra murakkabroq so'rov tillari, jumladan, raqamli ballardan foydalangan holda ifodalash mumkin. Ehtiyotkorlik bilan ishlov berish (ya'ni, odamlar tomonidan ishlab chiqish ma'lumotlari qoidalarini

sozlash orqali), bunday qoidalarning aniqligi juda yuqori bo'lishi mumkin. Jacobs va Rau (1990) Reuters axborot agentligi hujjatlarida 92% aniqlik va 88,5% esdalik bilan egallab olish haqidagi maqolalarni aniqlaganliklari va Hayes va Weinstein (1990) 675 toifadagi 94% esdalik va 84% aniqlik haqida xabar berishdi. Shunga qaramay, bunday yaxshi sozlangan qoidalarni yaratish uchun ish hajmi juda katta. O'rtacha hisob-kitob har bir sinf uchun 2 kunni tashkil etadi va qo'shimcha vaqt qoidalarni saqlashga ketishi kerak, chunki darslardagi hujjatlarning mazmuni vaqt o'tishi bilan o'zgaradi.

Agar sizda juda kam ma'lumotlar bo'lsa va siz nazorat ostidagi klassifikatorni o'rgatmoqchi bo'lsangiz, u holda mashinali o'qitish nazariyasi 3.6-bo'limda muhokama qilganimizdek, yuqori moyillikka ega bo'lgan klassifikatorga yopishib olishingiz kerakligini aytadi.

Masalan, u yerda ***Naive Bayes*** bunday sharoitlarda yaxshi bajaradigan nazariy va empirik natijalardir (Ng va Jordan 2001, Forman va Koen 2004), garchi bu ta'sir matnli ma'lumotlarga nisbatan tartibga solingan modellar bilan amalda mutlaqo kuzatilmaydi (Klein va Manning 2002).

Qanday bo'lmasin, eng yaqin qo'shni modeli kabi juda past yonbosish modeli teskari ko'rsatma bo'lishi mumkin. Nima bo'lishidan qat'i nazar, modelning sifati cheklangan ta'lim ma'lumotlariga salbiy ta'sir qiladi. Bu yerda nazariy jihatdan qiziqarli javob yarim nazorat ostida o'qitish usullarini qo'llashga harakat qilishdir.

Bunga yuklash yoki EM algoritmi kabi usullar kiradi, ularni 5.5-bo'limda kiritamiz.

Ushbu usullarda tizim ba'zi etiketli hujjatlarni va o'rganishga harakat qilishi mumkin bo'lgan yorliqsiz hujjatlarning yana katta zaxirasini oladi. ***Naive Bayes***ning katta afzalliklaridan biri shundaki, uni to'g'ridan-to'g'ri yarim nazorat ostida o'rganish algoritmi sifatida kengaytirish mumkin, ammo SVMlar uchun transduktiv SVMlar nomi ostida yarim nazorat ostida o'quv ishi ham mavjud. Ko'rsatkichlar uchun havolalarga qarang.

Ko'pincha, amaliy javob, iloji boricha tezroq etiketli ma'lumotlarni qanday qilib olishni ishlab chiqishdir.

Buni amalga oshirishning eng yaxshi usuli - odamlar o'zlarining tabiiy vazifalari sifatida siz uchun ma'lumotlarni belgilashga tayyor bo'ladigan jarayonga o'zingizni kiritishdir. Misol uchun, ko'p hollarda odamlar elektron pochtani o'z maqsadlari uchun saralaydi yoki yo'naltiradi va bu harakatlar sinflar haqida ma'lumot beradi.

Tasniflagichlarni o'qitish vazifasi uchun aniq inson yorlig'ini olishning muqobil variantini tashkil qilish ko'pincha qiyin va yorliqlash ko'pincha past sifatga ega, chunki teglar real vazifa kontekstiga kiritilmagan. Odamlar hujjatlarning barchasini yoki tasodifiy namunalarini belgilash o'rniga, faol o'rganish bo'yicha ham katta tadqiqotlar olib bordi, bu yerda inson qaysi hujjatlarni belgilashi kerakligini hal qiladigan tizim qurilgan. Odatda bular klassifikator to'g'ri tasnifga noaniq bo'lganlardir. Bu izohlash xarajatlarini 2–4 marta kamaytirishda samarali bo'lishi mumkin, ammo muammo bor - bir turdagi klassifikatorni o'rgatish uchun yorliqlanadigan yaxshi hujjatlar ko'pincha boshqa turdagi tasniflagichni o'rgatish uchun yaxshi hujjatlar emas.

Agar yorliqli ma'lumotlarning maqbul miqdori mavjud bo'lsa, unda siz matn tasnifi haqida taqdim etgan hamma narsadan foydalanish uchun mukammal holatdasiz. Masalan, siz SVM dan foydalanishni xohlashingiz mumkin. Biroq, agar siz SVM kabi chiziqli klassifikatorni ishlatayotgan bo'lsangiz, ehtimol siz mashinali o'qitish klassifikatori ustidan mantiqiy qoidaga asoslangan klassifikatorni qoplaydigan dasturni loyihalashingiz kerak.

Foydalanuvchilar ko'pincha to'g'ri kelmaydigan narsalarni o'zgartirishni yaxshi ko'radilar va agar rahbariyat telefondan qo'ng'iroq qilsa va ma'lum bir hujjatning tasnifini hoziroq tuzatishni xohlasa, buni qanday qilish kerakligini aniqlashdan ko'ra, qoidani qo'lda yozish ancha osondir. Umumiy tasnif aniqligini buzmasdan SVM og'irliklarini sozlash kerak.

Bu foydalanuvchi tomonidan talqin qilinadigan mantiqiy modellarni ishlab chiqaradigan qaror daraxtlari kabi mashinani o'qitish modellari katta mashhurlikni saqlab qolishining sabablaridan biridir.

Agar katta hajmdagi ma'lumotlar mavjud bo'lsa, unda klassifikatorni tanlash natijalaringizga unchalik ta'sir qilmaydi va eng yaxshi tanlov noaniq bo'lishi mumkin (Banko va Brill 2001).

Ta'limning miqyosi yoki hatto ish vaqti samaradorligiga asoslangan klassifikatorni tanlash yaxshi bo'lishi mumkin. Bu nuqtaga erishish uchun sizda katta hajmdagi ma'lumotlar bo'lishi kerak. Umumiy qoida shundan iboratki, o'quv ma'lumotlari hajmining har bir ikki baravar oshishi tasniflagichning ishlashida chiziqli o'sishni keltirib chiqaradi, lekin juda katta hajmdagi ma'lumotlar bilan yaxshilanish pastki chiziqli bo'ladi.

4.3.2. Tasniflagich ishlashini yaxshilash

Har qanday maxsus dastur uchun odatda domen yoki hujjatlar to'plamiga xos xususiyatlardan foydalanish orqali tasniflagich samaradorligini oshirish uchun katta imkoniyatlar mavjud. Ko'pincha hujjatlarda tasniflash uchun ayniqsa foydali bo'lgan zonalar mavjud. Ko'pincha tasniflashning optimal samaradorligi uchun maxsus davolashni talab qiladigan alohida sublug'atlar bo'ladi. Masalan, katta va qiyin toifali taksonomiyalar. Agar matnni tasniflash masalasi oz sonli yaxshi ajratilgan toifalardan iborat bo'lsa, unda ko'plab tasniflash algoritmlari yaxshi ishlaydi. Ammo ko'pgina haqiqiy tasniflash muammolari juda ko'p sonli ko'pincha juda o'xshash toifalardan iborat. O'quvchi veb-kataloglar (Yahoo! Katalogi yoki Ochiq katalog loyihasi), kutubxona tasnifi sxemalari (Dyui o'nlik yoki Kongress kutubxonasi) yoki yuridik yoki tibbiy ilovalarda ishlatiladigan tasniflash sxemalari kabi misollar haqida o'ylashi mumkin. Masalan, Yahoo! Katalog chuqur ierarxiyadagi 200000 dan ortiq toifalardan iborat. Bir-biriga yaqin bo'lgan sinflarning katta to'plamlarini aniq tasniflash tabiatan qiyin. Katta toifalar to'plamining aksariyati ierarxik tuzilishga ega va ierarxik tasniflash orqali ierarxiyadan foydalanishga urinish istiqbolli yondashuvdir. Biroq, hozirgi vaqtda ierarxiyaning sohalari bo'lgan sinflar bilan ishlashdan ko'ra, samaradorlik kam bo'lib qolmoqda. Ammo bu texnika katta ierarxiyalar bo'yicha klassifikatorlarni qurishning miqyoslanishini yaxshilash uchun juda foydali bo'lishi mumkin. Katta ierarxiyalar bo'yicha klassifikatorlarning miqyoslanishini yaxshilashning yana bir oddiy usuli - bu tajovuzkor xususiyatlarni tanlashdan foydalanish. 4.5-bo'limda ierarxik tasnif bo'yicha Ba'zi ishlarga havolalar beramiz. Misol sifatida 2.1-rasmdagi kichik ierarxiyadan foydalanib, soha sinflari sanoat kabi yuqori sinflardan farqli o'laroq parrandachilik va kofe kabidir.

Mashinali o'qitishning umumiy natijasi shundan iboratki, siz har doim bir nechta tasniflagichlarni birlashtirib, tasniflash aniqligini biroz oshirishingiz mumkin, faqat qilgan xatolar hech bo'lmaganda bir oz mustaqil bo'lsa. Hozirda ovoz berish va bir nechta tasniflagichlarni kuchaytirish kabi texnikalar bo'yicha katta adabiyot mavjud. Shunga qaramay, havolalarda ba'zi ko'rsatkichlar mavjud. Shunga qaramay, yetarlicha tasniflash aniqligiga erishish uchun oxir-oqibat gibrid

avtomatik/qo'lda yechim kerak bo'lishi mumkin. Bunday vaziyatlarda keng tarqalgan yondashuv birinchi navbatda klassifikatorni ishga tushirish va uning barcha yuqori ishonchli qarorlarini qabul qilish, ammo ishonch darajasi past bo'lgan qarorlarni qo'lda ko'rib chiqish uchun navbatga qo'yishdir. Bunday jarayon avtomatik ravishda mashinali o'qitish tasniflagichining kelajakdagi versiyalarida ishlatilishi mumkin bo'lgan yangi o'quv ma'lumotlarini ishlab chiqarishga olib keladi.

Biroq, shuni esda tutingki, bu natijada olingan o'quv ma'lumotlari hujjatlar maydonidan tasodifiy tanlanmasligi aniq. Matn uchun xususiyatlar - Ad hoc qidirishda ham, matn tasnifida ham sukut bo'yicha atamalardan xususiyatlar sifatida foydalanish kerak. Biroq, matnni tasniflash uchun ma'lum bir muammoga mos keladigan qo'shimcha funksiyalarni loyihalash orqali katta masofaga erishish mumkin. AQ so'rov tillaridan farqli o'laroq, bu xususiyatlar klassifikator uchun ichki bo'lganligi sababli, ushbu xususiyatlarni oxirgi foydalanuvchiga yetkazishda hech qanday muammo yo'q. Bu jarayon odatda xususiyat muhandisligi deb ataladi.

Hozirgi vaqtda texnik muhandislik mashinali o'qitish orqali amalga oshirilgan narsadan ko'ra insoniy hunarmandchilik bo'lib qolmoqda. Yaxshi xususiyat muhandisligi ko'pincha matn tasniflagichining ish faoliyatini sezilarli darajada yaxshilashi mumkin. Bu, ayniqsa, spam va porno filtrlash kabi matn tasnifining eng muhim ilovalarida foydalidir. Tasniflash muammolari ko'pincha qulay tarzda guruhlanishi mumkin bo'lgan va matnlarni tasniflash muammolarida o'xshash ovozga ega bo'lgan juda ko'p atamalarni o'z ichiga oladi. Odatda misollar yil eslatmasi yoki undov belgilari bo'lishi mumkin. Yoki ular ISBN yoki kimyoviy formulalar kabi ko'proq ixtisoslashgan tokenlar bo'lishi mumkin. Ko'pincha ularni to'g'ridan-to'g'ri klassifikatorda ishlatish, aytaylik, kimyoviy formula mavjudligini bilishdan tashqari, tasniflovchi kuchni ta'minlamasdan, so'z boyligini sezilarli darajada oshiradi. Bunday hollarda xususiyatlar soni va xususiyatlarning siyrakligini bunday elementlarni muntazam iboralar bilan moslashtirish va ularni ajratilgan belgilarga aylantirish orqali kamaytirish mumkin. Shunday qilib, samaradorlik va tasniflagich tezligi odatda yaxshilanadi. Ba'zan barcha raqamlar bitta xususiyatga aylantiriladi, lekin ko'pincha turli xil raqamlarni, masalan, to'rt xonali raqamlarni va o'nli kasrli haqiqiy raqamlarga nisbatan boshqa asosiy raqamlarni farqlash orqali ma'lum

qiymatga ega bo'lish mumkin. Shu kabi usullarni sanalar, ISBN raqamlari, sport o'yinlari ballari va boshqalar uchun qo'llash mumkin.

Boshqa yo'nalishga o'tadigan bo'lsak, ko'pincha so'zlarning qismlarini moslashtirish va ayniqsa kam tanlangan ko'p so'zli matnlarni moslashtirish orqali xususiyatlar sonini ko'paytirish foydali bo'ladi. So'zlarning qismlari ko'pincha ***k-gram*** belgilari bilan mos keladi. Bunday xususiyatlar, ayniqsa, klassifikator o'rnatilganda, noma'lum so'zlar uchun tasniflash maslahatlarini berishda yaxshi bo'lishi mumkin. Misol uchun, **-rase** bilan tugaydigan noma'lum so'z, garchi u mashg'ulot ma'lumotlarida ko'rilmagan bo'lsa ham, ferment bo'lishi mumkin. Yaxshi ko'p so'z namunalari ko'pincha o'ziga xos umumiy so'z juftliklarini izlash (ehtimol, 2.5.1-bo'limda funksiya tanlashda qo'llanilganiga o'xshash so'zlar o'rtasida o'zaro ma'lumot mezonidan foydalanish) va keyin baholangan xususiyatni tanlash usullari yordamida topiladi. Birikmaning tarkibiy qismlari tasniflash belgilari sifatida chalg'itadigan bo'lsa foydali bo'ladi. Masalan, etnik kalit so'z bo'lsa, shunday bo'ladi.

Oziq-ovqat va san'at toifalarining eng ko'p ko'rsatkichi bo'lib, tozalash kalit so'zi uy toifasini ko'rsatadi, ammo etnik tozalash o'rniga dunyo yangiliklari kategoriyasini ko'rsatadi. Ba'zi matn klassifikatorlari nomli obyektlarni tanib oluvchilarning xususiyatlaridan ham foydalanadi.

Matnni ajratish va kichik harflar bilan yozish kabi usullar matnni tasniflashda yordam beradimi? Har doimgidek, yakuniy sinov tegishli testlar to'plamida o'tkazilgan empirik baholashdir. Ammo shuni ta'kidlash kerakki, bunday usullarning tasniflash uchun foydali bo'lish imkoniyati cheklangan. AQ uchun siz tez-tez oksigenat va oksigenatsiya kabi so'z shakllarini yig'ishingiz kerak bo'ladi, chunki hujjatda ikkalasining ko'rinishi hujjat kislorod bilan bog'liq so'rovga mos kelishini yaxshi ko'rsatib beradi. Ko'p o'quv ma'lumotlarini hisobga olgan holda, ***stemming*** matn tasnifi uchun hech qanday ahamiyatga ega emas. Birgalikda joylashgan bir nechta shakllar o'xshash signalga ega bo'lsa, ularning barchasi uchun hisoblangan parametrlar bir xil og'irliklarga ega bo'ladi. ***Stemping*** kabi usullar faqat ma'lumotlarning siyrakligini qoplashda yordam beradi. Bu foydali rol bo'lishi mumkin (ushbu bo'limning boshida aytib o'tilganidek), lekin ko'pincha so'zning turli shakllari to'g'ri hujjat tasnifi haqida sezilarli darajada farq qilishi mumkin. Haddan tashqari aniqligi past ***stemming*** tasniflash samaradorligini osongina pasaytirishi mumkin. Matn tasnifidagi hujjat

zonalari birinchi qismning 6.1-bo'limida muhokama qilinganidek, hujjatlarda odatda mavzu va muallif kabi pochta xabari sarlavhalari yoki tadqiqot maqolasining sarlavhasi va kalit so'zlari kabi zonalar mavjud. Matn klassifikatorlari odatda o'qitish va tasniflash paytida ushbu zonalardan foydalanish orqali foyda olishlari mumkin. Hujjatlarni yuklash zonalari. Matnni tasniflash muammolarida siz turli xil hujjat zonalaridagi hissalarni farqlash orqali tez-tez samaradorlikni oshirishingiz mumkin. Ko'pincha, sarlavhali so'zlar ayniqsa samaralidir (Cohen and Singer 1999, p. 163). Qoida tariqasida, matnni tasniflash masalalarida sarlavhali so'zlarning og'irligini ikki baravar oshirish ko'pincha samaralidir. Bundan tashqari, aniq belgilangan zonalar bo'lmagan, ammo hujjat tuzilishi yoki mazmunidan olingan dalillar ularning muhimligini ko'rsatsa, matn qismlaridan yuqori og'irlikdagi so'zlardan ham qiymat olishingiz mumkin. Murata va boshqalar (2000) siz (newswire) hujjatning birinchi jumlasini yuksaltirishdan ham qiymat (ad hoc qidiruv kontekstida) olishingiz mumkinligini taklif qiladi.

Hujjat zonalari uchun alohida bo'shliqlar. Hujjat zonalari uchun ishlatilishi mumkin bo'lgan ikkita strategiya mavjud. Yuqorida ma'lum zonalarda paydo bo'ladigan so'zlarni ko'tardik. Bu shuni anglatadiki, ular bir xil xususiyatlardan foydalaniladi (ya'ni, parametrlar turli zonalar bo'ylab "bog'langan"), lekin alohida zonalarda atamalarning paydo bo'lishiga ko'proq e'tibor beramiz.

Muqobil strategiya - bu turli zonalarda uchraydigan so'zlar uchun mutlaqo alohida xususiyatlar to'plami va mos keladigan parametrlarga ega bo'lishdir. Bu printsipial jihatdan kuchliroqdir: so'z odatda sarlavhada Yaqin Sharq mavzusini, hujjat matnida esa Tovarlarni ko'rsatishi mumkin. Ammo, amalda, parametrlarni bog'lash odatda muvaffaqiyatli bo'ladi. Alohida xususiyatlar to'plamiga ega bo'lish ikki yoki undan ko'p baravar ko'p parametrlarga ega bo'lishni anglatadi, ularning ko'plari mashg'ulot ma'lumotlarida kamroq ko'rinadi va shuning uchun yomonroq baholar bilan, og'irlikni oshirish esa bunday yomon ta'sirga ega emas. Bundan tashqari, turli zonalarda paydo bo'lganda, so'zlar turli xil afzalliklarga ega bo'lishi juda kam uchraydi. Bu asosan ularni ovozlarining kuchiga moslashtirilishi kerak.

Shunga qaramay, pirovardida bu o'quv ma'lumotlarining tabiati va miqdoriga qarab shartli natijadir. Matnni umumlashtirishga ulanishlar mavjud. Birinchi qismning 8.7-bo'limida matnni umumlashtirish sohasi va bu sohadagi ko'pchilik ishlarning asosiy maqsadi jumlaning o'rnini

hisobga oladigan jumlalarning xususiyatlariga asoslanib, markaziy bo'lgan asl matn qismlarini ajratib olish va yig'ish kabi cheklangan maqsadni qanday qabul qilganini aytib o'tilgan edi. Ushbu ishning ko'p qismi matn tasnifi uchun alohida foydali bo'lishi mumkin bo'lgan zonalarni taklif qilish uchun ishlatilishi mumkin. Masalan, xususiyatlarni tanlash shaklini ko'rib chiqilgan va unda siz hujjatlarni faqat ma'lum zonalardagi so'zlar asosida tasniflaysiz. Matnni umumlashtirish bo'yicha tadqiqotlarga asoslanib, ular (i) faqat sarlavha, (ii) faqat birinchi xatboshi, (iii) faqat eng ko'p sarlavhali so'zlar yoki kalit so'zlarga ega bo'lgan paragraf, (iv) birinchi ikkita xatboshi yoki birinchi va oxirgi xatboshidan foydalanishni ko'rib chiqadi. Paragraf yoki (v) sarlavha so'zlari yoki kalit so'zlarning minimal soniga ega barcha jumlalar ham ko'rib chiqiladi. Umuman olganda, ushbu pozitsion xususiyatlarni tanlash usullari o'zaro ma'lumot kabi yaxshi natijalar berdi (2.5.1-bo'lim) va juda raqobatbardosh tasniflagichlarga olib keldi. Shuningdek, matnni umumlashtirish bo'yicha tadqiqotlardan ilhomlanib, sarlavhadagi so'zlar yoki hujjat mazmunida markaziy bo'lgan so'zlar bilan yuqori og'irlikdagi jumlalar bo'lib, tasniflash aniqligining deyarli 1% ga oshishiga olib keldi. Bu ishlaydi chunki bunday jumlalarning aksariyati hujjatning tashvishlariga ko'proq ahamiyat beradi.

Misol. Spam elektron pochta orqali o'tishga harakat qilish uchun ko'pincha turli xil yashirish usullaridan foydalanadi. Usullardan biri so'zga asoslangan matn tasniflagichlarini yo'q qilish uchun belgilarni to'ldirish yoki almashtirishdir. Masalan, spam xatida quyidagi kabi shartlarni ko'rasiz:

Rep1icaRolex bonmus Viiiaaaagra pi11z PHARlbdMACY [LEV]i[IT]l[RA] jinsiy ClAfLlS.

Ushbu strategiyani yengib o'tadigan xususiyatlarni qanday yaratishingiz mumkinligini muhokama qiling.

Misol. Elektron pochta spamlarini etkazib beruvchilar tomonidan tez-tez ishlatiladigan yana bir strategiya boshqa zararsiz manbadan (masalan, yangilik maqolasi) matn paragrafi bilan jo'natmoqchi bo'lgan xabarga (masalan, arzon aksiya sotib olish yoki boshqa narsa) amal qilishdir. Nima uchun bu strategiya samarali bo'lishi mumkin? Matn klassifikatori uni qanday hal qilish mumkin?

Misol. Qanday boshqa xususiyatlar elektron pochta spam tasniflagichida foydali bo'lib ko'rinadi?

Asosan atama va hujjatlarni baholash funksiyalarini qo'lda o'ylab topish o'rniga, o'qitish muammosining xususiyatlari sifatida tegishli signalning turli manbalarini (kosinus balli, sarlavha mosligi va boshqalar) ko'rishimiz mumkin. Har bir so'rovlar to'plami uchun tegishli va tegishli bo'lmagan hujjatlar misollari berilgan tasniflagich ushbu signallarning nisbiy og'irligini aniqlashi mumkin. Agar muammoni shunday sozlasak, hujjat va so'rov juftligi tegishli yoki ahamiyatsiz bo'lgan tegishlilik hukmi berilgan bo'lsa, bu muammoni ham matn tasniflash muammosi sifatida ko'rishimiz mumkin. Bunday tasniflash yondashuvini qo'llash har doim ham eng yaxshisi emas va 4.4.2-bo'limda muqobil variantni taqdim etamiz. Shunga qaramay, ko'rib chiqqan materialni hisobga olgan holda, boshlashning eng oddiy joyi bu muammoni tasniflash muammosi sifatida, hujjatlarni ikki toifali klassifikatorning tegishli qaroriga bo'lgan ishonchiga ko'ra tartiblashdir. Va bu harakat emas sof pedagogikadir. Aynan shu yondashuv ba'zan amalda qo'llaniladi.

4.4.1. Mashinada o'qitilgan ballarni hisoblashning oddiy misoli

Ushbu bo'limda birinchi qismning 6.1.2-bo'limi metodologiyasini ball funksiyasini mashina o'qitishga umumlashtiramiz. 6.1.2-bo'limda mantiqiy ko'rsatkichlarni birlashtirishimiz kerak bo'lgan holatni ko'rib chiqilgan. Bu yerda mashinada o'qitilgan dolzarblik tushunchasini yanada rivojlantirish uchun umumiyroq omillarni ko'rib chiqiladi.

4.3-jadval. **Mashinada o'qitilgan ballarni hisoblash uchun o'qitish misollari**

Example	DocID	Query	Cosine score	ω	Judgment
Φ_1	37	linux operating system	0.032	3	*relevant*
Φ_2	37	penguin logo	0.02	4	*nonrelevant*
Φ_3	238	operating system	0.043	2	*relevant*
Φ_4	238	runtime environment	0.004	2	*nonrelevant*
Φ_5	1741	kernel layer	0.022	3	*relevant*
Φ_6	2094	device driver	0.03	2	*relevant*
Φ_7	3191	device driver	0.027	5	*nonrelevant*
...	...	...	...	...	...

Xususan, hozir ko'rib chiqayotgan omillar **Boolean**dan tashqariga chiqadi.

6.1.2-bo'limda bo'lgani kabi hujjat zonalarida so'rovlar atamasi mavjudligi funksiyalari.

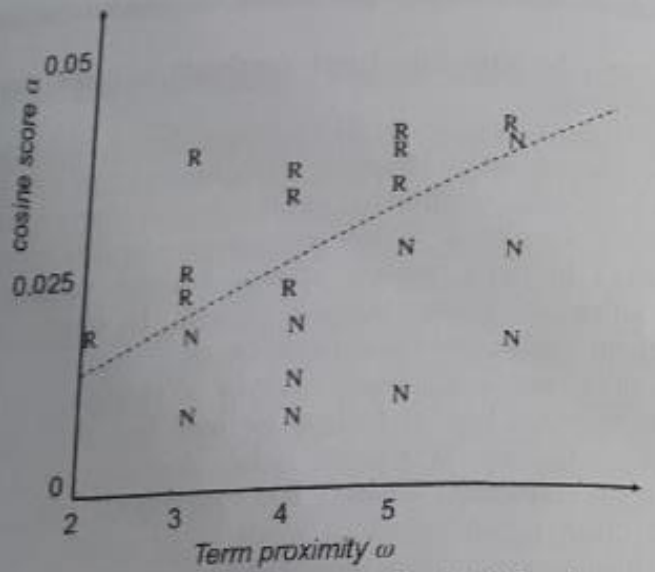
G'oyalarni baholash funksiyasi ikki omilning chiziqli birikmasidan iborat bo'lgan sharoitda ishlab chiqamiz: (1) so'rov va hujjat o'rtasidagi vektor fazosining kosinus o'xshashligi va (2) so'rov shartlari joylashgan oynaning minimal kengligi δ. So'rov atamasining yaqinligi ko'pincha hujjatning mavzu bo'yicha, ayniqsa uzunroq hujjatlar va Internetda hujjatning mavjudligini ko'rsatadi. Boshqa narsalar qatorida, bu miqdor ularga yashirin iboralarni amalga oshirish imkonini beradi.

Shunday qilib, ularda hujjatdagi so'rovlar to'plamidagi so'rovlar statistik ma'lumotlariga bog'liq bo'lgan bir omil bor, ikkinchisi esa yaqinlik vazniga bog'liq. G'oyalarni ishlab chiqishda faqat ikkita xususiyat ko'rib chiqiladi chunki ikki xususiyatli ekspozitsiya vizualizatsiya qilish uchun yetarlicha sodda bo'lib qoladi. Texnikani boshqa ko'plab xususiyatlar uchun umumlashtirish mumkin.

Ularga o'quv misollari to'plami taqdim etiladi, ularning har biri so'rov va hujjatdan iborat juftlik va shu so'rov bo'yicha tegishli yoki ahamiyatsiz bo'lgan ushbu hujjatga tegishli xulosa bilan birga. Har bir bunday misol uchun vektor fazosining kosinus o'xshashligini, shuningdek, oyna kengligi δ ni hisoblashimiz mumkin. Natijada 4.3-jadvalda ko'rsatilganidek o'xshash treninglar to'plamidir. Bu yerda ikkita xususiyat (kosinus balli a va oyna kengligi δ bilan ko'rsatilgan) haqiqiy baholi bashoratchilardir. Agar yana bir bor tegishli hukmni 1 va ahamiyatsiz 0 deb hisoblasak, 0 yoki 1 ga yaqin qiymat hosil qilish uchun xususiyatlar qiymatlarini birlashtirgan ball funksiyasini qidiramiz. Imkon qadar ularning ta'lim misollarimiz to'plamida foydalanamiz. Umumiylikni yo'qotmasdan, chiziqli klassifikator shakl xususiyatlarining chiziqli birikmasidan foydalanadi.

$$\Phi(d_i, d_j, q) = \psi(d_i, q) - \psi(d_j, q) \qquad (4.17)$$

O'quv ma'lumotlaridan o'rganish uchun a, b, c koeffitsientlari bilan berilgan. Buni xatolarni minimallashtirish muammosi sifatida shakllantirish mumkin bo'lsa-da, (4.17) tenglamaning geometriyasini tasavvur qilish juda foydali. 4.3-jadvaldagi misollarni kosinus ball a va oyna kengligi δ ga mos keladigan o'qlari bo'lgan ikki o'lchovli tekislikda tasvirlash mumkin. Bu 4.7-rasmda tasvirlangan.



4.7-rasm. **O‘qitish misollari to‘plami**

Har bir **R** tegishli yorliqli trening misolini bildiradi, har bir **N** esa tegishli emas deb belgilangan o‘quv misolidir.

Bu sozlamada (4.17) tenglamadagi **Score(a, ō)** funksiyasi “yuqorida osilgan” tekislikni ifodalaydi 4.7-rasm. Ideal holda, bu tekislik (4.7-rasmni o‘z ichiga olgan sahifaga perpendikulyar yo‘nalishda) **R** bilan belgilangan nuqtalar ustidagi **1** ga yaqin qiymatlarni va **N** bilan belgilangan nuqtalar ustidagi **0** ga yaqin qiymatlarni qabul qiladi. Chunki tekislik faqat 0 yoki ga yaqin qiymatlarni qabul qilishi dargumon. Mashg‘ulotning namunaviy nuqtalari ustidagi 1-bandida chegaradan foydalaniladi: tegishliligini aniqlamoqchi bo‘lgan har qanday so‘rov va hujjatni hisobga olsak, **th** qiymatini tanlaymiz va agar ***Ball(a,’n) > th*** bo‘lsa, hujjatni tegishli deb e’lon qiladilar, aks holda hujjatni ahamiyatsiz deb e’lon qilinadi. 3.8-rasmdan ma’lumki, ***Score(a, ’n) = th*** ni qanoatlantiradigan barcha nuqtalar chiziq hosil qiladi (4.7-rasmda kesik chiziq sifatida ko‘rsatilgan) va shu tariqa ularda tegishli bo‘lmagan misollarni ajratib turadigan chiziqli klassifikator mavjud. Geometrik jihatdan ajratuvchi chiziqni quyidagicha topishimiz mumkin. Balandligi 4.7-rasmni o‘z ichiga olgan sahifadan **th** yuqori bo‘lgan **Score(a, ’n)** tekisligidan o‘tuvchi chiziqni ko‘rib chiqaylik. Ushbu chiziqni pastga

tushiring (4.7-rasm). Bu 4.7-rasmdagi kesik chiziq bo‘ladi. Keyin, 4.7-rasmdagi kesilgan chiziqdan pastga tushgan har qanday keyingi so‘rov/hujjat juftligi ahamiyatsiz hisoblanadi. Kesilgan chiziqdan yuqorida, tegishli.

Shunday qilib, yuqoridagi kabi o‘quv misollari berilgan ikkilik tegishli/ahamiyatsiz hukm chiqarish muammosi 4.7-rasmdagi tegishli o‘quv misollarini aloqador bo‘lmaganlardan ajratib turadigan chiziqli chiziqni o‘rganishdan biriga aylanadi. Bu chiziq **a-ō** tekisligida bo‘lgani uchun **a** va **ō** ni o‘z ichiga olgan chiziqli tenglama sifatida ikkita parametrli (qiyalik va kesma) yozilishi mumkin. 2-15-boblarda ko‘rib chiqqan chiziqli tasniflash usullari ushbu qatorni tanlash usullarini taqdim etadi. Agar o‘quv namunalarining yetarlicha boy to‘plamini yarata olsak, 6.2.3-bo‘limdagi kabi qo‘lda ballni sozlash funksiyalaridan butunlay qochishimiz mumkin. Shubhasiz, muammo bu o‘qitish misollarining tegishli to‘plamini saqlab qolish qobiliyatidir, ularning ahamiyati ekspertlar tomonidan baholanishi kerak.

4.4.2. Mashinani o‘rganish bo‘yicha natijalarni tartiblash

Yuqoridagi g‘oyalarni ikkitadan ko‘p o‘zgaruvchilarning funksiyalari uchun osongina umumlashtirish mumkin. Hujjatning so‘rovga aloqadorligini ko‘rsatadigan ko‘plab boshqa ballar mavjud, jumladan statik sifat (**PageRank** uslubidagi o‘lchovlar, 10-bobda muhokama qilingan), hujjat yoshi, zona hissalari, hujjat uzunligi va boshqalar. Agar ushbu o‘lchovlar tegishli qarorlar bilan o‘quv hujjatlari to‘plami uchun hisoblanishi mumkin bo‘lsa, bunday chora-tadbirlarning istalgan soni mashinani o‘rganish tasniflagichini o‘qitish uchun ishlatilishi mumkin. Misol uchun, ikkilik ahamiyatga ega bo‘lgan mulohazalar bo‘yicha SVMni o‘rgatishimiz va hujjatlarning qaror chegarasidan imzolangan masofasi bilan monotonik bo‘lgan tegishlilik ehtimoli asosida hujjatlarga buyurtma berishimiz mumkin. Biroq, bu kabi AQ natijalari reytingiga yaqinlashish muammo haqida o‘ylashning to‘g‘ri usuli emas. Statistikalar odatda muammolarni birinchi navbatda tasniflash muammolariga (kategorik o‘zgaruvchi prognoz qilingan) va regressiya muammolariga (haqiqiy son bashorat qilingan) ajratadi. O‘rtada tartibli regressiyaning ixtisoslashgan sohasi joylashgan bo‘lib, unda reyting bashorat qilinadi. **Ad hoc** qidirish uchun mashinani o‘rganish eng to‘g‘ri

tartibli regressiya muammosi sifatida ko'rib chiqiladi, bu yerda maqsad xuddi shu turdagi o'quv ma'lumotlarini hisobga olgan holda so'rov uchun hujjatlar to'plamini tartiblashdir. Ushbu formula ba'zi bir qo'shimcha kuch beradi, chunki hujjatlarni xuddi shu so'rov uchun boshqa nomzodlik hujjatlariga nisbatan baholash mumkin, balki global miqyosdagi xaritada ko'rsatilishi shart emas, shu bilan birga muammoli maydonni zaiflashtiradi, chunki shunchaki reyting o'rniga talab qilinadi. Tegishlilikning mutlaq o'lchovidir. Reyting masalalari, ayniqsa, veb-qidiruvda dolzarbdir, bunda natijalar ro'yxatining eng yuqori qismidagi reyting juda muhim, hujjatning so'rovga aloqadorligi haqidagi qarorlar unchalik muhim emas. Bunday ish 4.2.2-bo'limda aytib o'tgan tuzilmaviy SVM tizimi yordamida olib borilishi mumkin va bunda bashorat qilinayotgan sinf so'rov natijalarining reytingidir, ammo bu yerda biroz soddaroq SVM reytingini taqdim etamiz.

Reytingli SVMni qurish quyidagicha davom etadi. Baholangan so'rovlar to'plamidan boshlaymiz. Har bir o'quv so'rovi **q** uchun so'rovga javoban qaytarilgan hujjatlar to'plami mavjud bo'lib, so'rovga tegishliligi uchun shaxs tomonidan to'liq buyurtma qilingan. 4.4.1-bo'limda ko'rib chiqilgan va boshqa ko'plab xususiyatlardan foydalangan holda har bir hujjat/so'rov juftligi uchun $\mathbf{p}_{sj} = \mathbf{p}_s(\mathbf{d}_{j,q})$ xususiyatlar vektorini quramiz. Ikkita $\mathbf{d}_i$ va $\mathbf{d}_j$ hujjatlari uchun xususiyatlar farqlari vektorini hosil qiladi:

$$\Phi(d_i, d_j, q) = \psi(d_i, q) - \psi(d_j, q)$$

Gipotezaga ko'ra, $\mathbf{d}_i$ va $\mathbf{d}_j$ dan biri muhimroq deb topilgan. Agar $\mathbf{d}_i$ $\mathbf{d}_j$ dan ko'ra ko'proq tegishli deb topilsa, $\mathbf{d}_i \prec \mathbf{d}_j$ bilan belgilanadi (natijalarni tartiblashda $\mathbf{d}_i$ $\mathbf{d}_j$ dan oldin bo'lishi kerak), u holda vektorga $\mathbf{P(d_i, d_j, q)}$ y $\mathbf{i_{ij}}$=+1 klassini beramiz aks holda -1. Keyin maqsad qaytib keladigan tasniflagichni yaratishdir.

$$\vec{w}^T \Phi(d_i, d_j, q) > 0 \quad if \quad d_i \prec d_j \tag{4.19}$$

Ushbu SVM o'rganish vazifasi ilgari ko'rgan boshqa misollarga o'xshash tarzda rasmiylashtirilgan:

$$\bullet \ \frac{1}{2}\vec{w}^T\vec{w} + C\sum_{i,j}\xi_{i,j} \ \text{is minimized} \tag{4.20}$$

$$\bullet \ \text{and for all } \{\Phi(d_i, d_j, q): d_i \prec d_j\}, \vec{w}^T\Phi(d_i, d_j, q) \geq 1 - \xi_{i,j}$$

Cheklov bayonotida $\mathbf{y_{ijq}}$ ni qoldirib qo'yishimiz mumkin, chunki faqat bir yo'nalishda tartiblangan hujjat juftlari uchun cheklovni ko'rib chiqishimiz kerak. Nimagaki, $\prec$ antisimmetrikdir. Keyinchalik bu cheklovlar, avvalgidek, hujjatlar juftlarini tartiblashi mumkin bo'lgan chiziqli tasniflagichni berish uchun hal qilinadi. Ushbu yondashuv standart ma'lumotlar to'plamlari bo'yicha AQ baholashda standart qo'lda tuzilgan tartiblash funksiyalaridan ustun bo'lgan tartiblash funksiyalarini yaratish uchun ishlatiladi. Bunday natijalarni taqdim etadigan hujjatlar uchun havolalarga qarang. Hozirgina ko'rib chiqqan ikkala usul ham ushbu sohadagi ko'plab ishlar kabi tegishlilik ko'rsatkichi bo'lgan hujjat xususiyatlarining chiziqli vaznidan foydalanadi. Shu sababli, an'anaviy AQ tortishishning ko'p qismi asosiy o'lchovlarning chiziqli bo'lmagan masshtabini o'z ichiga olishi (masalan, muddatli chastota yoki **idf**) og'irligini hisobga olish qiziq hisoblanadi. Hozirgi vaqtda mashinani o'rganish chiziqli kombinatsiyadagi (yoki boshqa shunga o'xshash cheklangan model sinflari) xususiyatlar uchun optimal og'irliklarni ishlab chiqarishda juda yaxshi, lekin asosiy o'lchovlarning yaxshi chiziqli bo'lmagan masshtablarini ishlab chiqishda yaxshi emas. Bu soha inson xususiyatlari muhandisligi sohasi bo'lib qolmoqda. Reyting funksiyalarini o'rganish g'oyasi bir necha yillardan buyon mavjud edi, ammo yaqinda bu usulni amaliy va qiziqarli qilish uchun mashinani o'rganish bo'yicha yetarli bilim, o'quv hujjatlari to'plami va hisoblash quvvati birlashdi. Shunday qilib, ma'lumotni qidirishda tartiblash bo'yicha mashinani o'rganish yondashuvlari haqida aniq bir narsa yozishga hali erta, ammo vaqt o'tishi bilan mashinada o'rganilgan reyting yondashuvlaridan foydalanish va ahamiyatini kutish uchun barcha asoslar mavjud. Malakali odamlar reyting funksiyalarini qo'lda belgilashda juda yaxshi ishni bajarishlari mumkin bo'lsa-da, qo'lda sozlash qiyin va har bir yangi hujjat to'plami va foydalanuvchilar toifasi uchun buni yana qilish kerak.

4.7-mashq. 4.3-jadvalning birinchi 7- qatorini **a-ō** tekisligida chizib, quyidagi rasmda shunday shakl hosil qiling. 4.7-rasm.

4.8-mashq. **Rs** ni **Ns** dan ajratuvchi **a-ō** tekislikdagi chiziq tenglamasini yozing.

4.9-mashq. Trening misolini keltiring (**a, ō** qiymatlari va ahamiyatlilik mulohazasidan iborat) o'quv to'plamiga qo'shilganda **a-ō** tekisligidagi chiziq yordamida **R** ni **N** dan ajratib bo'lmaydi.

Lester, Nicholas, Justin Zobel, and Hugh E. Williams.
2006.
Efficient online index maintenance for contiguous inverted lists.
IP&M 42 (4): 916-933.
DOI: *dx.doi.org/10.1016/j.ipm.2005.09.005*.
Levenshtein, Vladimir I.
1965.
Binary codes capable of correcting spurious insertions and deletions of ones.
Problems of Information Transmission 1: 8-17.
Lew, Michael S.
2001.
Principles of Visual Information Retrieval.
Springer.
Lewis, David D.
1995.
Evaluating and optimizing autonomous text classification systems.
In *Proc. SIGIR*. ACM Press.
Lewis, David D.
1998.
Naive (Bayes) at forty: The independence assumption in information retrieval.
In *Proc. ECML*, pp. 4-15. Springer.
Lewis, David D., and Karen Spärck Jones.
1996.
Natural language processing for information retrieval.
CACM 39 (1): 92-101.
DOI: *doi.acm.org/10.1145/234173.234210*.
Büttcher, Stefan, and Charles L. A. Clarke.
2005a.
Indexing time vs. query time: Trade-offs in dynamic information retrieval systems.
In *Proc. CIKM*, pp. 317-318. ACM Press.
DOI: *doi.acm.org/10.1145/1099554.1099645*.
Büttcher, Stefan, and Charles L. A. Clarke.

2005b.
A security model for full-text file system search in multi-user environments.
In *Proc. FAST*.
URL: *www.usenix.org/events/fast05/tech/buettcher.html*.
Büttcher, Stefan, and Charles L. A. Clarke.
2006.
A document-centric approach to static index pruning in text retrieval systems.
In *Proc. CIKM*, pp. 182-189.
DOI: *doi.acm.org/10.1145/1183614.1183644*.

4- bob bo'yicha nazariy va amaliy test savollari

1. SPIMI-INVERT – nima?
   A) Butun to'plam qayta ishlanmaguncha qayta-qayta chaqiriladigan tokenlash oqimi
   B) Blokni o'qish yoki yozish bajarilish paytida asosiy xotiraning qismi
   C) Indeks tuzish uchun atamalarni samaraliroq qilish usuli
   D) To'g'ri javob yo'q
2. Butun to'plam qayta ishlanmaguncha qayta-qayta chaqiriladigan tokenlash oqimi nima deb ataladi?
   A) SPIMI-INVERT
   B) termID
   C) taqsimlangan indekslash
   D) To'g'ri javob yo'q
3. Veb-qidiruv tizimlari indeks yaratish uchun qanday algoritmlaridan foydalanadi?
   A) taqsimlangan indekslash
   B) SPIMI-INVERT
   C) termID
   D) To'g'ri javob yo'q
4. Taqsimlangan indekslash bu nima?
   A) Veb-qidiruv tizimlari indeks yaratish algoritmi
   B) Butun to'plam qayta ishlanmaguncha qayta-qayta chaqiriladigan tokenlash oqimi

C) Blokni o'qish yoki yozish bajarilish paytida asosiy xotiraning qismi

D) To'g'ri javob yo'q

5. MapReduce bu – nima?

A) Bu shunday klasterki, klasterning maqsadi arzon tovar mashinalari yoki tugunlarida katta hisoblash muammolarini hal qilish uchun standart qismlardan (protsessor, xotira, disk) farqli ravishda qurilgan maxsus uskunaga ega superkompyuter yaratish

B) Individual ishchi tugunlariga vazifalarni qayta tayinlash va tayinlash jarayonini boshqarish

C) Butun to'plam qayta ishlanmaguncha qayta-qayta chaqiriladigan tokenlash oqimi

D) To'g'ri javob yo'q

6. Klasterning maqsadi arzon tovar mashinalari yoki tugunlarida katta hisoblash muammolarini hal qilish uchun standart qismlardan (protsessor, xotira, disk) farqli ravishda qurilgan maxsus uskunaga ega superkompyuter yaratish. Bu klasterning nomi nima?

A) MapReduce
B) Asosiy tugun
C) termID
D) To'g'ri javob yo'q

7. Asosiy tugunning vazifasi nima?

A) Individual ishchi tugunlariga vazifalarni qayta tayinlash va tayinlash jarayonini boshqarish

B) Bu shunday klasterki, klasterning maqsadi arzon tovar mashinalari yoki tugunlarida katta hisoblash muammolarini hal qilish uchun standart qismlardan (protsessor, xotira, disk) farqli ravishda qurilgan maxsus uskunaga ega superkompyuter yaratish

C) Veb-qidiruv tizimlari indeks yaratish algoritmi

D) Butun to'plam qayta ishlanmaguncha qayta-qayta chaqiriladigan tokenlash oqimi

8. Individual ishchi tugunlariga vazifalarni qayta tayinlash va tayinlash jarayonini boshqarish to'g'ri berilgan javobni toping.

A) Asosiy tugun
B) MapReduce
C) termID
D) To'g'ri javob yo'q

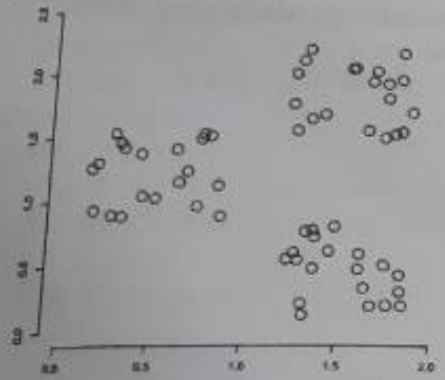
9. Sql serverda indeksning maqsadi nima?

A) Barcha qayd etilganlar
B) Yozuvga indeks berish
C) Tezkor qidiruvlarni amalga oshirish uchun
D) So'rovlar samaradorligini oshirish uchun

10. Qanday qilib klasterli bo'lmagan indeks ma'lumotlarga ishora qiladi?

A) U asosiy qiymatlarni o'z ichiga olgan ma'lumotlar qatorlarini ko'rsatish uchun ishlatiladi
B) U hech qachon biror narsaga ishora qilmaydi
C) Ma'lumotlar qatoriga ishora qiladi
D) To'g'ri javob yo'q

Klasterlash algoritmlari hujjatlar to'plamini kichik to'plamlarga yoki klasterlarga guruhlaydi. Algoritmlarning maqsadi ichki izchil, ammo bir-biridan aniq farq qiladigan klasterlarni yaratishdir. Boshqacha qilib aytganda, klaster ichidagi hujjatlar imkon qadar o'xshash bo'lishi kerak va bitta klasterdagi hujjatlar boshqa klasterlardagi hujjatlardan imkon qadar o'xshamasligi kerak.



5.1-rasm. **Aniq klaster tuzilmasi bo'lgan ma'lumotlar to'plamiga misol**

Klaster - bu nazoratsiz ta'limning eng keng tarqalgan shaklidir. Nazorat yo'q degani, darslarga hujjatlarni topshirgan mutaxassis yo'qligini anglatadi. Klasterlashda ma'lumotlarning taqsimlanishi va tuzilishi klaster a'zoligini belgilaydi. Oddiy misol - 5.1-rasm. Ko'rinib turibdiki, uchta alohida nuqta klasteri mavjud. Ushbu bob va 6-bobda bunday klasterlarni nazoratsiz tarzda topadigan algoritmlar keltirilgan. Klasterlash va tasniflash o'rtasidagi farq dastlab unchalik katta bo'lmasligi mumkin. Chunki ikkala holatda ham ularda hujjatlar to'plamining guruhlarga bo'linishi mavjud. Ammo ikkita muammo tubdan farq qiladi. Tasniflash - bu nazorat ostida o'rganishning bir shakli (2-bob). Ularning maqsadi inson nazoratchisi ma'lumotlarga qo'yadigan toifali farqni takrorlashdir. Klasterlash eng muhim misol bo'lgan nazoratsiz ta'limda ularni boshqaradigan o'qituvchi yo'q. Klasterlash algoritmiga asosiy kirish masofa o'lchovidir. 5.1-rasmda masofa o'lchovi 2D tekislikdagi masofadir. Ushbu o'lchov rasmda uchta turli klasterni taklif qiladi. Hujjatlarni klasterlashda masofa o'lchovi ko'pincha ***Evklid masofasi*** hisoblanadi. Turli xil masofa o'lchovlari turli xil guruhlarni keltirib chiqaradi. Shunday qilib, masofa o'lchovi klasterlash natijasiga ta'sir qilishi mumkin bo'lgan muhim vositadir. Yassi klasterlash klasterlarni bir-biriga bog'laydigan aniq tuzilishga ega bo'lmagan tekis klasterlar to'plamini yaratadi. Ierarxik klasterlash klasterlar ierarxiyasini yaratadi va 6-bobda ko'rib chiqiladi. 6-bob shuningdek, klasterlarni avtomatik ravishda markalashning qiyin muammosini hal qiladi. Ikkinchi muhim farqni qattiq va yumshoq klasterlash algoritmlari o'rtasida ajratish mumkin. Qattiq klasterlash qiyin topshiriqni hisoblab chiqadi - har bir hujjat aynan bitta klaster a'zosidir. Yumshoq klaster algoritmlarini tayinlash yumshoq - hujjatning topshirilishi barcha klasterlar bo'yicha taqsimotdir. Yumshoq topshiriqda hujjat bir nechta klasterlarda kasr a'zoligiga ega. Yashirin semantik indekslash, o'lchovni qisqartirish shakli yumshoq klasterlash algoritmidir (7-bob). Ushbu bob bir qator ilovalarni kiritish orqali ma'lumot qidirishda klasterlashdan foydalanishni qo'llab quvvatlaydi (5.1-bo'lim), ular klasterlashda hal qilmoqchi bo'lgan muammoni belgilaydi (6.2-bo'lim) va klaster sifatini baholash bo'yicha chora-tadbirlarni muhokama qiladi (5.3-bo'lim). Keyin ikkita yassi klasterlash algoritmi, ***K-means*** (5.4-bo'lim), qattiq klasterlash algoritmi va ***Kutish-maksimizatsiya*** (yoki EM) algoritmi (5.5-bo'lim), yumshoq klasterlash algoritmi. ***K-means*** oddiyligi va samaradorligi tufayli ehtimol, eng keng tarqalgan yassi klasterlash algoritmidir. EM algoritmi K-vositalarni umumlashtirish bo'lib, hujjatlarni ko'rsatish va tarqatishning ko'p turlariga qo'llanilishi mumkin.

5.1. Axborot qidirishda klasterlash

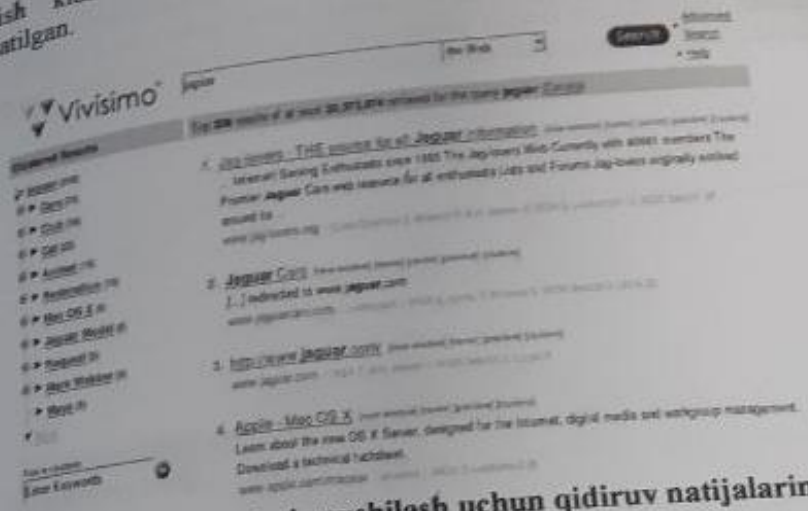
Klaster gipotezasi ma'lumotni qidirishda klasterlashdan foydalanganda qiladigan asosiy taxminni bildiradi. Klaster gipotezasi. Xuddi shu klasterdagi hujjatlar axborotga bo'lgan ehtiyojga nisbatan xuddi shunday yo'l tutadi. Gipotezada aytilishicha, agar qidiruv so'roviga tegishli bo'lgan klasterdan hujjat mavjud bo'lsa unda xuddi shu klasterdagi boshqa hujjatlar ham tegishli bo'lishi mumkin. Buning sababi shundaki, klasterlash ko'plab atamalarni birlashtiradigan hujjatlarni birlashtiradi. Klaster gipotezasi asosan 3-bobdagi qo'shnilik gipotezasidir. Ikkala holatda ham shunga o'xshash hujjatlar tegishlilik nuqtai nazaridan o'xshashligini taxmin qiladi.

5.1-jadval. **Axborot qidirishda klasterlashning ba'zi ilovalari**

Application	What is clustered?	Benefit	Example
Search result clustering	search results	more effective information presentation to user	Figure 16.2
Scatter-Gather	(subsets of) collection	alternative user interface: "search without typing"	Figure 16.3
Collection clustering	collection	effective information presentation for exploratory browsing	McKeown et al. (2002), http://news.google.com
Language modeling	collection	increased precision and/or recall	Liu and Croft (2004)
Cluster-based retrieval	collection	higher efficiency: faster search	Salton (1971a)

5.1-jadvalda ma'lumot qidirishda klasterlashning asosiy qo'llanilishi ko'rsatilgan. Ular klasterlashtirgan hujjatlar to'plamida - qidiruv natijalari, to'plam yoki to'plamning kichik to'plamlarida va ularni yaxshilashga harakat qilayotgan ma'lumot qidirish tizimi jihatida - foydalanuvchi tajribasi, foydalanuvchi interfeysi, qidiruv tizimining samaradorligi yoki samaradorligi bilan farqlanadi. Ammo ularning barchasi klaster gipotezasi tomonidan bayon qilingan asosiy taxminga asoslanadi. 5.1-jadvalda keltirilgan birinchi dastur qidiruv natijalari klasteridir, bunda qidiruv natijalari deganda so'rovga javoban qaytarilgan hujjatlar tushuniladi. Ma'lumotni qidirishda qidiruv natijalarining standart taqdimoti oddiy ro'yxatdir. Foydalanuvchilar o'zlari qidirayotgan ma'lumotni topmaguncha ro'yxatni yuqoridan pastga qarab skanerlashadi. Buning o'rniga, qidiruv natijalarini klasterlash qidiruv natijalarini o'xshash hujjatlar bilan birga paydo bo'lishi uchun birlashtiradi. Ko'p individual hujjatlarga qaraganda bir nechta izchil guruhlarni skanerlash ko'pincha osonroq. Bu ayniqsa, qidiruv so'zi turli so'z ma'nolariga ega bo'lsa foydalidir. 5.2-rasmdagi misol yaguar. Internetda uchta tez-tez uchraydigan hislar mashina, hayvon va Apple operatsion tizimini anglatadi. ***Vivisimo*** qidiruv tizimi (***http://vivisimo.com***) tomonidan qaytarilgan Klasterli natijalar paneli oddiy hujjatlar ro'yxatidan ko'ra qidiruv natijalarida nima borligini tushunish uchun samaraliroq foydalanuvchi interfeysi bo'lishi mumkin. Yaxshiroq foydalanuvchi interfeysi 5.1-jadvaldagi ikkinchi dastur bo'lgan ***Scatter-Gather*** dasturining maqsadidir. ***Scatter-Gather*** foydalanuvchi tanlashi yoki to'plashi mumkin bo'lgan hujjatlar guruhlarini olish uchun butun to'plamni to'playdi. Tanlangan guruhlar

birlashtiriladi va natijada olingan to'plam yana klasterlanadi. Bu jarayon qiziqish klasteri topilmaguncha takrorlanadi. Misol 5.3-rasmda ko'rsatilgan.

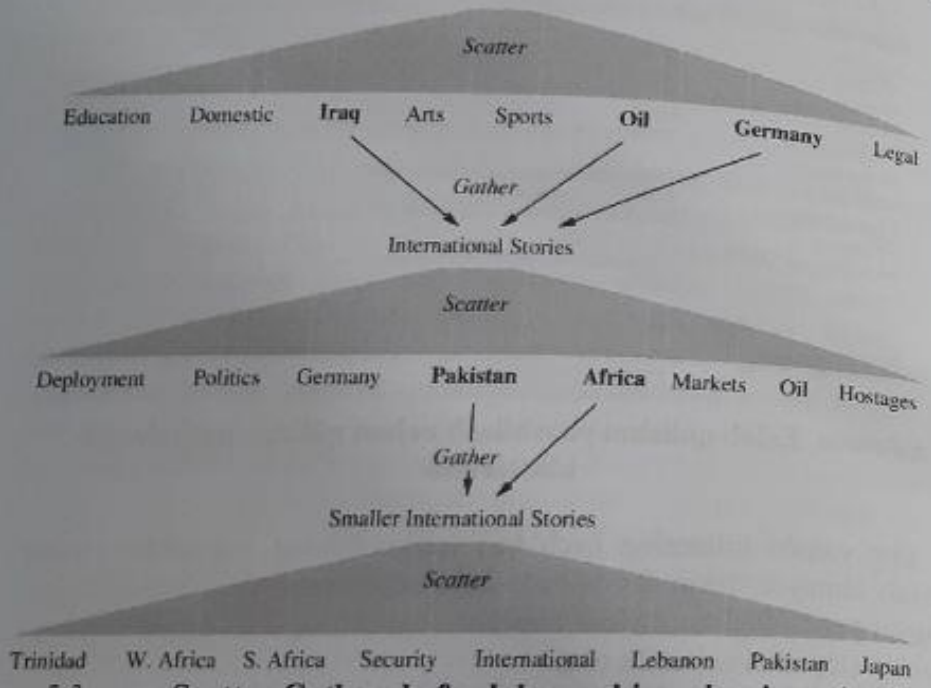


5.2-rasm. **Eslab qolishni yaxshilash uchun qidiruv natijalarini klasterlash**

Eng yaxshi hitlarning hech biri hayvonlarning yaguar tuyg'usini qamrab olmaydi, lekin foydalanuvchilar chap tomondagi klasterlangan natijalar panelidagi mushuklar klasterini bosish orqali osongina kirishlari mumkin (tepadan uchinchi o'q).

5.3-rasmdagi kabi avtomatik tarzda yaratilgan klasterlar **http://dmoz.org** saytidagi ochiq katalog kabi qo'lda tuzilgan ierarxik daraxt kabi tartibli tarzda tashkil etilmagan. Bundan tashqari, klasterlar uchun tavsiflovchi teglarni avtomatik ravishda topish qiyin muammodir (6.7-bo'lim). Ammo klasterga asoslangan navigatsiya kalit so'zlarni qidirishga qiziqarli muqobildir, standart ma'lumotni qidirish paradigmasi. Bu, ayniqsa, foydalanuvchilar qidiruvdan ko'ra ko'rishni afzal ko'rgan stsenariylarda to'g'ri keladi chunki ular qaysi qidiruv so'zlarini ishlatishni bilmaydi. ***Scatter-Gather***-da foydalanuvchi vositachiligidagi iterativ klasterlashga muqobil sifatida foydalanuvchilarning o'zaro ta'siriga ta'sir qilmaydigan to'plamning statik ***ierarxik klasterini*** ham hisoblash mumkin (5.1-jadvaldagi to'plam klasteri). *Google News* va uning

prekursori ***Columbia News Blaster*** tizimi bu yondashuvga misol bo'la oladi. Yangiliklar bo'lsa, foydalanuvchilarning so'nggi yangiliklarga kirishiga ishonch hosil qilish uchun tez-tez klasterlashni qayta hisoblash kerak. Klasterlash yangiliklar to'plamiga kirish uchun juda mos keladi chunki yangiliklarni o'qish aslida qidiruv emas balki so'nggi voqealar haqidagi hikoyalar to'plamini tanlash jarayonidir.



5.3-rasm. **Scatter-Gather-da foydalanuvchi sessiyasiga misol**

New York Times yangiliklar to'plami sakkizta klasterga (yuqori qatorga) to'plangan ("tarqalgan"). Foydalanuvchi ulardan uchtasini qo'lda kichikroq Xalqaro hikoyalar to'plamiga to'playdi va boshqa tarqatish operatsiyasini bajaradi. Bu jarayon tegishli hujjatlarga ega kichik klaster topilmaguncha takrorlanadi (masalan, Trinidad).

Klasterlashning to'rtinchi ilovasi to'liq to'plamni klasterlash asosida qidiruv natijalarini yaxshilash uchun klaster gipotezasini to'g'ridan-to'g'ri ishlatadi. Ular so'rovga mos keladigan dastlabki hujjatlar to'plamini aniqlash uchun standart teskari indeksdan foydalaniladi, lekin so'rovga o'xshashligi past bo'lsa ham, bir xil klasterlardan boshqa

hujjatlar qo'shiladi. Misol uchun, agar so'rov avtomobil bo'lsa va avtomobil hujjatlari klasteridan bir nechta avtomobil hujjatlari olingan bo'lsa, ushbu klasterdan avtomobildan boshqa atamalar ishlatadigan hujjatlarni qo'shish mumkin (avtomobil, transport vositasi va boshqalar). Bu eslab qolishni oshirishi mumkin chunki o'zaro o'xshashligi yuqori bo'lgan hujjatlar guruhi odatda bir butun sifatida tegishli. Yaqinda bu g'oya tilni modellashtirish uchun ishlatilgan. Tenglama (11.10) AQ ga til modellashtirish yondashuvida kam ma'lumotlar muammolarini oldini olish uchun **d** hujjat modeli yig'ish modeli bilan interpolyatsiya qilinishi mumkinligini ko'rsatdi. Ammo to'plamda **d** ga xos bo'lmagan atamalar bilan ko'plab hujjatlar mavjud. To'plam modelini **d** klasteridan olingan model bilan almashtirib, **d** da atamalarning paydo bo'lish ehtimolini aniqroq baholash mumkin. Klasterlash ham qidiruvni tezlashtirishi mumkin. Vektor fazo modelida qidirish so'rovga eng yaqin qo'shnilarni topishga teng. Inverted indeks standart AQ sozlamasi uchun eng yaqin qo'shnini tezkor qidirishni qo'llab-quvvatlaydi. Biroq, ba'zida teskari indeksdan samarali foydalana olmaslik mumkin masalan, yashirin semantik indekslashda (7-bob). Bunday hollarda so'rovning har bir hujjatga o'xshashligini hisoblanilishi mumkin, ammo bu sekin.

Klaster gipotezasi muqobil variantni taklif qiladi. Bu so'rovga eng yaqin bo'lgan klasterlarni toping va faqat shu klasterlardagi hujjatlarni ko'rib chiqishni anglatadi. Ushbu ancha kichikroq to'plamda o'xshashliklarni to'liq hisoblash va hujjatlarni odatiy tarzda tartiblash imkoniyatini yaratadi. Hujjatlarga qaraganda klasterlar kamroq bo'lganligi sababli, eng yaqin klasterni tezda topish va so'rovga mos keladigan hujjatlar bir-biriga o'xshash bo'lgani uchun ular bir xil klasterlarda bo'lishga moyildir. Ushbu algoritm noto'g'ri bo'lsa-da, qidiruv sifatining kutilayotgan pasayishi unchalik katta emas. Bu asosan 6.1.6-bo'limda yoritilgan klasterlashning qo'llanilishidir.

Misol. Ikkita hujjatni o'xshash deb belgilang, agar ularda ***Klinton*** yoki ***Sarkozi*** kabi kamida ikkita xos nom bo'lsa. Axborotga bo'lgan ehtiyoj va ikkita hujjatga misol keltiring, ular uchun klaster gipotezasi bu o'xshashlik tushunchasiga mos kelmaydi.

Misol. Klasterga asoslangan qidirishning noaniqligi namoyon bo'ladigan ikkita klaster bilan oddiy bir o'lchovli misol (ya'ni chiziqdagi nuqtalar) tuzing. Sizning misolingizda so'rovga yaqin klasterlarni olish eng yaqin qo'shni qidirishdan ko'ra yomonroq bo'lishi kerak.

5.2. Muammo bayoni

Qattiq yassi klasterlashda maqsadni quyidagicha belgilash mumkin. Berilgan (i) hujjatlar to'plami $D = \{d_1, \ldots d_N\}$, (ii) a kerakli miqdordagi klasterlar **K** va (iii) klasterlash sifatini baholovchi maqsad funksiyasi, topshiriqni hisoblash mumkin. **g**: $\gamma : D \to \{1, \ldots K\}$ maqsad funksiyasini minimallashtiradigan (yoki boshqa hollarda maksimal darajaga keltiradigan) birlik. Aksariyat hollarda **g** ning subyektiv bo'lishini ya'ni **K** klasterlarining hech biri bo'sh bo'lmasligini ham talab qiladi. Maqsad funksiyasi ko'pincha hujjatlar orasidagi o'xshashlik yoki masofa nuqtai nazaridan aniqlanadi. Quyida **K**-vositalarini klasterlashdan maqsad hujjatlar va ularning markazlari orasidagi o'rtacha masofani minimallashtirish yoki shunga o'xshash hujjatlar va ularning markazlari o'rtasidagi o'xshashlikni maksimal darajada oshirish ekanligini ko'rish mumkin. 3-bobda o'xshashlik o'lchovlari va masofa ko'rsatkichlari muhokamasi ushbu bobga ham tegishli. 3-bobda bo'lgani kabi, hujjatlar orasidagi bog'liqlik haqida gapirish uchun ham o'xshashlik, ham masofadan foydalaniladi.

Hujjatlar uchun xohlagan o'xshashlik turi odatda vektor fazo modelida mavzu o'xshashligi yoki bir xil o'lchamdagi yuqori qiymatlardir. Misol uchun, Xitoy haqidagi hujjatlar Xitoy, Pekin va Mao kabi o'lchamlar bo'yicha yuqori qiymatlarga ega, Buyuk Britaniya haqidagi hujjatlar esa London, Britaniya va Qirolicha uchun yuqori qiymatlarga ega. Mavzu o'xshashligini vektor fazoda kosinus o'xshashligi yoki Evklid masofasi bilan taxmin qilinadi (1- qismning 6-bob). Agar mavzudan boshqa turdagi o'xshashlikni masalan, tilning o'xshashligini qo'lga kiritmoqchi bo'lsak unda boshqa taqdimot mos bo'lishi mumkin. Mavzu o'xshashligini hisoblashda to'xtash so'zlariga e'tibor bermaslik mumkin ammo ular ingliz tili (tez-tez va tez-tez uchraydigan) va fransuz hujjatlari (kamdan-kam uchraydigan va tez-tez uchraydigan) klasterlarini ajratish uchun muhim signaldir. Terminologiya bo'yicha eslatma. Qattiq klasterlashning muqobil ta'rifi shundan iboratki, hujjat bir nechta klasterning to'liq a'zosi bo'lishi mumkin. Bo'limli klasterlash har doim har bir hujjat aynan bitta klasterga tegishli bo'lgan klasterga ishora qiladi. (Ammo qisman ierarxik klasterlashda (6-bob) klasterning barcha a'zolari, albatta, uning asosiy a'zolaridir.) Ko'p a'zolikka ruxsat beruvchi qattiq klasterlashning ta'rifiga kelsak, yumshoq

klasterlash va qattiq klasterlash o'rtasidagi farq shundaki, a'zolik qiymatlari qattiq klasterlash 0 yoki 1 ga teng, yumshoq klasterlashda esa ular har qanday salbiy bo'lmagan qiymatni olishi mumkin.

Ba'zi tadqiqotchilar har bir hujjatni klasterga belgilaydigan to'liq klasterlar va to'liq bo'lmagan klasterlar o'rtasida farqlaydi, bunda ba'zi hujjatlar hech qanday klasterga biriktirilmaydi. Har bir hujjat hech qanday klaster yoki bitta klaster a'zosi bo'lgan to'liq bo'lmagan klasterlar eksklyuziv deyiladi. Ular ushbu kitobda klasterlashni to'liq deb belgilanadi.

5.2.1. Kardinallik – klasterlar soni

Klasterlashda qiyin masala klasterlar sonini yoki klasterning kardinalligini aniqlashdir, buni **K** deb belgilanadi. Ko'pincha **K** tajriba yoki domenga asoslangan yaxshi taxmindan boshqa narsa emas. Ammo ***K-vositalari*** uchun **K** ni tanlashning evristik usulini va **K** ni tanlashni maqsad funksiyasiga kiritishga urinishni ham kiritish mumkin. Ba'zan dastur **K** diapazoniga cheklovlar qo'yadi. Masalan, ***Scatter-Gather***. 5.3-rasmdagi interfeys 1990-yillarning boshlarida kompyuter monitorlarining o'lchamlari tufayli har bir qatlam uchun taxminan K = 10 dan ortiq klasterni ko'rsata olmadi.

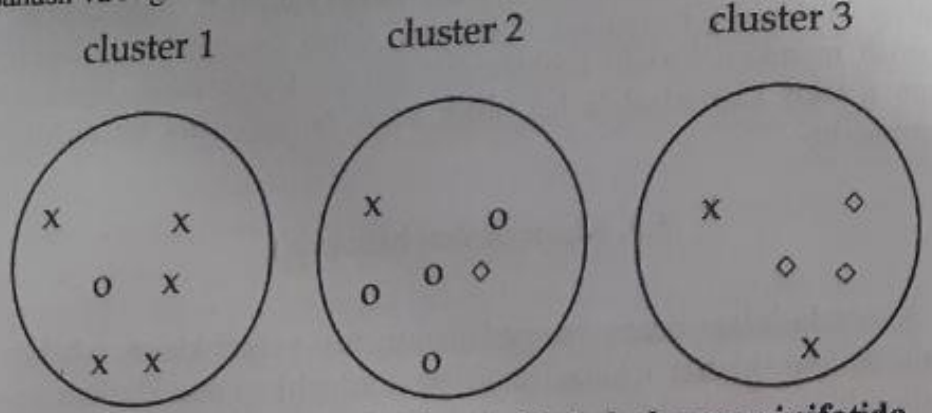
Maqsad maqsadli funksiyani optimallashtirish bo'lganligi sababli, klasterlash asosan qidiruv muammosidir. Qo'pol kuch yechimi barcha mumkin bo'lgan klasterlarni sanab o'tish va eng yaxshisini tanlashdir. Biroq, eksponent ravishda ko'p bo'limlar mavjud, shuning uchun bu yondashuv amalga oshirilmaydi. Shu sababli, ko'pchilik tekis klasterlash algoritmlari dastlabki bo'linishni iterativ tarzda aniqlaydi. Agar qidiruv noqulay boshlang'ich nuqtada boshlansa, global optimallikni o'tkazib yuborish mumkin. Yaxshi boshlanish nuqtasini topish kerak, shuning uchun u tekis klasterlashda hal qilish kerak bo'lgan yana bir muhim muammodir.

5.3. Klasterlashni baholash

Klasterlashdagi odatiy maqsad funksiyalari yuqori klaster ichidagi o'xshashlikka (klaster ichidagi hujjatlar o'xshash) va past klasterlararo o'xshashlikka (turli klasterlarning hujjatlari bir-biriga o'xshamaydi)

erishish maqsadini rasmiylashtiradi. Bu klasterlash sifatining ichki mezonidir. Ammo ichki mezon bo'yicha yaxshi ballar ilovada yaxshi samaradorlikka aylanishi shart emas. Ichki mezonlarga muqobil qiziqishni qo'llashda bevosita baholashdir. Qidiruv natijalarini klasterlash uchun o'lchash foydalanuvchilarning turli klasterlash algoritmlari bilan javob topishi zarur bo'lgan vaqt. Bu eng to'g'ridan-to'g'ri baholashdir lekin bu qimmat, ayniqsa, katta foydalanuvchi tadqiqotlari zarur bo'lsa. Foydalanuvchi mulohazalari uchun o'rinbosar sifatida baholash mezonlari yoki oltin standartidagi sinflar to'plamidan foydalanish mumkin. Oltin standart ideal darajada hakamlararo kelishuvga ega bo'lgan inson hakamlari tomonidan ishlab chiqariladi. Keyin klasterlashning oltin standart sinflariga qanchalik mos kelishini baholaydigan tashqi mezonni hisoblash mumkin. Masalan, 5.2-rasmdagi yaguar uchun qidiruv natijalarining optimal klasteri uchta sezgi: ***avtomobil***, ***hayvon*** va ***operatsion tizim***ga mos keladigan uchta sinfdan iborat ekanligini aytish mumkin. Ushbu turdagi baholashda sinf belgilaridan emas, balki faqat oltin standart tomonidan taqdim etilgan bo'limdan foydalaniladi.

Ushbu bo'limda klasterlash sifatining to'rtta tashqi mezonlari keltirilgan. Oddiy va shaffof baholash o'lchovidir. Normallashtirilgan o'zaro ma'lumot axborot-nazariy talqin qilinishi mumkin. ***Rand indeksi*** klasterlash paytida noto'g'ri ijobiy va noto'g'ri salbiy qarorlarni kamaytiradi. **F** o'lchovi qo'shimcha ravishda ushbu ikki turdagi xatolarning differentsial vaznini qo'llab-quvvatlaydi. Tozalikni hisoblash uchun har bir klaster klasterda eng ko'p uchraydigan sinfga yuklatiladi, so'ngra ushbu topshiriqning to'g'riligi to'g'ri berilgan hujjatlar sonini sanash va N ga bo'lish yo'li bilan o'lchanadi.



5.4-rasm. **Soflik klaster sifatini tashqi baholash mezoni sifatida**

Klasterlar sonining yuqori chegarasi **K** $N/K!$ **N** ta hujjatning **K** klasterlarga bo'linishlarining aniq soni ikkinchi turdagi Stirling sonidir. http://mathworld.wol`fram.com/StirlingNumberoftheSecondKind.h ml yoki Comtet (1974) ga qarang.

Ko'pchilik sinfi va uchta klaster uchun ko'pchilik sinfi a'zolari soni: x, 5 (klaster 1); o, 4 (klaster 2); va ◦, 3 (klaster 3). Soflik (1/17) × (5 + 4 + 3) = 0,71.

5.2-jadval. **56.4-rasmda klasterlash uchun qo'llaniladigan to'rtta tashqi baholash chorasi**

	purity	NMI	RI	F_5
lower bound	0.0	0.0	0.0	0.0
maximum	1	1	1	1
value for Figure 16.4	0.71	0.36	0.68	0.46

Rasmiy ravishda quyidagich hisoblanadi:

$$purity\ (\Omega, C) = \frac{1}{N}\sum_{k}\max_{j}|w_k \cap c_j| \qquad (5.1)$$

bu yerda klasterlar to'plami $P(\omega_k)$, $P(c_j)$, va $P(\omega_k \cap c_j)$ sinflar to'plamidir. Ular (5.1) tenglamadagi hujjatlar to'plami ω_k va c_j hujjatlar to'plami sifatida ω_k talqin qilinadi. c_j. 5.4-rasmda shovqindan tozalashni hisoblash misoli keltirilgan. Yomon klasterlar 0 ga yaqin soflik qiymatlariga ega, mukammal klasterlar esa aniqlikka ega.

$$NMI\ (\Omega, C) = \frac{I(\Omega, C)}{[H(\Omega) + H(C)]/2} \qquad (5.2)$$

5.2-jadvalda aniqlik ushbu bobda muhokama qilingan boshqa uchta o'lchov bilan taqqoslanadi.

$$I(\Omega, C) = \sum_{k}\sum_{j} P(w_k \cap c_j)\log\frac{P(w_k \cap c_j)}{P(w_k)P(c_j)}$$
$$= \sum_{k}\sum_{j}\frac{|w_k \cap c_j|}{N}\log\frac{N|w_k \cap c_j|}{|w_k||c_j|} \qquad (5.3)$$

$$NMQ\ (T, C) = \frac{I(\Omega, C)}{[H(\Omega) + H(C)]/2} \qquad (5.4)$$

Ayniqsa, klasterlar soni ko'p bo'lsa yuqori aniqlikka erishish oson, agar har bir hujjat o'z klasteriga ega bo'lsa aniqlik 1 ga teng. Shunday qilib, klasterlar soniga nisbatan klasterlash sifatini almashtirish uchun aniqlikdan foydalanish mumkin emas. Hujjatning mos ravishda klasterda, sinfda va kesishmalarida bo'lish ehtimoli normalangan o'zaro ma'lumotdir. Tenglama (5.4) ehtimolliklarning maksimal ehtimollik taxminlari uchun (5.3) tenglamaga ekvivalentdir ya'ni, har bir ehtimolning bahosi mos keladigan nisbiy chastotadir.

$$H(\Omega) = -\sum_k P(w_k) \log P(w_k) \tag{5.5}$$

$$= -\sum_k \frac{|w_k|}{N} \log \frac{|w_k|}{N} \tag{5.6}$$

bu yerda yana ikkinchi tenglama ehtimolliklarning maksimal ehtimollik baholariga asoslanadi. (5.3) tenglamadagi $(\Omega; C)$ klasterlar nima ekanligi aytilganda, sinflar haqidagi bilimlarimiz ortib borayotgan axborot miqdorini o'lchaydi. Klasterlash sinf a'zoligiga nisbatan tasodifiy bo'lsa minimal $I(\Omega; C)$ 0 ga teng. Bunday holda hujjatning ma'lum bir klasterda ekanligini bilish ularga uning sinfi qanday bo'lishi mumkinligi haqida yangi ma'lumot bermaydi. Klasslarni mukammal qayta yaratuvchi Ω_{max} klasterlash uchun maksimal o'zaro ma'lumotlarga erishiladi - lekin Ω_{max} klasterlar kichikroq klasterlarga bo'lingan bo'lsa ham. Xususan, K = N bitta hujjatli klasterli klasterlash maksimal MI ga ega. Shunday qilib, MI aniqlik bilan bir xil muammoga ega. U katta kardinalliklarni kamaytirmaydi va shuning uchun boshqa narsalar teng bo'lsa kamroq klasterlar yaxshiroq bo'lgan ularga yaqinlarni rasmiylashtirmaydi. (5.2) tenglamadagi $H(\Omega) + H(C)$ maxraji bo'yicha normallashtirish bu muammoni hal qiladi chunki entropiya klasterlar soni bilan ortib boradi. Misol, $H(\Omega) = N$ uchun **NMI** maksimal logorifmlanadi, bu K = N uchun NMI past bo'lishini ta'minlaydi. **NMI** normallashtirilganligi sababli, uni turli xil klasterlar soni bilan klasterlarni solishtirish uchun ishlatish mumkin. Maxrajning maxsus shakli tanlangan chunki $[H(\Omega) + H(C)]/2^{I(\Omega;C)}$ ning qattiq yuqori chegarasi mavjud. Shunday qilib, NMI har doim 0 dan 1 gacha bo'lgan raqamdir.

Klasterlashning ushbu axborot-nazariy talqiniga alternativa uni to'plamdagi **N(N-1)/2** juft hujjatlarning har biri uchun bittadan qarorlar seriyasi sifatida ko'rishdir. Ikkita hujjatni bir xil klasterga tayinlash

mumkin agar ular o'xshash bo'lsa. Haqiqiy ijobiy (TP) qaror bir xil klasterga ikkita o'xshash hujjatni tayinlaydi, haqiqiy salbiy (TN) qaror ikkita o'xshash bo'lmagan hujjatni turli klasterlarga belgilaydi. Ikki xil xatoga yo'l qo'yish mumkin. Noto'g'ri ijobiy (FP) qaror bir xil klasterga ikkita o'xshash bo'lmagan hujjatni belgilaydi. Noto'g'ri salbiy (FN) qarori ikkita o'xshash hujjatni turli klasterlarga belgilaydi. ***Rand indeksi*** **(RI)** RI to'g'ri qarorlar foizini o'lchaydi. Ya'ni, bu oddiygina aniqlikdir.

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

Misol tariqasida 5.4-rasm uchun **RI** ni hisoblash mumkin. Birinchi navbatda TP + FP hisoblanadi. Uchta klaster mos ravishda 6, 6 va 5 ballni o'z ichiga oladi shuning uchun bir xil klasterdagi "ijobiy" yoki juft hujjatlarning umumiy soni quyidagicha hisoblanadi:

$$TP + FP = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40$$

Ulardan 1-klasterdagi **x** juftlari, 2-klasterdagi o juftlari, 3-klasterdagi ⋄ juftlari va 3-klasterdagi x juftlari haqiqiy musbat hisoblanadi:

$$TP = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20$$

Shunday qilib, FP = 40 - 20 = 20.

FN va **TN** xuddi shunday hisoblab chiqiladi natijada quyidagi favqulodda vaziyatlar jadvali olinadi:

	Same cluster	Different clusters
Same class	TP = 20	FN = 24
Different classes	FP = 20	TN = 72

Rand indeksi noto'g'ri ijobiy va noto'g'ri salbiylarga teng vazn beradi. O'xshash hujjatlarni ajratish ba'zan bir xil klasterga bir-biriga o'xshash bo'lmagan hujjatlarni qo'yishdan ko'ra yomonroqdir. **F** o'lchovidan **b > 1** qiymatini tanlab, noto'g'ri negativlarni noto'g'ri musbatlarga qaraganda kuchliroq kamaytirish uchun foydalanish mumkin, bu esa eslash uchun ko'proq og'irlik qiladi.

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \quad F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

Favqulodda vaziyatlar jadvalidagi raqamlarga asoslanib, **P = 20/40 = 0,5** va **R = 20/44 ≈ 0,455**. Bu ularga **b = 1** uchun **F_1 ≈ 0,48** va **b = 5** uchun

$F_5 \approx 0{,}456$ ni beradi. Axborotni qidirishda **F** bilan klasterlashni baholash - o'lchov tadqiqot sohasiga allaqachon tanish bo'lgan afzalliklarga ega.

Misol. 5.4-rasmdagi har bir **d** nuqtani bir sinfdagi **d** ning ikkita bir xil nusxasi bilan almashtiring. (i) 5.4-rasmdagi 17 balldan farqli o'laroq bu 34 balldan iborat to'plamni to'plash bir xil darajada qiyinmi yoki qiyinroqmi? (ii) 34 ball bilan klasterlash uchun aniqlik, **NMI**, **RI** va F_5 ni hisoblang. Ballar sonini ikki baravar oshirgandan keyin qaysi ko'rsatkichlar ortadi va qaysi biri o'zgarmaydi? (iii) (i) dagi baholash va (ii) dagi natijalar hisobga olinsa, ikkita klaster sifatini solishtirish uchun qaysi ko'rsatkichlar eng mos keladi?

5.4. K- means

K-means eng muhim yassi klasterlash algoritmidir. Uning maqsadi klaster markazi klasterdagi hujjatlarning o'rtacha yoki **centroid** ~ μ si sifatida belgilangan klaster markazlaridan hujjatlarning o'rtacha kvadratik *Evklid masofasini* minimallashtirishdan iborat:

$$\vec{\mu}(w) = \frac{1}{|w|}\sum_{\vec{x}\in w}\vec{x}$$

Hujjatlar tanish tarzda haqiqiy qiymatli fazoda uzunlik normalangan vektorlar sifatida taqdim etilishini nazarda tutadi. 3-bobda ***Rocchio tasnifi*** uchun centroidlardan foydalanilgan. Bu yerda xuddi shunday rol o'ynaydi. K-o'rtachadagi ideal klaster bu markazning tortishish markazi bo'lgan shardir. Ideal holda, klasterlar bir-birining ustiga tushmasligi kerak. ***Rokkio tasnifidagi*** darslarga bo'lgan xohish bir xil edi. Farqi shundaki, ularda klasterda qaysi hujjatlar bo'lishi kerakligini biladigan yorliqli o'quv majmuasi yo'q. Markazlarning o'z klasterlari a'zolarini qanchalik yaxshi ifodalashining o'lchovi kvadratlarning qoldiq yig'indisi yoki **RSS**, har bir vektorning markazidan barcha vektorlar bo'yicha yig'ilgan kvadrat masofasidir:

$$RSS_k = \sum_{\vec{x}\in w_k}|\vec{x} - \vec{\mu}(w_k)|^2 \qquad (5.7)$$

$$RSS = \sum_{k=1}^{K} RSS_k$$

RSS K-vositalaridagi maqsad funksiyasi va maqsad uni minimallashtirishdir. **N** sobit bo'lganligi sababli, **RSS**ni minimallashtirish o'rtacha kvadrat masofani minimallashtirishga teng, ya'ni centroidlar ularning hujjatlarini qanchalik yaxshi aks ettiruvchi o'lchovdir.

```
K-MEANS({x⃗_1, ..., x⃗_N}, K)
1  (s⃗_1, s⃗_2, ..., s⃗_K) ← SELECTRANDOMSEEDS({x⃗_1, ..., x⃗_N}, K)
2  for k ← 1 to K
3  do μ⃗_k ← s⃗_k
4  while stopping criterion has not been met
5  do for k ← 1 to K
6     do ω_k ← {}
7     for n ← 1 to N
8     do j ← arg min_j |μ⃗_j − x⃗_n|
9        ω_j ← ω_j ∪ {x⃗_n}   (reassignment of vectors)
10    for k ← 1 to K
11    do μ⃗_k ← (1/|ω_k|) Σ_{x⃗∈ω_k} x⃗   (recomputation of centroids)
12 return {μ⃗_1, ..., μ⃗_K}
```

5.5-rasm. **K-o'rtacha algoritmi**

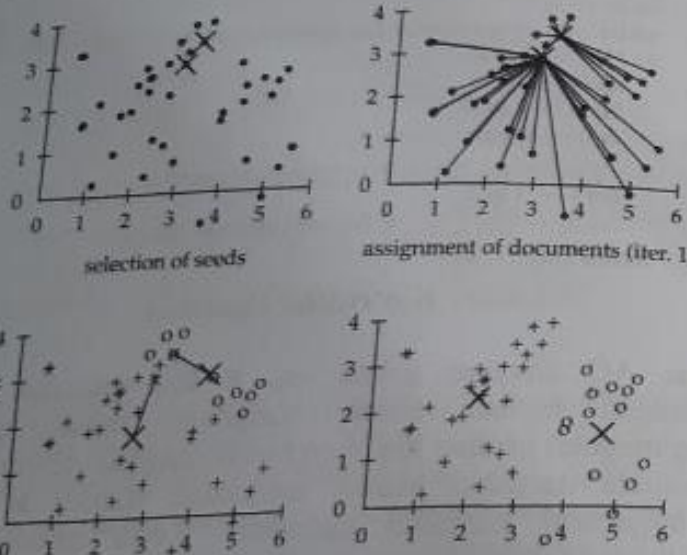
Ko'pgina AQ ilovalari uchun ~$\mathbf{x_n} \in \mathbf{R_M}$ vektorlari uzunligi normallashtirilgan bo'lishi kerak. Natijalarni tanlash va ishga tushirishning muqobil usullari keyinroq ko'rib chiqiladi. K-vositalarning birinchi bosqichi dastlabki klaster markazlari sifatida **K** tasodifiy tanlangan hujjatlarni natijasini tanlashdir. Keyin algoritm **RSS**ni minimallashtirish uchun klaster markazlarini kosmosda harakatga keltiradi. 5.5-rasmda ko'rsatilganidek, bu to'xtash mezoni bajarilgunga qadar ikki bosqichni takrorlash orqali takrorlanadi. Hujjatlarni eng yaqin markazga ega bo'lgan klasterga qayta tayinlash va har bir markazni uni klasterining joriy a'zolari asosida qayta hisoblash. 5.6-rasmda nuqtalar to'plami uchun ***K-o'rtacha algoritmining*** to'qqizta takrorlanishidan olingan suratlar ko'rsatilgan. 6.2-jadvalning "markaziy" ustunida centroidlarning misollari keltirilgan. Quyidagi tugatish shartlaridan birini qo'llash mumkin.

- Iteratsiyalarning belgilangan soni bajarildi. Bu holat klasterlash algoritmining ishlash vaqtini cheklaydi lekin ba'zi hollarda iteratsiyalar soni yetarli emasligi sababli klasterlash sifati yomon bo'ladi.
- Hujjatlarni klasterlarga belgilash (g bo'lish funksiyasi) iteratsiyalar orasida o'zgarmaydi. Mahalliy minimal va minimal bo'lgan holatlar

bundan mustasno, bu yaxshi klasterlashni keltirib chiqaradi, ammo ish vaqtlari qabul qilib bo'lmaydigan darajada uzoq bo'lishi mumkin.

• **Centroids ~μ** takrorlashlar orasida o'zgarmaydi. Bu **g** ning o'zgarmasligiga teng.

• **RSS** chegaradan pastga tushganda tugating. Ushbu mezon klasterlash tugatilgandan keyin kerakli sifatga ega bo'lishini ta'minlaydi. Amalda, tugatishni kafolatlash uchun uni takrorlash soni bo'yicha chegara bilan birlashtirish kerak.



5.6-rasm. **A K-R₂ da K = 2 ga misol degan ma'noni anglatadi**

Ikki markazning joylashuvi (yuqori to'rtta panelda **~m'** sifatida ko'rsatilgan) to'qqiz marta takrorlangandan so'ng birlashadi.

• **RSS pasayish th** chegarasidan pastga tushganda tugatiladi. Kichik **th** uchun bu yaqinlashishga yaqin ekanligini ko'rsatadi. Shunga qaramay, juda uzoq ish vaqtlarini oldini olish uchun uni iteratsiyalar soni bo'yicha chegara bilan birlashtirish kerak. Endi **RSS** ning har bir iteratsiyada ***monoton ravishda*** kamayib borishini isbotlash orqali **K**-vositalarining yaqinlashishi ko'rsatiladi. Ushbu bo'limda ma'noni kamaytirish yoki o'zgarmasligini kamaytirishdan foydalaniladi.

Birinchidan, qayta tayinlash bosqichida **RSS** kamayadi chunki har bir vektor eng yaqin markazga tayinlanadi, shuning uchun uning ***RSSga hissa*** qo'shadigan masofasi kamayadi. Ikkinchidan, u qayta hisoblash bosqichida kamayadi chunki yangi **centroid vektor ~v RSS k** minimal darajaga yetadi.

$$RSS_k(\vec{v}) = \sum_{\vec{x}\in w_k} |\vec{v} - \vec{x}|^2 = \sum_{\vec{x}\in w_k}\sum_{m=1}^{M} (\vec{v}_m - \vec{x}_m)^2 \tag{5.8}$$

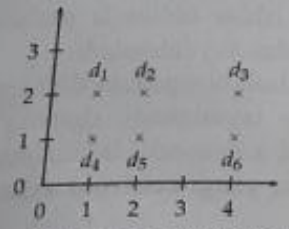
$$\frac{\partial RSS_k(\vec{v})}{\partial v_m} = \sum_{\vec{x}\in w_k} 2(v_m - x_m) \tag{5.9}$$

bu yerda $\mathbf{x_m}$ va $\mathbf{v_m}$ tegishli vektorlarining **m-** komponentlari. Qisman lotinni nolga o'rnatib quyidagicha hisoblanadi:

$$v_m = \frac{1}{|w_k|}\sum_{\vec{x}\in w_k} x_m \tag{5.8}$$

Bu markazning komponentli ta'rifidir. Shunday qilib, eski **centroid** yangi markaz bilan almashtirilganda **RSS k** ni minimallashtiradi. **RSS**, ya'ni **RSS k** yig'indisi, keyin qayta hisoblash paytida ham kamayishi kerak. Mumkin bo'lgan klasterlarning faqat cheklangan to'plami mavjud bo'lganligi sababli, ***monoton ravishda*** pasayuvchi algoritm oxir-oqibat minimal darajaga yetadi. Biroq, aloqalarni doimiy ravishda uzish uchun ehtiyot bo'ling masalan, bir nechta teng masofadagi markazlar mavjud bo'lsa hujjatni eng past indeksli klasterga belgilash orqali aniqlanadi. Aks holda algoritm bir xil narxga ega bo'lgan klasterlar qatorida abadiy aylanishi mumkin. Bu **K**-o'rtachalarning yaqinlashuvini isbotlasa-da, afsuski, maqsad funksiyasida global minimumga erishish kafolati yo'q. Hujjatlar to'plamida boshqa hujjatlardan uzoq bo'lgan va shuning uchun

hech qanday klasterga yaxshi mos kelmaydigan ko'plab chet elliklar mavjud bo'lsa bu alohida muammo hisoblanadi. Ko'pincha, agar boshlang'ich urg'u sifatida chet elchi tanlansa keyingi iteratsiyalarda unga boshqa vektor tayinlanmaydi. Shunday qilib, bitta klasterga (faqat bitta hujjatga ega klaster) ega bo'lamiz, garchi pastroq **RSS** bilan klaster mavjud bo'lsa ham. 5.7-rasmda boshlang'ich urg'ularni noto'g'ri tanlash natijasida kelib chiqadigan ***suboptimal klasterlash*** misoli ko'rsatilgan.



5.7-rasm. **K-vositalarda klasterlashning natijasi**

d_2 va d_5 urg'ulari uchun ***K-o'rtacha*** $\{\{d_1, d_2, d_3\}, \{d_4, d_5, d_6\}\}$ ga yaqinlashadi, suboptimal klasterlash. d_2 va d_3 urug'lari uchun u $\{\{d_1,d_2, d_4, d_5\}, \{d_3, d_6\}\}$ ga yaqinlashadi, $K = 2$ uchun global optimal hisoblanadi.

Tez-tez uchraydigan ***suboptimal klasterlash***ning yana bir turi bo'sh klasterlarga ega. Urg'ularni tanlash uchun samarali evristika quyidagilarni o'z ichiga oladi. (i) urg'ulik to'plamidan chetga chiqqanlarni istisno qilish. (ii) bir nechta boshlang'ich nuqtalarni sinab ko'rish va eng kam xarajat bilan klasterlashni tanlash va (iii) ierarxik klasterlash kabi boshqa usuldan urg'ularni olish. ***Deterministik ierarxik klasterlash*** usullari ***K-o'rtachalarga*** qaraganda ko'proq bashorat qilinadigan bo'lgani uchun **iK** o'lchamdagi kichik tasodifiy namunaning ierarxik klasteri (masalan, i = 5 yoki i = 10 uchun) ko'pincha yaxshi urg'ularni beradi (***Buckshot algoritmining*** tavsifiga qarang, 6-bob). Boshqa ishga tushirish usullari klaster qilinadigan vektorlardan tanlanmagan urg'ularni hisoblab chiqadi. Hujjatlarni tarqatishning keng turlari uchun yaxshi ishlaydigan ishonchli usul har bir klaster uchun **i** (masalan, i = 10) tasodifiy vektorlarni tanlash va ularning centroidini ushbu klaster uchun urg'u sifatida ishlatishdir. Murakkab ishga tushirishlar uchun 5.6-bo'limga qarang. ***K-vositalar***ning vaqt murakkabligi nima? Ko'p vaqt vektor masofalarini hisoblash uchun sarflanadi. Bunday operatsiyalardan biri $\Theta(M)$ turadi. Qayta tayinlash bosqichi **KN** masofalarini hisoblaydi, shuning uchun uning umumiy

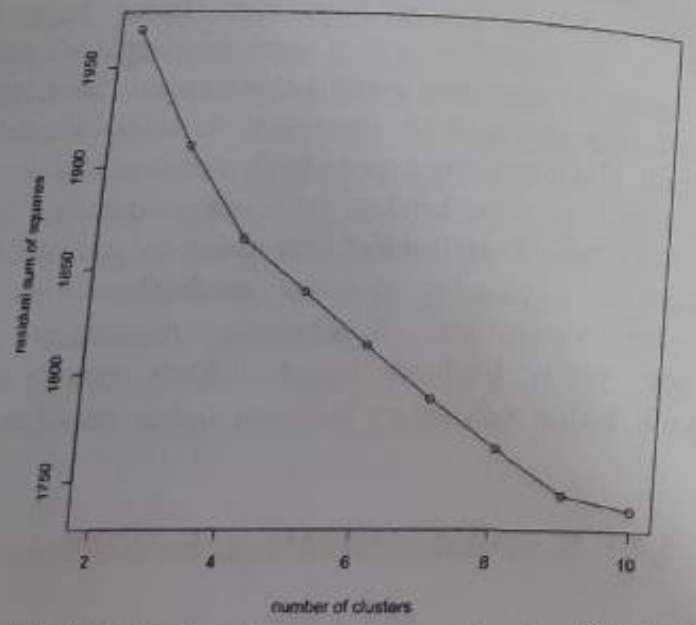
murakkabligi $\Theta(KNM)$ ga teng. Qayta hisoblash bosqichida har bir vektor markazga bir marta qo'shiladi, shuning uchun bu bosqichning murakkabligi $\Theta(NM)$ ga teng. Ruxsat etilgan takrorlash soni uchun I, umumiy murakkablik shuning uchun $\Theta(IKNM)$ ga teng. Shunday qilib, ***K-o'rtacha*** barcha tegishli omillarda chiziqli: iteratsiyalar, klasterlar soni, vektorlar soni va fazoning o'lchovliligi. Bu shuni anglatadiki, ***K-vositalari*** 6-bobdagi ierarxik algoritmlarga qaraganda samaraliroq. Takrorlashlar sonini tuzatish zarur, bu amalda qiyin bo'lishi mumkin. Ammo ko'p hollarda **K** tezda to'liq yaqinlashuvga yoki yaqinlashuvga yaqin bo'lgan klasterga erishadi. Ikkinchi holda, agar keyingi iteratsiyalar hisoblansa bir nechta hujjatlar a'zolikni o'zgartiradi, ammo bu klasterlashning umumiy sifatiga ozgina ta'sir qiladi holos.

Oldingi argumentda bitta noqulaylik mavjud. Agar $\Theta(...)$ argumentlaridan biri - **M** odatda katta bo'lsa hatto chiziqli algoritm juda sekin bo'lishi mumkin. Yuqori o'lchamlilik ikki hujjat orasidagi masofani hisoblash uchun muammo emas. Ularning vektorlari siyrak, shuning uchun nazariy jihatdan mumkin bo'lgan **M** komponentli farqlarning faqat kichik qismini hisoblash kerak. Biroq, ***centroidlar*** zichdir chunki ular o'z klasterlarining har qanday hujjatlarida uchraydigan barcha atamalarni birlashtiradi. Natijada, masofaviy hisob-kitoblar ***K-vositalar***ni sodda tarzda amalga oshirishda ko'p vaqt talab etadi. Biroq, hujjat - hujjat o'xshashliklari kabi tez hisoblash uchun markaziy hujjat o'xshashligini yaratish uchun oddiy va ***samarali evristikalar*** mavjud. Markazlarni eng muhim **k** atamalarga qisqartirish (masalan, k=1000) klaster sifatini deyarli pasaytiradi, shu bilan birga qayta biriktirish bosqichining sezilarli tezlashishiga erishadi (5.6-bo'limdagi havolalarga qarang). Xuddi shu samaradorlik muammosi ***K-medoidlar*** tomonidan hal qilinadi, bu klaster markazlari sifatida markazlar o'rniga ***medoidlar***ni hisoblaydigan vositalarning bir variantidir. Klasterning medioidini ***MEDOID centroidiga*** eng yaqin bo'lgan hujjat vektori sifatida belgilaydi. Medoidlar siyrak hujjat vektorlari bo'lgani uchun masofani hisoblash tezdir.

5.4.1. K-vositalaridagi klaster kardinalligi

5.2-bo'limda ***K klasterlar*** soni ko'pchiligi yassi klasterlash algoritmlariga kirish ekanligini aytdgan edik. Agar ular **K** uchun ishonchli

taxmin qilinsa, nima bo'ladi? Maqsad funksiyasi bo'yicha **K** ning optimal qiymatini, ya'ni **RSS**ni minimallashtiradigan **K** qiymatini tanlash sodda yondashuv bo'ladi. $RSS_{min}(K)$ ni **K** klasterli barcha klasterlarning *minimal* **RSS**-si sifatida belgilab, ularni **K** da $RSS_{min}(K)$ monoton kamayuvchi funksiya ekanligini ko'rish mumkin, u **K=N** uchun minimal 0 ga yetadi, bu yerda N – hujjatlar soni. Ular har bir hujjatning o'z klasterida bo'lishi bilan yakunlanadi. Shubhasiz, bu optimal klasterlash emas. Ushbu muammoni hal qiladigan ***evristik usul*** $RSS_{min}(K)$ ni quyidagicha baholanadi. Birinchi navbatda **K** klasterlari bilan **i** (masalan, $i = 10$) klasterlari amalga oshiriladi (har biri boshqa ishga tushirilgan) va har birining **RSS**-ni hisoblanadi. Keyin **i RSS** qiymatlarining minimali olinadi. Bu minimum $RSS_{min}(K)$ bilan belgilanadi. Endi $RSS_{min}(K)$ qiymatlari **K** ortishi bilan tekshirishiriladi va egri chiziqdagi "tizza"ni topishimiz mumkin – bu nuqta **d** RSS_{min} ning ketma-ket pasayishi sezilarli darajada kichiklashadi. 5.8-rasmda ikkita shunday nuqta bor, biri **K = 4** da, bu yerda *gradient* biroz tekislanadi va **K=9** da aniqroq tekislanadi. Bu odatiy holder. Klasterlarning eng yaxshi soni kamdan-kam uchraydi. **K** ning bir qancha mumkin bo'lgan qiymatlaridan tanlash uchun hali ham tashqi cheklovdan foydalanish kerak.



5.8-rasm. **K-o'rtachadagi klasterlar soniga bog'liq bo'lgan kvadratlarning minimal qoldiq yig'indisi**

1203 ta **Reuters-RCV1** hujjatlar klasterida **d** RSS_{min} egri chizig'i tekislanadigan ikkita nuqta mavjud. 4 ta klasterda va 9 ta klasterda. Hujjatlar Xitoy, Germaniya, Rossiya va Sport toifalaridan tanlangan, shuning uchun $K = 4$ klasteri ***Reuters*** tasnifiga eng yaqin.

Klaster kardinalligi mezonining ikkinchi turi har bir yangi klaster uchun talab qo'yadi - bu yerda kontseptual jihatdan barcha hujjatlarni o'z ichiga olgan yagona klasterdan boshlanadi va keyin **K** ni ketma-ket bittaga oshirish orqali ***K klasterlarining optimal sonini*** qidiriladi. Klasterning kardinalligini shu tarzda aniqlash uchun ikkita elementni birlashtirgan umumlashtirilgan maqsadli funksiya yaratiladi. Buzilish, hujjatlar o'z klasterlarining prototipidan qanchalik og'ishini o'lchovi (masalan, K-vositalari uchun RSS) va model murakkabligi o'lchovi. Bu yerda klasterlashni ***ma'lumotlar modeli*** sifatida izohlanadi. Klasterlashdagi modelning murakkabligi odatda klasterlar soni yoki ularning funksiyasidir. K-vositalari uchun **K** ning ushbu tanlov mezonini olinadi.

$$K = \arg\min_{K}[\, RSS_{min}(K) + \lambda K \,] \qquad (5.11)$$

bu yerda λ - og'irlik omili. λ ning katta qiymati bir nechta klasterli jarima yo'q va **K=N** eng yaxshi yechimdir.

(5.11) tenglamaning aniq qiyinligi shundaki, unda λ ni aniqlash kerak. Agar bu to'g'ridan-to'g'ri **K** ni aniqlashdan osonroq bo'lmasa birinchi kvadratga qaytiladi. Ba'zi hollarda avvagi o'xshash ma'lumotlar to'plamlari uchun yaxshi ishlagan λ qiymatlarini tanlash mumkin. Misol uchun, agar vaqti-vaqti bilan yangiliklar kanalidan yangiliklarni klaster qilinsa har bir keyingi klasterda ularga to'g'ri **K** ni beradigan λ ning qat'iy qiymati bo'lishi mumkin. Ushbu ilovada **K** ni o'tgan tajribaga asoslangan holda aniqlash mumkin chunki **K** o'zgaradi. (5.11) tenglamaning nazariy asoslanishi bu ***Akaike axborot mezoni*** yoki *AIC* bo'lib, buzilishni model murakkabligi bilan almashtiradigan axborot-nazariy o'lchovdir. **AIC**ning umumiy shakli quyidagicha hisoblanadi:

$$AIC: \quad K = \arg\min_{K} [-2L(K) + 2qK] \qquad (5.12)$$

Bu yerda **K** klasterlari uchun ma'lumotlarning manfiy maksimal log-ehtimolligi - **L(K)** buzilish o'lchovidir va **K** klasterli modelning parametrlari soni **q(K)** model murakkabligining o'lchovidir. Bu yerda **AIC**ni olish shart emas lekin uni ***intuitiv*** ravishda tushunish oson.

Ma'lumotlarning yaxshi modelining birinchi xususiyati shundaki, har bir ma'lumot nuqtasi model tomonidan yaxshi modellashtirilgan. Bu past buzilishning maqsadidir.

Ammo modellar ham kichik bo'lishi kerak chunki faqat ma'lumotlarni tavsiflovchi (shuning uchun nol buzilishlarga ega) model foydasizdir. **AIC** *modelni tanlashda ushbu ikki omilni, ya'ni buzilish va modelning murakkabligini baholashning muayyan usulini nazariy asoslab beradi.* ***K-vositalari*** *uchun* **AIC***ni quyidagicha ifodalash mumkin:*

$$AIC: \quad K = \arg\min_{K} [RSS_{\min}(K) + 2MK] \qquad (5.13)$$

(5.13) tenglamada $\lambda = 2M$ *uchun (5.11) tenglamaning maxsus holatidir. (5.12) tenglamadan (5.13) tenglamani olish uchun* $q(K) = KM$ *- degani ekanligini kuzating chunki* **K** *markazlarining har bir elementi mustaqil ravishda o'zgarishi mumkin bo'lgan parametrdir va agar* ***K-vositalari*** *asosida yotgan modelni qattiq tayinlangan, bir xil klaster avariyalari va bir xil sferik* ***kovariatsiya matritsalari*** *bilan* ***Gauss aralashmasi*** *sifatida ko'rilsa,* $L(K) = -(1/2)RSS_{\min}(K)$ *(sobit moduli) bo'ladi.* **AIC***ni chiqarish bir qator taxminlarga asoslanadi, masalan, ma'lumotlar mustaqil va bir xil taqsimlangan.*

Ushbu taxminlar faqat ma'lumotni qidirishda ma'lumotlar to'plami uchun taxminan to'g'ri.

Natijada, **AIC** *kamdan-kam hollarda matn klasterini o'zgartirmasdan qo'llanilishi mumkin. 5.8-rasmda vektor fazosining o'lchovliligi* $M \approx 50{,}000$ *ga teng. Shunday qilib,* $RSS_{\min}(1) < 5000$, *asoslangan atamada ustunlik qiladi (d* $RSS_{min(1)}<5000$, *rasmda ko'rsatilmagan) va ifodaning minimaliga* **K = 1** *uchun erishiladi. Lekin bilganimizdek, K = 4 (Xitoy, Germaniya, Rossiya va Sport) to'rtta sinfga to'g'ri keladigan* **K=1** *dan ko'ra yaxshiroq tanlovdir. Amalda (5.11) tenglama (5.13) ga qaraganda ko'pincha foydaliroqdir - ogohlantirish bilan l uchun taxmin?*

Misol. Nima uchun kontsept-kar uchun bir xil atama ishlatilmaydigan hujjatlar K-means klasterlashda bir xil klasterda tugashi mumkin?

Misol. K-vositalari uchun mumkin bo'lgan tugatish shartlaridan ikkitasi: (1) tayinlash o'zgarmaydi, (2) markazlar o'zgarmaydi. Bu ikki shart bir-birini anglatadimi?

5.5. Modelga asoslangan klasterlash

Ushbu bo'limda K-o'rtachalarning umumlashtirilishini, EM algoritmi tasvirlanadi. U K-vositalariga qaraganda ko'proq turli xil hujjatlar taqdimoti va tarqatish uchun qo'llanilishi mumkin. K-da yaxshi vakil bo'lgan ***centroidlarni*** topishga harakat qiladi. ***K centroidlar to'plami***ni ma'lumotlarni ishlab chiqaruvchi model sifatida ko'rish mumkin. Ushbu modeldagi hujjatni yaratish dastlab markazni tasodifiy tanlash va keyin biroz shovqin qo'shilishidan iborat. Agar shovqin normal taqsimlangan bo'lsa bu protsedura sharsimon shakldagi klasterlarga olib keladi. Modelga asoslangan klasterlash ma'lumotlar model tomonidan yaratilgan deb taxmin qilinadi va ma'lumotlardan asl modelni tiklashga harakat qilinadi. Ular ma'lumotlardan tiklanadigan modeldan keyin klasterlarni va hujjatlarni klasterlarga tayinlashni belgilaydi.

Model parametrlarini baholash uchun tez-tez ishlatiladigan mezon - maksimal ehtimollikdir. K-da, exp(−RSS) miqdori ma'lum bir model (centroidlar to'plami) ma'lumotlarni yaratish ehtimoli bilan mutanosibdir. ***K-vositalari*** uchun maksimal ehtimollik va minimal RSS ekvivalent mezondir. Ular model parametrlari Θ bilan belgilanadi. K-da $\Theta = \{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$ bilan belgilanadi. Umuman olganda, maksimal ehtimollik mezoni **D** ma'lumotlarini yaratishning log-ehtimolligini maksimal darajada oshiradigan parametrlarni **D** tanlashdir:

$$\Theta = \arg\max_{\Theta} L(D \mid \Theta) = \arg\max_{\Theta} \log \prod_{n=1}^{N} P(d_n \mid \Theta) = \arg\max_{\Theta} \sum_{n=1}^{N} P(d_n \mid \Theta)$$

$L(D|\Theta)$ $L(D \mid \Theta)$ - klasterning yaxshiligini o'lchaydigan maqsad funksiyasidir. Klasterlar soni bir xil bo'lgan ikkita klasterni hisobga olsak **L(D|D)** yuqori bo'lganini afzal ko'riladi. Tilni modellashtirish uchun 1-bobda va matn tasnifi uchun 2.1-bo'limda xuddi shunday yondashuv mavjud. Matnni tasniflashda ma'lum bir hujjatni yaratish ehtimolini maksimal darajada oshiradigan sinf tanlanadi. Bu yerda ma'lum hujjatlar to'plamini yaratish ehtimolini maksimal darajada oshiradigan **D** klasteri tanlanadi. Ularda Θ ga ega bo'lgandan so'ng, har bir hujjat-klaster juftligi uchun $P(d \mid w_k; \Theta)$ tayinlanish ehtimolini hisoblash mumkin. Ushbu tayinlash ehtimolining to'plami yumshoq klasterlashni belgilaydi. Yumshoq topshiriqning misoli shundaki, Xitoy avtomobillari haqidagi

hujjat Xitoy va avtomobillar klasterlarining har birida 0,5 dan kichik a'zolikka ega bo'lishi mumkin, bu ikkala mavzuning ham tegishli ekanligini aks ettiradi. K-vositalari kabi qattiq klaster bir vaqtning o'zida ikkita mavzuga tegishliligini modellashtira olmaydi.

Modelga asoslangan klasterlash domen haqidagi bilimlarimizni birlashtirish uchun asos yaratadi. K-vositalari va 6-bobdagi ierarxik algoritmlar ma'lumotlar haqida juda qat'iy taxminlarni keltirib chiqaradi. Masalan, ***K-o'rtachadagi klasterlar*** sharlar sifatida qabul qilinadi. Modelga asoslangan klasterlash ko'proq moslashuvchanlikni ta'minlaydi. Klasterlash modeli ma'lumotlarning asosiy taqsimoti haqida bilgan narsaga moslashtirilishi mumkin, xoh u ***Bernoulli*** (5.3-jadvaldagi misolda bo'lgani kabi), ***sferik*** bo'lmagan dispersiyaga ega ***Gauss*** (hujjatlarni klasterlashda muhim bo'lgan boshqa model) yoki a'zo bo'lsin. Modelga asoslangan klasterlash uchun tez-tez ishlatiladigan algoritm bu ***Expectation Maximization algoritmi*** yoki ***EM algoritmi***dir. ***EM klasterlash*** L(D|T) ni maksimal darajaga ko'taruvchi iterativ algoritmdir. **EM** ko'p turli xil ehtimollik modellashtirish uchun qo'llanilishi mumkin. Bu yerda ko'p o'zgaruvchan ***Bernoulli*** taqsimotlari aralashmasi bilan ishlanadi, ular birinchi qismning 11.3-bo'lim va 2.3-bo'limda keltirilgan.

$$P(d \mid w_k;\Theta) = \left(\prod_{t_m \in d} q_{mk}\right)\left(\prod_{t_m \notin d} (1-q_{mk})\right) \qquad (5.14)$$

$\Theta = \{\Theta_1,\ldots,\Theta_K\}, \Theta_k = (\alpha_k, q_{1k},\ldots,q_{Mk})$ va $q_{mk} = P(U_m = 1 \mid w_k)$

model parametrlari hisoblanadi. $P(U_m = 1 \mid w_k)$ klasteridagi hujjatda $\mathbf{t_m}$ atamasi mavjudligi ehtimoli. $\boldsymbol{\alpha_k}$ ehtimolligi $\tilde{\mathbf{o}}_\mathbf{k}$ klasterining oldingisi: **d** hujjatining α_k da bo'lish ehtimolidir, agar ularda **d** haqida ma'lumot bo'lmasa. Keyin aralashmaning modeli hisoblanadi:

$$P(d \mid \Theta) = \sum_{k=1}^{K} \alpha_k \left(\prod_{t_m \in d} q_{mk}\right)\left(\prod_{t_m \notin d} (1-q_{mk})\right) \qquad (5.15)$$

Ushbu modelda avval $\boldsymbol{\alpha_k}$ ehtimollik bilan **k** klasterni tanlab, so'ngra $\mathbf{q_{mk}}$ parametrlariga muvofiq hujjat shartlarini hosil qilish orqali hujjatni yaratamiz. Eslatib o'tamiz, ***Bernulli*** ko'p o'zgaruvchanligining hujjatdagi

ko'rinishi **M** mantiqiy qiymatlar vektoridir (haqiqiy qiymatli vektor emas).

U_m- ***Bernoulli Naive Bayes*** modeli uchun 2.3-bo'limda aniqlagan tasodifiy o'zgaruvchidir. U 1 (hujjatda $\mathbf{t_m}$ atamasi mavjud) va 0 (hujjatda $\mathbf{t_m}$ atamasi mavjud emas) qiymatlarini oladi.

Ma'lumotlardan klasterlash parametrlarini aniqlash uchun **EM** dan qanday foydalaniladi? Ya'ni, **L(D|T)** ni maksimallashtiruvchi **D** parametrlarni qanday tanlaymiz? **EM** K-vositalariga o'xshaydi chunki u qayta tayinlashga mos keladigan kutish bosqichi va model parametrlarini qayta hisoblashga mos keladigan maksimallashtirish bosqichi o'rtasida almashinadi. K-o'rtachalar parametrlari centroidlar, bu bo'limdagi **EM** misolining parametrlari $\mathbf{a_k}$ va $\mathbf{q_{mk}}$. Maksimallashtirish bosqichi parametrlarni $\mathbf{q_{mk}}$ va $\mathbf{a_k}$ **a** priorlarini quyidagicha hisoblab chiqadi:

Maksimallashtirish bosqichi:

$$q_{mk} = \frac{\sum_{n=1}^{N} r_{nk} I(t_m \in d_n)}{\sum_{n=1}^{N} r_{nk}} \qquad \alpha_k = \frac{\sum_{n=1}^{N} r_{nk}}{N} \qquad (5.16)$$

Bu yerda $\mathbf{I(t_m \in d_n) = 1}$, agar $\mathbf{t_m \in d_n}$ va aks holda 0 bo'lsa va $\mathbf{r_{nk}}$ oldingi iteratsiyada hisoblangan **k** klasteriga $\mathbf{d_n}$ hujjatining oson tayinlanishidir (bir zumda ishga tushirish masalasi ko'rib chiqiladi). Bular 2.3-jadvaldagi ko'p o'zgaruvchan ***Bernulli parametrlari*** uchun maksimal ehtimollik taxminlaridir. Bundan tashqari hujjatlar bu yerda klasterlarga qisman tayinlangan. Ushbu maksimal ehtimollik taxminlari model berilgan ma'lumotlarning ehtimolini maksimal darajada oshiradi. Kutish bosqichi joriy $\mathbf{q_{mk}}$ va $\mathbf{a_k}$ parametrlarini hisobga olgan holda klasterlarga hujjatlarning yumshoq tayinlanishini hisoblaydi.

Kutish bosqichi:

$$r_{nk} = \frac{\alpha_k\left(\prod_{t_m \in d_n} q_{mk}\right)\left(\prod_{t_m \notin d_n} (1-q_{mk})\right)}{\sum_{k=1}^{K} \alpha_k\left(\prod_{t_m \in d_n} q_{mk}\right)\left(\prod_{t_m \notin d_n} (1-q_{mk})\right)} \qquad (5.16)$$

Ushbu kutish bosqichi (5.14) va (5.15) tenglamalarni $\tilde{\mathbf{o}}_\mathbf{k}$ hujjat $\mathbf{d_n}$ hosil qilish ehtimolini hisoblash uchun qo'llaniladi. Bu 2.3-jadvaldagi ko'p o'zgaruvchan ***Bernulli*** uchun tasniflash protsedurasidir. Shunday qilib, kutish bosqichi **Bernoulli Naive Bayes** tasnifidan boshqa narsa

emas (shu jumladan normallashtirish, ya'ni klasterlar bo'yicha ehtimollik taqsimotini olish uchun maxrajga bo'lish). Ular 5.3-jadvalda EM dan foydalangan holda 11 ta hujjatlar to'plamini ikkita klasterga birlashtiradi. 25-iteratsiyada konvergentsiyadan so'ng dastlabki 5 ta hujjat 1-klasterga ($r_{i,1} = 1{,}00$) va oxirgi 6 tasi 2-klasterga ($r_{i,1} = 0{,}00$) tayinlanadi. Odatda, yakuniy topshiriq bu yerda qiyin topshiriqdir. EM odatda yumshoq topshiriqga yaqinlashadi. 25-iteratsiyada 1-klaster uchun oldingi α_1 5/11 $\approx 0{,}45$ ni tashkil qiladi chunki 11 ta hujjatdan 5 tasi 1-klasterda joylashgan. Masalan, 2-klasterga a'zolik birinchi iteratsiyada 7-hujjatdan 8-hujjatga tarqaladi chunki shakarni taqsimlaydi (1-iteratsiyada $r_{8,1} = 0$). Noaniq kontekstda uchraydigan atamalar parametrlari uchun konvergentsiya ko'proq vaqt oladi.

5.3-jadval. **EM klasterlash algoritmi**

(a)

docID	document text	docID	document text
1	hot chocolate cocoa beans	7	sweet sugar
2	cocoa ghana africa	8	sugar cane brazil
3	beans harvest ghana	9	sweet sugar beet
4	cocoa butter	10	sweet cake icing
5	butter truffles	11	cake black forest
6	sweet chocolate		

(b)

Parameter	Iteration of clustering							
	0	1	2	3	4	5	15	25
α_1		0.50	0.45	0.53	0.57	0.58	0.54	0.45
$r_{1,1}$		1.00	1.00	1.00	1.00	1.00	1.00	1.00
$r_{2,1}$		0.50	0.79	0.99	1.00	1.00	1.00	1.00
$r_{3,1}$		0.50	0.84	1.00	1.00	1.00	1.00	1.00
$r_{4,1}$		0.50	0.75	0.94	1.00	1.00	1.00	1.00
$r_{5,1}$		0.50	0.52	0.66	0.91	1.00	1.00	1.00
$r_{6,1}$	1.00	1.00	1.00	1.00	1.00	0.83	0.00	0.00
$r_{7,1}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$r_{8,1}$		0.00	0.00	0.00	0.00	0.00	0.00	0.00
$r_{9,1}$		0.00	0.00	0.00	0.00	0.00	0.00	0.00
$r_{10,1}$		0.50	0.40	0.14	0.01	0.00	0.00	0.00
$r_{11,1}$		0.50	0.57	0.58	0.41	0.07	0.00	0.00
$q_{africa,1}$		0.000	0.100	0.134	0.158	0.158	0.169	0.200
$q_{africa,2}$		0.000	0.083	0.042	0.001	0.000	0.000	0.000
$q_{brazil,1}$		0.000	0.000	0.000	0.000	0.000	0.000	0.000
$q_{brazil,2}$		0.000	0.167	0.195	0.213	0.214	0.196	0.167
$q_{cocoa,1}$		0.000	0.400	0.432	0.465	0.474	0.508	0.600
$q_{cocoa,2}$		0.000	0.167	0.090	0.014	0.001	0.000	0.000
$q_{sugar,1}$		0.000	0.000	0.000	0.000	0.000	0.000	0.000
$q_{sugar,2}$		1.000	0.500	0.585	0.640	0.642	0.589	0.500
$q_{sweet,1}$		1.000	0.300	0.238	0.180	0.159	0.153	0.000
$q_{sweet,2}$		1.000	0.417	0.507	0.610	0.640	0.608	0.667

Jadvalda hujjatlar to'plami (a) va EM klasterlash (b) paytida tanlangan iteratsiyalar uchun parametr qiymatlari ko'rsatilgan. Ko'rsatilgan parametrlar oldingi α_1, yumshoq tayinlash ballari $r_{n,1}$, 1 (ikkalasi 2-klaster uchun olib tashlangan) va bir necha atamalar uchun leksik parametrlar $q_{w,z}$. Mualliflar dastlab 6-hujjatni 1-klasterga va 7-hujjatni 2-klasterga (iteratsiya 0) tayinlagan. EM 25 iteratsiyadan keyin birlashadi. Tekislash uchun $r_{n,k}$ (5.16) tenglamadagi $r_{n,k} + \varepsilon$ bilan almashtirildi, bu yerda $\varepsilon = 0.0001$ ikkalasida ham yaqinlik mavjud. Natijada, atama 2-klaster bilan bir ma'noda bog'lanishi uchun 25 iteratsiya kerak bo'ladi. (25-iteratsiyada q yaqinlik, 1 = 0.) Yaxshi urg'ularni topish EM uchun K-o'rtachalarga qaraganda ancha muhimroqdir. Agar urg'ular yaxshi tanlanmagan bo'lsa EM mahalliy optimaga yopishib qolishga moyil bo'ladi. Bu EM ning boshqa ilovalarida ham uchraydigan umumiy muammo. Shuning uchun, K-vositalarida bo'lgani kabi, hujjatlarni klasterlarga dastlabki tayinlash ko'pincha boshqa algoritm bilan hisoblanadi. Misol uchun, qattiq K-vositalarni klasterlash dastlabki topshiriqni ta'minlashi mumkin, EM keyinchalik "yumshatilishi" mumkin.

Misol. Uuqorida K- o'rtachalarning vaqt murakkabligi $\mathbf{D(IK_{NM})}$ ekanligini ko'rdik. EM ning vaqt murakkabligi qanday bo'ladi?

Misol. Ω sinf tuzilmasini $\mathbf{C}$ ni aniq takrorlaydigan klasterlash va Ω dagi Ba'zi klasterlarni yanada bo'linadigan klasterlash bo'lsin. $I(\Omega.; C) =$ $I(\Omega.'; C)$ ekanligini ko'rsating.

Misol. $I(\Omega; C) \leq [H(\Omega) + H(C)]/2$ ekanligini ko'rsating.

Misol. O'zaro ma'lumot simmetrik bo'lib, agar klasterlar va sinflarning rollari almashtirilsa, uning qiymati o'zgarmaydi: $I(\Omega; C) =$ $I(C; \Omega)$. Boshqa uchta baholash ko'rsatkichlaridan qaysi biri bu ma'noda simmetrikdir?

Misol. 5.7-rasmdagi ikkita klaster uchun **RSS** ni hisoblang.

Misol. (i) nuqtalar to'plamiga va uchta boshlang'ich markazga (nuqtalar to'plamining a'zosi bo'lishi shart emas) misol keltiring, ular uchun 3-o'lchov bo'sh klasterli klasterga yaqinlashadi. (ii) Bo'sh klasterli klasterlash RSSga nisbatan global optimal bo'lishi mumkinmi?

Reuters-21578-ni yuklab oling. 10-sinflar sotib olish, makkajo'xori, xom, daromad, don, foiz, pul-fx, kema, savdo va bug'doydan birida sodir bo'lmagan hujjatlarni tashlang. Ushbu 10 ta sinfdan ikkitasida mavjud

bo'lgan hujjatlarni tashlang. (i) Ushbu kichik to'plamning 10 ta klasterga bo'lgan K-o'rtacha klasterini hisoblang. WEKA (Witten and Frank 2005) va R (R Development Core Team 2005) kabi K-vositalarni amalga oshiradigan bir qator dasturiy paketlar mavjud. (ii) 10-sinf bo'yicha klasterlash uchun tozalik, normallashtirilgan o'zaro ma'lumot, F_1 va **RI** ni hisoblash. (iii) 10 ta sinf va 10 ta klaster uchun chalkashlik matritsasini tuzing (3.5-jadval). Noto'g'ri ijobiy va noto'g'ri salbiylarni keltirib chiqaradigan sinflarni aniqlang.

Misol. RSSmin(K) ning K da monoton kamayib borayotganini isbotlang.

Misol. Klasterdagi hujjatning kasr a'zoligini uning markazdan Δ masofaning monoton kamayuvchi funksiyasi sifatida hisoblaydigan K-vositalarning yumshoq versiyasi mavjud, masalan, $e^{-\Delta}$ kabi. Ushbu yumshoq versiya uchun qattiq K-vositalarni qayta tayinlash va qayta hisoblash bosqichlarini o'zgartiring.

Misol. 5.3-jadvaldagi oxirgi takrorlashda 6-hujjat 1-klaster uchun boshlang'ich urg'u bo'lsa ham 2-klasterda. Nima uchun hujjat a'zolikni o'zgartiradi?

Miosl. 5.3-jadvaldagi 25-iteratsiyadagi $\mathbf{q_{mk}}$ parametrlarining qiymatlari yaxlitlangan. EM qanday qiymatlarga yaqinlashadi?

Misol. 5.3-jadvaldagi hujjatlar uchun K-vositalarni klasterlashni bajaring. Necha iteratsiyadan keyin K-vositalari yaqinlashadi? Natijani 5.3-jadvaldagi EM klasterlash bilan solishtiring va farqlarni muhokama qiling.

Misol. Gauss aralashmasi uchun EM ni kutish va maksimallashtirish bosqichlarini o'zgartiring. Maksimallashtirish bosqichi har bir klaster uchun $\alpha_k, \vec{\mu}_k$, va $\sum_k$ ning maksimal ehtimollik parametrlarini hisoblab chiqadi. Kutish bosqichi har bir vektor uchun joriy parametrlari asosida klasterlarga (Gausslar) yumshoq tayinlashni hisoblab chiqadi. (5.16) va (5.17) tenglamalarga mos keladigan Gauss aralashmalari tenglamalarini yozing.

Misol. Agar dispersiya juda kichik va barcha kovariatsiyalar 0 bo'lsa, K-o'rtachalarni Gauss aralashmalari uchun EM ning cheklovchi holati sifatida ko'rish mumkinligini ko'rsating.

Misol. Klasterlashning nuqta ichidagi tarqalishi $\sum_k \frac{1}{2}\sum_{\vec{x}_i \in \omega_k}\sum_{\vec{x}_j \in \omega_k} |\vec{x}_i - \vec{x}_j|^2$ sifatida aniqlanadi. RSS-ni minimallashtirish va nuqta ichidagi tarqalishni minimallashtirish ekvivalentdir.

Misol. (5.12) tenglamadan ko'p o'zgaruvchan Bernulli aralashmasi modeli uchun AIC kriteriyasini chiqaring.

5- bob bo'yicha foydalanilgan adabiyotlar

Williams, Hugh E., and Justin Zobel.
2005.
Searchable words on the web.
International Journal on Digital Libraries 5 (2): 99-105.
DOI: *dx.doi.org/10.1007/s00799-003-0050-z.*

Williams, Hugh E., Justin Zobel, and Dirk Bahle.
2004.
Fast phrase querying with combined indexes.
TOIS 22 (4): 573-594.

Witten, Ian H., and Timothy C. Bell.
1990.
Source models for natural language text.
International Journal Man-Machine Studies 32 (5): 545-579.

Witten, Ian H., and Eibe Frank.
2005.
Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition.
Morgan Kaufmann.

Witten, Ian H., Alistair Moffat, and Timothy C. Bell.
1999.
Managing Gigabytes: Compressing and Indexing Documents and Images, 2nd edition.
Morgan Kaufmann.

Wong, S. K. Michael, Yiyu Yao, and Peter Bollmann.
1988.
Linear structure in information retrieval.
In *Proc. SIGIR*, pp. 219-232. ACM Press.

Woodley, Alan, and Shlomo Geva.
2006.
NLPX at INEX 2006.
In *Proc. INEX*, pp. 302-311.
Witten, Ian H., Alistair Moffat, and Timothy C. Bell.
1999.
Managing Gigabytes: Compressing and Indexing Documents and Images, 2nd edition.
Morgan Kaufmann.
Wong, S. K. Michael, Yiyu Yao, and Peter Bollmann.
1988.
Linear structure in information retrieval.
In *Proc. SIGIR*, pp. 219-232. ACM Press.

5- bob bo'yicha nazariy va amaliy test savollari

1. Klasterli indeks haqida qaysi biri to'g'ri?
   A) Klasterli indeks sukut bo'yicha yagona kalit ustunlarida qurilgan
   B) Klasterli indeks jadval bilan bog'lanmagan
   C) Klasterli indeks noyob kalit ustunlar ustiga qurilgan emas
   D) To'g'ri javob yo'q
2. Indekslar haqida nimasi to'g'ri?
   A) Katta kalitlarga ega bo'lgan indekslar ustida ishlash uchun sql server dvigatellari ishlashini qiyinlashtiradi
   B) Jadval tez-tez yangilanib tursa ham indekslar unumdorligini oshiradi
   C) SQL server dvigatellari katta kalitlarga ega bo'lgan indekslarda ishlashini qiyinlashtirmaydi.
   D) To'g'ri javob yo'q
3. Indeks diskda joy egallaydimi?
   A) Ha, indekslar diskda saqlanadi
   B) Xotirani kerak bo'lganda saqlaydi
   C) Indekslar hech qachon diskda saqlanmaydi
   D) To'g'ri javob yo'q
4. Kompozit indekslar nima?

   A) Kompozit indeks 2 yoki undan ortiq ustunlardagi indekslarning birikmasidir
   B) Ichki foydalanish uchun ma'lumotlar bazasi tomonidan tuzilganlar
   C) Kompozit indeksni hech qachon yaratib bo'lmaydi
   D) To'g'ri javob yo'q
5. 2 yoki undan ortiq ustunlardagi indekslarning birikmasi nima deb nomlanadi?
   A) Kompozit indekslar
   B) Klasterli indeks
   C) klasterli bo'lmagan indeks
   D) To'g'ri javob yo'q
6. Agar _____ indeks bo'lsa, metama'lumotlar va statistika mavjud bo'lib qoladi.
   A) Ishdan chiqsa
   B) Tushib qolsa
   C) O'zgarsa
   D) A va B
7. Ma'lumotlar bazasi indeksi - bu nima?
   A) Qo'shimcha yozish xarajatlari evaziga ma'lumotlar bazasi jadvalidagi ma'lumotlarni qidirish operatsiyalari tezligini yaxshilaydigan ma'lumotlar tuzilmasi.
   B) Bu so'rov natijalari to'plamidagi barcha ustunlar klasterli bo'lmagan indekslardan olinadigan so'rov
   C) yozuv uchun barcha ustunlarni birga saqlash o'rniga, har bir ustun indeksdagi boshqa barcha qatorlar bilan alohida saqlanadi
   D) B javob to'g'ri
8. _____ indeksida yozuv uchun barcha ustunlarni birga saqlash o'rniga, har bir ustun indeksdagi boshqa barcha qatorlar bilan alohida saqlanadi.
   A) Ustunli
   B) Klasterlangan
   C) Klaster bo'lmagan
   D) Qatorli
9. _____ indeks - bu so'rovda so'ralgan barcha ustunlarni klasterlangan indeksni qo'shimcha izlamasdan qondiradigan ko'rsatkich.
   A) Qoplash
   B) Klasterlangan

C) Klasterli emas

D) B javob to'g'ri

10. Qoplangan so'rov – nima?

A) Bu so'rov natijalari to'plamidagi barcha ustunlar klasterli bo'lmagan indekslardan olinadigan so'rov

B) yozuv uchun barcha ustunlarni birga saqlash o'rniga, har bir ustun indeksdagi boshqa barcha qatorlar bilan alohida saqlanadi

C) Qo'shimcha yozish xarajatlari evaziga ma'lumotlar bazasi jadvalidagi ma'lumotlarni qidirish operatsiyalari tezligini yaxshilaydigan ma'lumotlar tuzilmasi

D) Ichki foydalanish uchun ma'lumotlar bazasi tomonidan tuziladi

VI BOB. IERARXIK KLASTERLASH

Yassi klasterlash samarali va kontseptual jihatdan sodda, ammo ular 5-bobda ko'rganimizdek uning bir qator kamchiliklari bor. 5-bobda keltirilgan algoritmlar bir tekis tuzilmagan klasterlar to'plamini qaytaradi, kirish sifatida oldindan belgilangan klasterlar sonini talab qiladi va aniqlanmaydi. Ierarxik klasterlash yassi klasterlash orqali qaytarilgan tuzilmagan klasterlar to'plamidan ko'ra ko'proq ma'lumotga ega bo'lgan ierarxiyani chiqaradi.

Ierarxik klasterlash ulardan klasterlar sonini oldindan belgilashni talab qilmaydi va AQda qo'llaniladigan ierarxik algoritmlarning aksariyati deterministikdir. Ierarxik klasterlashning bunday afzalliklari samaradorlikning pastligi bilan bog'liq. Eng keng tarqalgan ierarxik klasterlash algoritmlari ***K-vosita*** va **EM** ning chiziqli murakkabligi bilan solishtirganda hujjatlar sonida kamida kvadratik bo'lgan murakkablikka ega (5.4-bo'lim). Ushbu bob birinchi navbatda ***aglomerativ ierarxik*** klasterlash bilan tanishadi (6.1-bo'lim) va 6.2, 6.4-bo'limlarda to'rt xil aglomerativ algoritmni taqdim etadi, ular o'zlari qo'llaydigan o'xshashlik ko'rsatkichlari bilan farqlanadi: bir-bo'g'in, to'liq havola, guruh-o'rtacha va markaziy o'xshashlik. Keyin 6.5-bo'limda ierarxik klasterlashning optimallik shartlari muhokama qilinadi. 6.6-bo'lim yuqoridan pastga ierarxik klasterlashni joriy qilinadi. 6.7-bo'limda klasterlarni avtomatik ravishda etiketlash ko'rib chiqiladi, bu muammo odamlar klasterlash natijasi bilan o'zaro aloqada bo'lganda hal qilinishi kerak. Ular 6.8-bo'limda amalga oshirish masalalari muhokama qilinadi. 6.9-bo'lim keyingi o'qish uchun ko'rsatmalar beradi, shu jumladan yumshoq ierarxik klasterlarga havolalar yoritiladi. Axborotni qidirishda tekis va ierarxik klasterlashning qo'llanilishi o'rtasida bir nechta farqlar mavjud. Xususan, ierarxik klasterlash 5.1-jadvalda ko'rsatilgan har qanday ilovalar uchun mos keladi (5.6-bo'lim). Aslida, yig'ish klasteri uchun berilgan misol ierarxikdir. Umuman olganda, yassi klasterlashni samaradorlik muhim bo'lganda va ierarxik klasterni yassi klasterlashning mumkin bo'lgan muammolaridan biri (yetarli tuzilma, oldindan belgilangan klasterlar soni, determinizm) tashvishlantirganda tanlanadi. Bundan tashqari, ko'plab tadqiqotchilar ierarxik klasterlash tekis klasterlardan ko'ra yaxshiroq klasterlarni hosil qiladi deb hisoblashadi. Biroq, bu masala bo'yicha konsepsiya yo'q (6.9-bo'limdagi havolalar).

1.Ushbu bobda faqat 6.1-rasmda ko'rsatilgandek ikkilik daraxtlar bo'lgan ierarxiyalar ko'rib chiqiladi ammo ierarxik klasterlash boshqa turdagi daraxtlarga osonlik bilan kengaytirilishi mumkin.

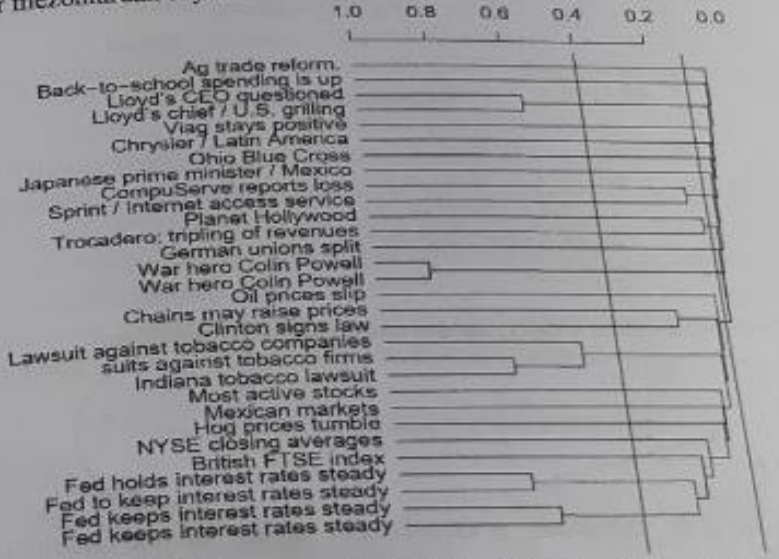
6.1. Ierarxik aglomerativ klasterlash

Ierarxik klasterlash algoritmlari yuqoridan pastga yoki pastdan yuqoriga hisoblash mumkin. Pastdan yuqoriga ko'tarish algoritmlari har bir hujjatni boshida yagona klaster sifatida ko'rib chiqadi va keyin barcha klasterlar barcha hujjatlarni o'z ichiga olgan yagona klasterga birlashtirilgunga qadar klaster juftlarini ketma-ket birlashtiradi (yoki aglomeratsiya qiladi). Shuning uchun pastdan yuqoriga ierarxik klasterlash ierarxik ***aglomerativ klasterlash*** yoki ***HAC*** deb ataladi. Yuqoridan pastga klasterlash klasterni bo'lish usulini talab qiladi. U alohida hujjatlarga erishilgunga qadar klasterlarni rekursiv bo'lish orqali davom etadi. HAC yuqoridan pastga klasterlashdan ko'ra AQda tez-tez qo'llaniladi va bu bobning asosiy mavzusidir. 8.2-17.4 bo'limlarda OAKda qo'llaniladigan o'xshashlik o'lchovlarini ko'rib chiqishdan oldin, birinchi navbatda ierarxik klasterlarni grafik tarzda tasvirlash usuli kiritiladi. OAK larning bir nechta asosiy xususiyatlarini muhokama qilinadi va OAKni hisoblash uchun oddiy algoritmni taqdim etiladi.

OAK klasteri odatda 6.1-rasmda ko'rsatilganidek, ***dendrogramma*** sifatida tasvirlanadi. Har bir birlashma gorizontal chiziq bilan ifodalanadi. Gorizontal chiziqning y-koordinatasi - bu birlashtirilgan ikkita klasterning o'xshashligi, bu yerda hujjatlar yagona klaster sifatida ko'rib chiqiladi. Bu o'xshashlik birlashtirilgan klasterning kombinatsiya o'xshashligi deb ataladi. Misol uchun, 6.1-rasmda so'ralgan ***Lloyd's*** bosh direktori va Lloyd's boshlig'i / AQSh grillidan iborat klasterning kombinatsiyasi o'xshashligi $\approx$ 0,56 ni tashkil qiladi. ***Singleton*** klasterining kombinatsiya o'xshashligini hujjatning o'ziga o'xshashligi sifatida aniqlaymiz (kosinus o'xshashligi uchun 1,0). Pastki qatlamdan yuqori tugunga o'tish orqali ***dendrogramma*** tasvirlangan klasterga olib keladigan birlashishlar tarixini qayta tiklashga imkon beradi. Misol uchun, urush qahramoni Kolin Pauell nomli ikkita hujjat 6.1-rasmda birinchi bo'lib birlashtirilganligini va oxirgi birlashma Ag savdo islohotini boshqa 29 hujjatdan iborat klasterga qo'shganini ko'ramiz. OAKda asosiy taxmin shundan iboratki, birlashma jarayoni

monotonikdir. ***Monotonik*** degani, agar $s_1, s_2, \ldots, s_{K-1}$ - HAC ning ketma-ket birlashish kombinatsiyasi o'xshashliklari, keyin $s_1 \geq s_2 \geq \ldots \geq s_{K-1}$ ushlab turadi. ***INVERSION*** bo'lmagan ***monotonik ierarxik klasterlash*** kamida bitta inversiya $s_i < s_{i+1}$ ni o'z ichiga oladi va har bir bosqichda mavjud bo'lgan eng yaxshi birlashmani tanlaganimiz haqidagi asosiy taxminga zid keladi. Inversiya misolini 6.12-rasmda ko'ramiz.

Ierarxik klasterlash oldindan belgilangan klasterlar sonini talab qilmaydi. Biroq, ba'zi ilovalarda tekis klasterlashda bo'lgani kabi, ajratilgan klasterlarning bo'linishini xohlaymiz. Bunday hollarda, ierarxiya bir nuqtada kesilishi kerak. Kesish nuqtasini aniqlash uchun bir qator mezonlardan foydalanish mumkin:



• Oldindan belgilangan o'xshashlik darajasida kesish. Misol uchun, agar minimal kombinatsiya o'xshashligi 0,4 bo'lgan klasterlarni xohlasak, ***dendrogrammani*** 0,4 ga kesib tashlaymiz. 6.1-rasmda diagrammani y = 0,4 da kesish natijasida 24 ta klaster (faqat o'xshashligi yuqori bo'lgan hujjatlarni birgalikda guruhlash) va y = 0,1 da kesish natijasida 12 ta klaster (bir yirik moliyaviy yangiliklar klasteri va 11 ta kichikroq klaster) hosil bo'ladi.

• Ikki ketma-ket kombinatsiya o'xshashligi orasidagi bo'shliq eng katta bo'lgan ***dendrogrammani*** kesib oling. Bunday katta bo'shliqlar, shubhasiz, "tabiiy" klasterlarni ko'rsatadi. Yana bitta klaster qo'shilishi

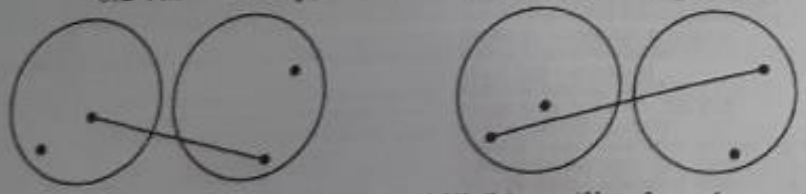
klasterlash sifatini sezilarli darajada pasaytiradi, shuning uchun keskin pasayish sodir bo'lgunga qadar kesish maqsadga muvofiqdir. Ushbu strategiya 5.8-rasmda ***K-o'rtacha grafigi***da tizzani qidirishga o'xshaydi.

- *(5.11)* tenglamani qo'llash:

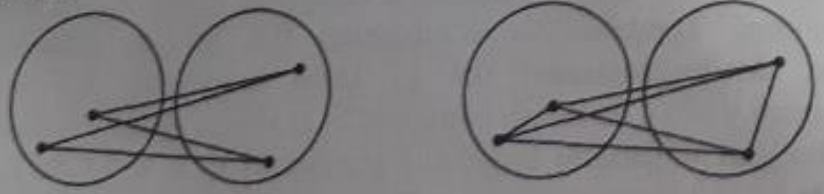
$$K = \arg\min_{K'} [RSS(K') + \lambda K']$$

```
SIMPLEHAC(d_1, ..., d_N)
 1  for n ← 1 to N
 2  do for i ← 1 to N
 3     do C[n][i] ← SIM(d_n, d_i)
 4     I[n] ← 1 (keeps track of active clusters)
 5  A ← [] (assembles clustering as a sequence of merges)
 6  for k ← 1 to N − 1
 7  do (i, m) ← arg max_{(i,m): i≠m ∧ I[i]=1 ∧ I[m]=1} C[i][m]
 8     A.APPEND((i, m)) (store merge)
 9     for j ← 1 to N
10     do C[i][j] ← SIM(i, m, j)
11        C[j][i] ← SIM(i, m, j)
12     I[m] ← 0 (deactivate cluster)
13  return A
```

6.2-rasm. **Oddiy, ammo samarali HAC algoritmi**



(a) bitta bo'g'in: maksimal o'xshashlik(b) to'liq havola: minimal o'xshashlik

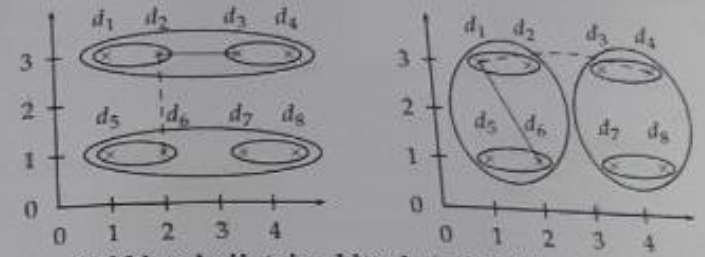


(c) centroid: o'rtacha o'xshashlik

(d) o'rtacha guruh: barcha o'xshashliklarning o'rtachasi

6.3-rasm. **To'rtta HAC algoritmlari tomonidan qo'llaniladigan klaster o'xshashligining turli tushunchalari**

O'zaro o'xshashlik - bu turli klasterlardagi ikkita hujjat o'rtasidagi o'xshashlik.



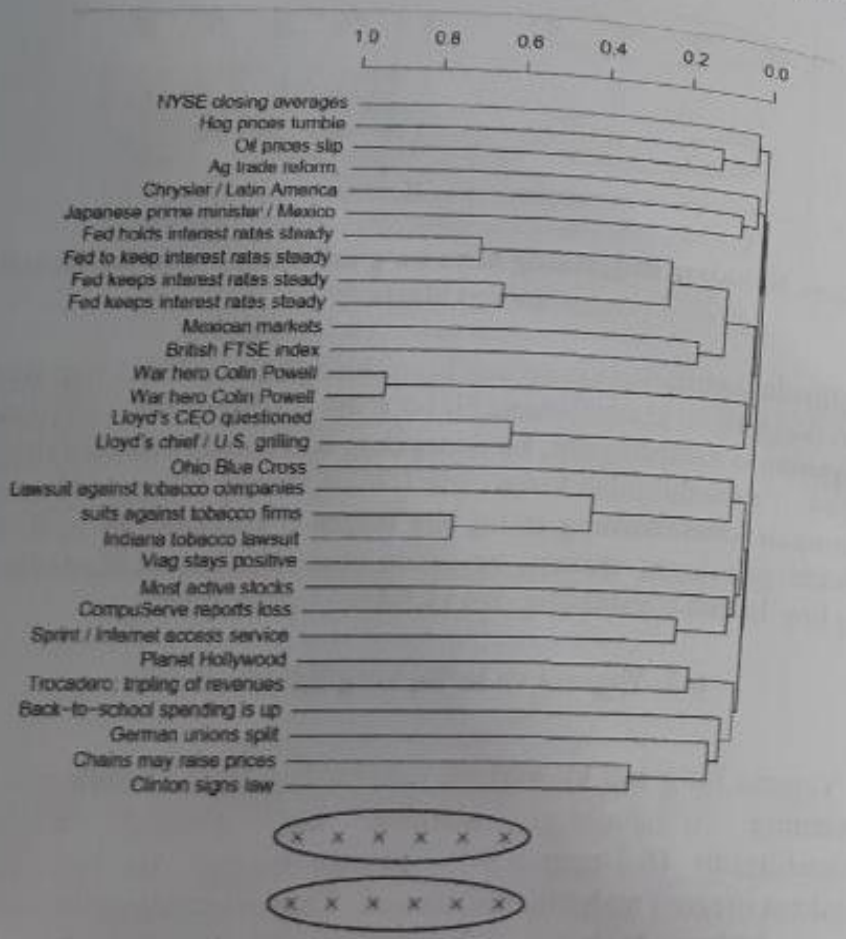
6.4-rasm. **Sakkizta hujjatning bitta bo'g'inli (chap) va to'liq havolali (o'ng) klasteri**

Ellipslar ketma-ket klasterlash bosqichlariga mos keladi. Chap: ikkita yuqori ikki nuqtali klasterlarning bir bo'g'inli o'xshashligi $\mathbf{d_2}$ va $\mathbf{d_3}$ (qattiq chiziq) ning o'xshashligidir, bu ikkita chap ikki nuqtali klasterlarning bir bo'g'inli o'xshashligidan kattaroqdir (chiziq chiziq). O'ng: ikkita yuqori ikki nuqtali klasterlarning to'liq bog'langan o'xshashligi $\mathbf{d_1}$ va $\mathbf{d_4}$ ning o'xshashligi (chiziq chiziq), bu ikkita chap ikki nuqtali klasterlarning to'liq bog'lanish o'xshashligidan kichikroq (qattiq chiziq).

6.2. Yagona va to'liq bo'g'inli klasterlash

Yagona bo'g'inli klasterlash yoki bir bog'lamli klasterlashda ikkita klasterning o'xshashligi ularning eng o'xshash a'zolarining o'xshashligidir (6.3-rasm, $^{(a)}$- rasmga qarang). Bu bitta bo'g'inli birlashma mezoni mahalliy hisoblanadi. Faqat ikkita klaster bir-biriga eng yaqin joylashgan hududga e'tibor beramiz. Klasterning boshqa, uzoqroq qismlari va klasterlarning umumiy tuzilishi hisobga olinmaydi. To'liq bo'g'inli klaster yoki to'liq bog'langan klasterlashda ikkita klasterning o'xshashligi ularning eng o'xshash a'zolarining o'xshashligidir (6.3-rasm, (b)-rasm). Bu birlashuvi eng kichik diametrga ega bo'lgan klaster juftligini tanlashga teng. Ushbu to'liq havolani birlashtirish mezoni mahalliy emas, klasterlashning butun tuzilishi birlashish qarorlariga ta'sir qilishi mumkin. Bu uzun, zerikarli klasterlarga nisbatan kichik diametrli

ixcham klasterlarni afzal ko'rishga olib keladi lekin ayni paytda tashqi ko'rsatkichlarga nisbatan sezgirlikni keltirib chiqaradi. Markazdan uzoqda joylashgan bitta hujjat nomzodlarni birlashtirish klasterlarining diametrini keskin oshirishi va yakuniy klasterni butunlay o'zgartirishi mumkin.



6.6-rasm. **Bir bo'g'inli klasterlashda zanjirband qilish**

Yagona bo'g'inli klasterlashda mahalliy mezon kiruvchi cho'zilgan klasterlarni keltirib chiqarishi mumkin.

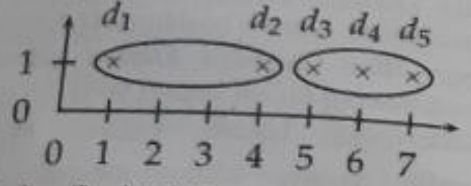
6.4-rasmda sakkizta hujjatning bitta bo'g'inli va to'liq bo'g'inli klasteri tasvirlangan. Birinchi to'rtta qadam, har biri ikkita hujjat juftligidan iborat klasterni ishlab chiqaradi, ya'ni bir xil. Keyin bitta

bo'g'inli klaster yuqori ikkita juftlikni (undan keyin pastki ikkita juftlikni) birlashtiradi chunki klaster o'xshashligining maksimal o'xshashlik ta'rifiga ko'ra, bu ikki klaster eng yaqin. To'liq havola klasteri chap ikkita juftlikni (keyin o'ngdagi ikkita juftlikni) birlashtiradi chunki ular klaster o'xshashligining minimal o'xshashlik ta'rifiga ko'ra eng yaqin juftliklardir.

6.1-rasmda hujjatlar to'plamining bir bo'g'inli klasterlanishiga misol va 6.5-rasmda bir xil to'plamning to'liq bo'g'inli klasterlanishi ko'rsatilgan. 6.5-rasmdagi so'nggi birlashmani kesishda o'xshash o'lchamdagi ikkita klasterni olamiz (1-16 hujjatlar, ***NYSE*** yopilish o'rtacha ko'rsatkichlaridan ***Lloyd's*** boshlig'i/***AQSh*** grilliga qadar va 17-30 hujjatlar, ***Ogayo Blue Crossdan Klinton*** qonun imzolaydi). 6.1-rasmda ***dendrogrammaning*** bir xil kesimi yo'q, bu ularga teng darajada ***muvozanatli klasterlash***ni beradi. ***Yagona va to'liq bog'lamli klasterlash***da grafik nazariy talqini mavjud. s_k ni k qadamda birlashtirilgan ikkita klasterning kombinatsiyasi o'xshashligini va $G(s_k)$ barcha ma'lumotlar nuqtalarini kam o'xshashlik bilan bog'laydigan grafikni aniqlang. So'ngra bir bog'lamli klasterlashda bosqichdan keyingi klasterlar: $G(s_k)$ ning bog'langan komponentlari va to'liq bog'lanishli klasterlashda bosqichdan keying klasterlar $G(s_k)$ ning maksimal klasterlaridir. Bog'langan komponent - bu har bir juftlikni bog'laydigan yo'l bo'lishi uchun ulangan nuqtalarning maksimal to'plami. ***Klik*** - bu bir-biri bilan to'liq bog'langan nuqtalar to'plami. Ushbu grafik-nazariy talqinlar bitta va to'liq havolali klasterlash atamalarini rag'batlantiradi. **k** qadamdagi bir bo'g'inli klasterlar $s \geq s_k$ o'xshashlikdagi kamida bitta bo'g'in (bitta havola) orqali bog'langan nuqtalarning maksimal to'plamidir; **k** qadamdagi to'liq bog'langan klasterlar $s \geq s_k$ o'xshashlikdagi zvenolar orqali bir-biri bilan to'liq bog'langan nuqtalarning maksimal to'plamidir. Yagona bo'g'inli va to'liq bo'g'inli klasterlash klaster sifatini baholashni bir juft hujjatlar o'rtasidagi yagona o'xshashlikka tushiradi: bitta bo'g'inli klasterlashda ikkita eng o'xshash hujjat va to'liq havolali klasterlashda ikkita eng o'xshash bo'lmagan hujjat. Bir juftlikka asoslangan o'lchov klasterdagi hujjatlarning taqsimlanishini to'liq aks ettira olmaydi. Shuning uchun ikkala algoritm ham ko'pincha istalmagan klasterlarni keltirib chiqarishi ajablanarli emas.

Yagona bo'g'inli klasterlash 6.6-rasmda ko'rsatilganidek, straggling klasterlarni hosil qilishi mumkin. Birlashtirish mezoni qat'iy mahalliy bo'lganligi sababli, nuqtalar zanjiri paydo bo'lgan klasterning umumiy shaklini hisobga olmagan holda uzoq masofalarga uzaytirilishi mumkin. Ushbu ta'sir **zanjirli** deb ataladi.

Agar bog'lanish ehtimoli sizni bezovta qilayotgan bo'lsa, d_i ning koordinatalari $(1+\varepsilon, 3-\varepsilon)$ va qolgan barcha nuqtalar butun son koordinatalariga ega deb hisoblang.



6.7-rasm. **To'liq bog'lanishli klasterlashdagi o'zgarishlar**

Beshta hujjat x-koordinatalariga ega: $1 + 2\varrho$, 4, $5 + 2\varrho$, 6 va $7 - \varrho$. To'liq havola klasteri ellips shaklida ko'rsatilgan ikkita $1+2\varepsilon, 4, 5+2\varepsilon, 6$ klasterni yaratadi. Eng intuitiv ikki klasterli $\{\{d_1\}, \{d_2, d_3, d_4, d_5\}\}$ klasterlash, lekin to'liq bog'lanishli klasterlashda chetdagi ko'rsatkich d $\{d_2, d_3, d_4, d_5\}$ ko'rsatilganidek.

Zanjirlash effekti 6.1-rasmda ham ko'rinadi. Yagona bo'g'inli klasterlashning oxirgi o'n bir birlashmasi (0,1 qatordan yuqori bo'lganlar) zanjirga mos keladigan bitta hujjatlar yoki juftlik hujjatlariga qo'shiladi. 6.5-rasmdagi to'liq havola klasteri bu muammoni bartaraf qiladi. Hujjatlar oxirgi birlashmada ***dendrogrammani*** kesib tashlaganimizda taxminan bir xil o'lchamdagi ikkita guruhga bo'linadi. Umuman olganda, bu zanjirlar bilan klasterlashdan ko'ra ma'lumotlarning foydaliroq tashkilotidir. Biroq, to'liq havola klasteri boshqa muammoga duch keladi. U klasterning global tuzilishiga mos kelmaydigan ko'rsatkichlarga, nuqtalarga juda ko'p e'tibor beradi. 6.7-rasmdagi misolda to'rtta d_2, d_3, d_4, d_5 chap chetidagi d_1 chegarasi tufayli bo'lingan. To'liq havolali klasterlash ushbu misolda eng intuitiv klaster tuzilishini topa olmaydi.

6.2.1. Ierarxik klasterlashning vaqt murakkabligi

6.2-rasmdagi sodda HAC algoritmining murakkabligi $\Theta(N^3)$ chunki ular **N×N** matritsasi **CxC** ni har bir **N - 1** iteratsiyada eng katta o'xshashlik uchun to'liq skanerlaydi. Ushbu bobda muhokama qilingan to'rtta ***HAC*** usullari uchun 6.8-rasmda ko'rsatilgan ustuvor navbat algoritmi yanada samaraliroq algoritmdir. Uning vaqt murakkabligi $\Theta(N^2 \log N)$ o'xshashlik matritsasi **C** ning **C[k]** satrlari ustuvorlik navbatlaridagi o'xshashlikning kamayishi tartibida tartiblanadi. **P[k].MAX()** so'ngra hozirda eng yuqori o'xshashlikka ega bo'lgan **P[k]** klasterini qaytaradi. 5-bobdagi kabi k^{th} klasterni belgilash uchun w_{k_i} dan foydalaniladi. w_{k_2}, w_{k_1} ning birlashtirilgan klasterini yaratgandan so'ng, uning vakili sifatida w_{k_1} ishlatiladi. SIM funksiyasi potentsial birlashma juftliklari uchun o'xshashlik funksiyasini hisoblab chiqadi: bitta havola uchun eng katta o'xshashlik, to'liq havola uchun eng kichik o'xshashlik, *GAAC* uchun o'rtacha o'xshashlik (6.3-bo'lim) va markazlashgan klasterlash uchun markaziy o'xshashlik (6.4-bo'lim)

```
EFFICIENTHAC(d⃗1, ..., d⃗N)
 1  for n ← 1 to N
 2  do for i ← 1 to N
 3     do C[n][i].sim ← d⃗n · d⃗i
 4        C[n][i].index ← i
 5     I[n] ← 1
 6     P[n] ← priority queue for C[n] sorted on sim
 7     P[n].DELETE(C[n][n]) (don't want self-similarities)
 8  A ← []
 9  for k ← 1 to N − 1
10  do k1 ← arg max_{k:I[k]=1} P[k].MAX().sim
11     k2 ← P[k1].MAX().index
12     A.APPEND(⟨k1, k2⟩)
13     I[k2] ← 0
14     P[k1] ← []
15     for each i with I[i] = 1 ∧ i ≠ k1
16     do P[i].DELETE(C[i][k1])
17        P[i].DELETE(C[i][k2])
18        C[i][k1].sim ← SIM(i, k1, k2)
19        P[i].INSERT(C[i][k1])
20        C[k1][i].sim ← SIM(i, k1, k2)
21        P[k1].INSERT(C[k1][i])
22  return A
```

clustering algorithm	$SIM(i, k_1, k_2)$
single-link	$\max(SIM(i,k_1), SIM(i,k_2))$
complete-link	$\min(SIM(i,k_1), SIM(i,k_2))$
centroid	$(\frac{1}{N_m}\vec{v}_m) \cdot (\frac{1}{N_i}\vec{v}_i)$
group-average	$\frac{1}{(N_m+N_i)(N_m+N_i-1)}[(\vec{v}_m+\vec{v}_i)^2-(N_m+N_i)]$

compute $C[5]$

1	2	3	4	5
0.2	0.8	0.6	0.4	1.0

create $P[5]$ (by sorting)

2	3	4	1
0.8	0.6	0.4	0.2

merge 2 and 3, update similarity of 2, delete 3

2	4	1
0.3	0.4	0.2

delete and reinsert 2

4	2	1
0.4	0.3	0.2

6.8-rasm. **OAK uchun ustuvorlik-navbat algoritmi**

Yuqori: Algoritm. Markaz: to'rt xil o'xshashlik o'lchovi. Pastki: 6 va 16–19-bosqichlarni qayta ishlashga misol. Bu **5×5** matritsa **C** uchun **P[5]** ni ko'rsatadigan tuzilgan misoldir.

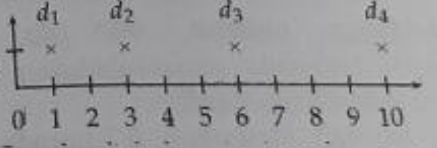
```
SINGLELINKCLUSTERING(d_1, ..., d_N)
 1  for n ← 1 to N
 2  do for i ← 1 to N
 3     do C[n][i].sim ← SIM(d_n, d_i)
 4        C[n][i].index ← i
 5     I[n] ← n
 6     NBM[n] ← arg max_{X∈{C[n][i]:n≠i}} X.sim
 7  A ← []
 8  for n ← 1 to N − 1
 9  do i_1 ← arg max_{i:I[i]=i} NBM[i].sim
10     i_2 ← I[NBM[i_1].index]
11     A.APPEND(⟨i_1, i_2⟩)
12     for i ← 1 to N
13     do if I[i] = i ∧ i ≠ i_1 ∧ i ≠ i_2
14           then C[i_1][i].sim ← C[i][i_1].sim ← max(C[i_1][i].sim, C[i_2][i].sim)
15        if I[i] = i_2
16           then I[i] ← i_1
17     NBM[i_1] ← arg max_{X∈{C[i_1][i]:I[i]=i∧i≠i_1}} X.sim
18  return A
```

6.9-rasm. **NBM massividan foydalangan holda yagona bo'g'inli klasterlash algoritmi**

Ikkita i_1 va i_2 klasterlarini birlashtirgandan so'ng, birinchisi (i_1) birlashtirilgan klasterni ifodalaydi. Agar I[i] = i bo'lsa, u holda **i** uni joriy klasterining vakili hisoblanadi. Agar **I[i]!=I** bo'lsa, u holda **I[i]** bilan ifodalangan klasterga birlashtirilgan va shuning uchun $NBM[i_1]$ ni

yangilashda e'tiborga olinmaydi. **C** qatori qanday qayta ishlanishiga misol keltiramiz (6.8-rasm, pastki panel). 1–7 qatorlardagi halqa $\Theta(N^2)$ va 9–21-qatorlardagi halqa $\Theta(N^2 \log N)$ ga o'chirish va kiritishni qo'llab-quvvatlaydigan ustuvor navbatlarni amalga oshirish uchun $\Theta(\log N)$. Algoritmning umumiy murakkabligi uchun $\Theta(N^2 \log N)$.

SIM funksiyasini aniqlashda $\vec{v}_m$ va $\vec{v}_i$ mos ravishda $w_{k_1} \cup w_{k_2}$ va w_i ning vektor yig'indilari, N_m va N_i esa hujjatlar soni $w_{k_1} \cup w_{k_2}$ va w_i mos ravishda. 6.8-rasmdagi ***EFFICIENTHAC*** argumenti vektorlar to'plamidir (umumiy hujjatlar to'plamidan farqli o'laroq) chunki ***GAAC*** va ***markazlashtirilgan klasterlash*** (6.3 va 6.4-bo'limlar) kirish sifatida vektorlarni talab qiladi. ***EFFICIENTHAC*** ning to'liq havola versiyasi vektor sifatida ko'rsatilmagan hujjatlarga ham qo'llanilishi mumkin. Yagona havola uchun 6.9-rasmda ko'rsatilganidek, keyingi optimallashtirish sifatida keyingi eng yaxshi birlashma massivini (NBM) kiritishimiz mumkin. NBM har bir klaster uchun eng yaxshi birlashma nima ekanligini kuzatib boradi. 6.9-rasmdagi ikkita yuqori darajadagi for-sikllarining har biri $\Theta(N^2)$ shuning uchun bitta bo'g'inli klasterlashning umumiy murakkabligi $\Theta(N^2)$ ga teng.



6.10-rasm. **To'liq havolali klasterlash**

To'liq havolali klasterlash eng yaxshi birlashma doimiy emas. Dastlab, **d2**, **d3** uchun eng yaxshi birlashma klasteridir. Lekin **d1** va **d2** ni birlashtirgandan so'ng, **d4** **d3** ning eng yaxshi birlashma nomzodiga aylanadi. Yagona havola kabi eng yaxshi birlashuvchi doimiy algoritmda **d3** ning eng yaxshi birlashma klasteri {**d1**, **d2**} bo'ladi.

Boshqa uchta ***HAC algoritmi***ni ***NBM massivi*** bilan tezlashtira olamizmi? Buni qila olmaymiz chunki faqat bitta bo'g'inli klasterlash eng yaxshi birlashma doimiydir. Faraz qilaylik, w_k uchun eng yaxshi birlashma klasteri bitta bo'g'inli klasterlashda w_j bo'lsin. Keyin w_j ni

uchinchi klaster $w_j \neq w_k$ bilan birlashtirgandan so'ng, w_k va w_j ning birlashishi $\bar{o}_k$ ning eng yaxshi w_k birlashma klasteri bo'ladi. Boshqacha qilib aytadigan bo'lsak, birlashtirilgan klaster uchun eng yaxshi birlashuvchi nomzod uning tarkibiy qismlarini yagona bog'langan klasterlashda eng yaxshi birlashuvchi ikkita nomzoddan biri hisoblanadi. Bu shuni anglatadiki, **C** har bir iteratsiyada $\Theta(N)$ da yangilanishi mumkin - qolgan $\leq N$ klasterlarning har biri uchun 6.9-rasmdagi 14-qatordagi ikkita qiymatdan iborat oddiy maksimal qiymatni olishi mumkin. 6.10-rasmda ko'rsatilgandek, eng yaxshi birlashma qat'iyligi to'liq bog'langan klasterlash uchun mos kelmaydi, ya'ni klasterlashni tezlashtirish uchun ***NBM massivi***dan foydalana olmaymiz. $\mathbf{d_3}$ ning eng yaxshi birlashma nomzodi $\mathbf{d_2}$ bilan $\mathbf{d_1}$ klasterini birlashtirgandan so'ng, o'zaro bog'liq bo'lmagan $\mathbf{d_4}$ klaster $\mathbf{d_3}$ uchun eng yaxshi birlashma nomzodiga aylanadi. Buning sababi, to'liq havolani birlashtirish mezoni mahalliy bo'lmagan va unga quyidagi nuqtalar ta'sir qilishi mumkin. Ikkita birlashuvchi nomzod uchrashadigan hududdan katta masofa. Amalda $\Theta(N^2 \log \bar{N})$ algoritmining samaradorligi bir bo'g'inli algoritmga nisbatan kichik bo'ladi chunki ikkita hujjat o'rtasidagi o'xshashlikni hisoblash kattalik tartibidir. Saralashda ikki ***skalyar solishtirish***dan sekinroq. Ushbu bobdagi barcha to'rtta ***HAC algoritmlari*** o'xshashlik hisoblariga nisbatan $\Theta(N^2)$ dir. Shunday qilib, algoritmlardan birini tanlashda murakkablikdagi farq kamdan-kam hollarda amaliyotda tashvish tug'diradi.

Misol. To'liq havolali klasterlash 6.7-rasmda tasvirlangan ikki klasterli klasterni yaratishini ko'rsating.

6.3. Guruhning o'rtacha aglomerativ klasterlashuvi

Guruh bo'yicha o'rtacha ***aglomerativ klasterlash*** yoki *GAAC* (6.3-rasm, (d)-rasmga qarang) klaster sifatini hujjatlar o'rtasidagi barcha o'xshashliklarga asoslanib baholaydi va shu bilan klaster o'xshashligi bilan klaster o'xshashligini tenglashtiradigan yagona va to'liq bo'g'in mezonlaridan qochadi. Bitta juft hujjatlar **GAAC** shuningdek, guruhning ***o'rtacha klasteri*** va **o'rtacha havolali klaster** deb ataladi. **GAAC** barcha hujjatlar juftlarining, shu jumladan ***bir klasterdagi juftlarning*** **SIM-GA**

o'rtacha o'xshashligini hisoblab chiqadi. Ammo o'ziga o'xshashliklar o'rtacha ko'rsatkichga kiritilmagan:

$$SIM-GA(w_i, w_j) = \frac{1}{(N_i+N_j)(N_i+N_j-1)} \sum_{d_m \in w_i \cup w_j} \sum_{d_n \in w_i \cup w_j, d_m \neq d_n} \vec{d}_m \cdot \vec{d}_n \quad (6.1)$$

bu yerda **d** hujjatning uzunlik-normalangan vektori, · nuqta mahsulotini bildiradi, $\mathbf{N_i}$ va $\mathbf{N_j}$ esa mos ravishda w_i va w_j dagi hujjatlar soni. **GAAC** uchun motivatsiya shundan iboratki, **HAC**da keyingi birlashma sifatida ikkita w_i va $\mathbf{w_j}$ klasterlarini tanlashdan maqsadimiz shundan iboratki, natijad $w_k = w_i \cup w_j$ birlashma klasteri izchil bo'lishi kerak. w_k ning kogerentligini baholash uchun $\mathbf{w_k}$ doirasidagi barcha hujjat hujjatidagi o'xshashliklarni, shujumladan w_i va w_j ichida sodir bo'ladiganlarni ko'rib chiqishimiz kerak. **SIM-GA** o'lchovini samarali hisoblashimiz mumkin chunki ***individual vektor*** o'xshashliklari yig'indisi ularni yig'indilarining o'xshashligiga teng:

$$\sum_{d_m \in w_i} \sum_{d_n \in w_j} (\vec{d}_m \cdot \vec{d}_n) = (\sum_{d_m \in w_i} \vec{d}_m) \cdot (\sum_{d_n \in w_j} \vec{d}_n) \quad (6.2)$$

(6.2) bilan quyidagiga ega bo'lamiz:

$$SIM-GA(w_i, w_j) = \frac{1}{(N_i+N_j)(N_i+N_j-1)} [(\sum_{d_m \in w_i \cup w_j} \vec{d}_m)^2 - (N_i+N_j)] \quad (6.3)$$

O'ngdagi atama $(\mathbf{N_i} + \mathbf{N_j})$ 1,0 qiymatdagi $\mathbf{N_i}+\mathbf{N_j}$ o'ziga o'xshashliklarining yig'indisidir. Ushbu hiyla yordamida doimiy vaqt ichida klaster o'xshashligini hisoblashimiz mumkin (agar ularda ikkita vektor yig'indisi mavjud bo'lsa $\sum_{d_m \in w_i} \vec{d}_m$ va $\sum_{d_m \in w_j} \vec{d}_m$ o'rniga $\Theta(N_i N_j)$. Bu juda muhim chunki **GAAC** ni samarali amalga oshirish uchun doimiy vaqt ichida *EFFICIENTHAC* (6.8-rasm) 18 va 20-qatorlardagi **SIM** funksiyasini hisoblashimiz kerak. Ikkita yagona klaster uchun (6.3) tenglama nuqta mahsulotiga ekvivalent ekanligini unutmang. Tenglama (6.2) vektor qo'shishga nisbatan nuqta mahsulotining taqsimlanishiga tayanadi. Bu **GAAC** klasterini samarali hisoblash uchun juda muhim bo'lganligi sababli, usulni haqiqiy qiymatli vektor bo'lmagan hujjatlar taqdimotiga osongina qo'llash mumkin emas. Bundan tashqari, (6.2) tenglama faqat nuqta mahsuloti uchun amal qiladi. Ushbu kitobda keltirilgan ko'plab algoritmlar nuqta mahsuloti, kosinus o'xshashligi va *Evklid masofasi* (3.1-bo'lim) bo'yicha deyarli ekvivalent tavsiflarga ega

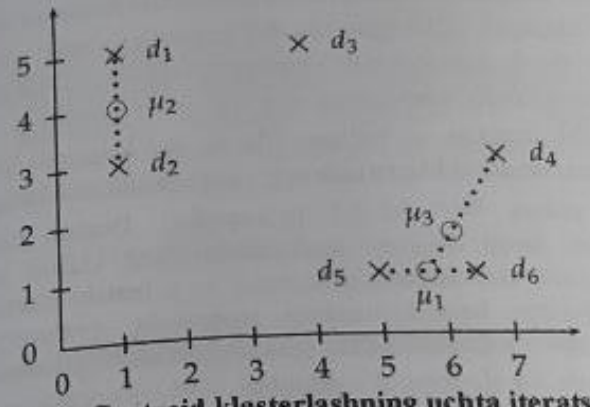
bo'lsa-da, faqat (6.2) tenglama bo'lishi mumkin. Bu bitta havolali/to'liq havolali klasterlash va **GAAC** o'rtasidagi asosiy farq yo'q. Birinchi ikkitasi faqat kirish sifatida o'xshashliklarning kvadrat matritsasini talab qiladi va bu o'xshashliklar qanday hisoblanganiga ahamiyat bermaydi. Xulosa qilib aytadigan bo'lsak, **GAAC** (i) vektor sifatida taqdim etilgan hujjatlarni, (ii) vektorlarning uzunligini normallashtirishni, shuning uchun o'z-o'zidan o'xshashlik 1,0 ga teng bo'lishini va (iii) vektorlar va vektorlar yig'indisi o'rtasidagi o'xshashlik o'lchovi sifatida nuqta mahsulotini talab qiladi. **GAAC** va to'liq havola klasterlash uchun birlashma algoritmlari bir xil, faqat (6.3) tenglamadan 6.8-rasmdagi o'xshashlik funksiyasi sifatida foydalaniladi. Shuning uchun, **GAAC** ning umumiy vaqt murakkabligi to'liq havolali klasterlash bilan bir xil: $D(N_2*\log N)$. To'liq havolali klasterlash singari, **GAAC** ham eng yaxshi birlashma emas. Bu shuni anglatadiki, **GAAC** uchun 6.9-rasmdagi bir bo'g'inli $\Theta(N_2)$ algoritmiga o'xshash hech qanday $D(N_2)$ algoritmi yo'q. Shuningdek, guruh o'rtacha o'xshashligini o'ziga o'xshashlik sifatida belgilashimiz mumkin:

$$SIM-GA'(w_i,w_j)=\frac{1}{(N_i+N_j)^2}\Big(\sum_{d_m\in w_i\cup w_j}\vec{d}_m\Big)^2=\frac{1}{N_i+N_j}\sum_{d_m\in w_i\cup w_j}[\vec{d}_m\cdot\vec{\mu}(w_i\cup w_j)] \quad (6.4)$$

bu yerda **centroid** $\vec{\mu}(w)$ tenglama (3.1) dagi kabi aniqlanadi. Ushbu ta'rif klaster sifatining ***intuitiv ta'rifiga*** teng bo'lib, hujjatlarning o'rtacha o'xshashligi $\vec{d}_m$ klaster markazi $\vec{\mu}$. O'ziga o'xshashlik har doim 1,0 ga teng bo'lib, uzunligi normallashtirilgan vektorlar uchun mumkin bo'lgan maksimal qiymat.

(6.4) tenglamadagi o'ziga o'xshashlik nisbati i o'lchamli klaster uchun $i/i^2=1/i$ Bu kichik klasterlarga nohaq afzallik beradi chunki ular mutanosib ravishda ko'proq o'ziga o'xshashliklarga ega bo'ladi. s o'xshashligi bo'lgan ikkita $\mathbf{d_1}$, $\mathbf{d_2}$ hujjat uchun ularda $SIM-GA(d_1,d_2)=s\le(1+s)/2$ mavjud. Aksincha, $SIM-GA(d_1,d_2)=s\le(1+s)/2$ Ikki hujjatning **SIM-GA($\mathbf{d_1}$, $\mathbf{d_2}$)** o'xshashligi bitta bo'g'inli, to'liq havolali va markazlashtirilgan klasterlashdagi kabidir. (6.3) tenglamadagi ta'rifni afzal ko'ramiz, bu o'rtacha qiymatdan o'ziga o'xshashliklarni istisno qiladi chunki katta klasterlarni kichikroq o'xshashlik nisbati uchun jazolamoqchi emasmiz va barcha to'rtta ***HAC algoritmlari***da hujjat juftligi uchun s izchil o'xshashlik qiymati bo'lishini xohlaymiz.

Misol. 6.6 va 6.7-rasmlardagi nuqtalarga guruh-o'rtacha klasterlashni qo'llang. Uzunlik bo'yicha normallashtirilgan vektorlarni olish uchun ularni uch o'lchovli fazoda birlik sharining yuzasiga xaritalang. Guruhning o'rtacha klasteri bitta va to'liq havola klasterlaridan farq qiladimi?



6.11-rasm. **Centroid klasterlashning uchta iteratsiyasi**

Har bir iteratsiya markazlari eng yaqin joylashgan ikkita klasterni birlashtiradi.

6.4. Centroid klasterlash

Markaziy klasterlashda ikkita klasterning o'xshashligi ularni markazlarining o'xshashligi sifatida aniqlanadi:

$$SIM-CENT(w_i,w_j)=\vec{\mu}(w_i)\cdot\vec{\mu}(w_j)$$

$$=\Big(\frac{1}{N_i}\sum_{d_m\in w_i}\vec{d}_m\Big)\cdot\Big(\frac{1}{N_j}\sum_{d_n\in w_j}\vec{d}_n\Big) \quad (6.5)$$

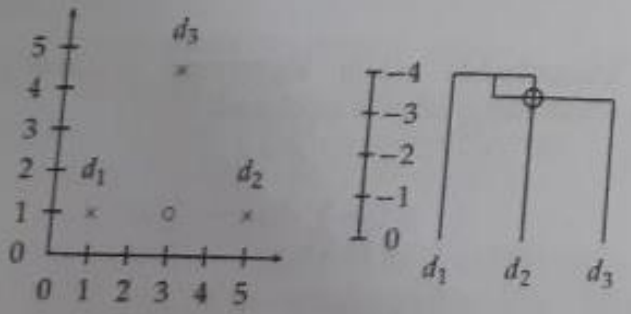
$$=\frac{1}{N_iN_j}\sum_{d_m\in w_i}\sum_{d_n\in w_j}\vec{d}_m\cdot\vec{d}_n \quad (6.6)$$

(6.5) tenglama markaziy o'xshashlikdir. Tenglama (6.6) shuni ko'rsatadiki, markaziy o'xshashlik turli klasterlardagi barcha hujjatlar

juftlarining o'rtacha o'xshashligiga tengdir. Shunday qilib, **GAAC** va ***centroid klasterlash*** o'rtasidagi farq shundaki, **GAAC** o'rtacha juftlik o'xshashligini hisoblashda barcha hujjatlar juftlarini hisobga oladi (6.3-rasm, (d)), markaziy klasterlash esa bir xil klasterdagi juftlarni istisno qiladi (6.3-rasm, (c)). 6.11-rasmda markazlashtirilgan klasterning dastlabki uch bosqichi ko'rsatilgan. Birinchi ikkita iteratsiya centroid μ_1 va $\{\mathbf{d_1}, \mathbf{d_2}\}$ centroid μ_2 bo'lgan $\{\mathbf{d_5}, \mathbf{d_6}\}$ klasterlarini hosil qiladi chunki $\langle \mathbf{d_5}, \mathbf{d_6} \rangle$ va $\langle \mathbf{d_1}, \mathbf{d_2} \rangle$ juftliklari eng yuqori markazga ega o'xshashliklardir. Uchinchi iteratsiyada markazning eng yuqori o'xshashligi μ_3 va $\mathbf{d_4}$ o'rtasida bo'lib, markaz μ_3 bo'lgan $\{\mathbf{d_4}, \mathbf{d_5}, \mathbf{d_6}\}$ klasterini hosil qiladi. GAAC singari, centroid klasterlash eng yaxshi birlashma barqaror emas va shuning uchun $\Theta(N^2 \log N)$ (6.6-mashq). Boshqa uchta HAC algoritmlaridan farqli o'laroq, markazlashtirilgan klaster monotonik emas. Inversiyalar deb ataladigan holatlar ro'y berishi mumkin. 6.12-rasmdagi misoldagi kabi klasterlash jarayonida o'xshashlik ortishi mumkin, bu yerda o'xshashlik salbiy masofa sifatida belgilanadi. Birinchi birlashtirishda $\mathbf{d_1}$ va $\mathbf{d_2}$ ning o'xshashligi $-(4-\varepsilon)$ ga teng. Ikkinchi birlashmada $\mathbf{d_1}$ va $\mathbf{d_2}$ (doira) va $\mathbf{d_3}$ markazlarining o'xshashligi

$$sim \approx -\cos(\pi/6) \times 4 = -\sqrt{3}/2 \times 4 \approx -3.46 > -(4-\varepsilon)$$

. Bu inversiyaga misol: ikkita klasterlash bosqichining ushbu ketma-ketligida o'xshashlik ortadi. Monotonik HAC algoritmida o'xshashlik takrorlanishdan iteratsiyaga monoton ravishda kamayadi.



HAC klasterlash bosqichlari qatorida o'xshashlikning kuchayishi kichik klasterlar katta klasterlarga qaraganda ko'proq mos keladi degan asosiy taxminga zid keladi. Dendrogrammadagi inversiya oldingi birlashma chizig'idan pastroq bo'lgan gorizontal birlashma chizig'i sifatida namoyon bo'ladi. 6.1 va 6.5-rasmlardagi barcha birlashma

chiziqlari avvalgilaridan yuqoriroqdir chunki bitta bo'g'inli va to'liq bo'g'inli klasterlash monotonik klasterlash algoritmlaridir. Monoton bo'lmaganligiga qaramay, markazlashtirilgan klasterlash ko'pincha qo'llaniladi chunki uning o'xshashligi ikkita markazning o'xshashligini o'lchaydi - *GAAC*dagi barcha juftlik o'xshashliklarining o'rtacha qiymatidan kontseptual jihatdan sodda. 6.11-rasm markaziy klasterni tushunish uchun kerak bo'lgan barcha narsadir. *GAAC* qanday ishlashini tushuntirib beradigan oddiy grafik yo'q.

Misol. N ta hujjatning belgilangan to'plami uchun bitta va to'liq havolali klasterlashdagi klasterlar o'rtasida N^2 tagacha aniq o'xshashliklar mavjud. **GAAC** va ***centroid*** klasterlashda nechta aniq klaster o'xshashliklari mavjud?

6.5. Ierarxik klasterlashning optimalligi

Ierarxik klasterlashning optimallik shartlarini aniq ifodalash uchun birinchi navbatda klasterlashning **COMB-SIM** kombinatsiyasi o'xshashligini aniqlaymiz $\Omega = \{w_1, \ldots, w_k\}$ **K** klasterlarining eng kichik kombinatsiyasi o'xshashligi sifatida hisoblanadi:

$$COMB - SIM(\{w_1, \ldots, w_k\}) = \min_k COMB - SIM(w_k)$$

Eslatib o'tamiz, w_1 *va* w_2 ning birlashuvi sifatida yaratilgan δ klasterining kombinatsiya o'xshashligi w_1 *va* w_2 ning o'xshashligidir. Keyin $\Omega = \{w_1, \ldots, w_k\}$ optimal bo'ladi, agar **k** klasterli barcha Ω' klasterlar, $\mathbf{k} \leq \mathbf{K}$, kombinatsiya o'xshashligi kamroq bo'lsa:

$$|\Omega| \leq |\Omega| \Rightarrow COMB - SIM(\Omega') \leq COMB - SIM(\Omega)$$

6.12-rasmda centroid klasterlash optimal emasligini ko'rsatadi. Klasterlash $\{\{\mathbf{d_1}, \mathbf{d_2}\}, \{\mathbf{d_3}\}\}$ (K=2 uchun) $-(4-\varrho)$ va $\{\{\mathbf{d_1}, \mathbf{d_2}, \mathbf{d_3}\}\}$ (K = 1 uchun) kombinatsiya o'xshashligiga ega. Shunday qilib, birinchi birlashmada hosil qilingan $\{\{\mathbf{d_1}, \mathbf{d_2}\}, \{\mathbf{d_3}\}\}$ klasterlari optimal emas chunki kamroq klasterli $(\{\{\mathbf{d_1}, \mathbf{d_2}, \mathbf{d_3}\}\})$ yuqoriroq klasterlar mavjud. Centroid klasterlash optimal emas chunki inversiyalar paydo bo'lishi mumkin. Yuqoridagi optimallikning ta'rifi, agar u faqat birlashish tarixi bilan birga klasterlash uchun qo'llanilsa cheklangan holda foydalanish mumkin edi. Biroq, uchta inversiyasiz algoritm uchun kombinatsiya o'xshashligini uning tarixini bilmasdan klasterdan o'qib chiqish mumkinligini ko'rsatishimiz mumkin. Kombinatsiyaviy o'xshashlikning to'g'ridan-

to'g'ri ta'riflari quyidagicha: Klasterning kombinatsiyalangan o'xshashligi klasterning har qanday ikki bo'linmasining eng kichik o'xshashligidir, bunda ikki qismning o'xshashligi ikki qismdan har qanday ikkita hujjat o'rtasidagi eng katta o'xshashlikdir:

$$COMB - SIM(w) = \min_{(w',w-w')} \max_{d_i \in w'} \max_{d_j \in w-w'} SIM(d_i, d_j)$$

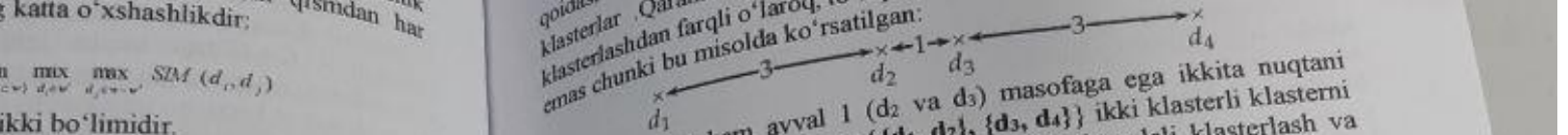
bu yerda har bir $\langle w', w - w' \rangle$ w-ning ikki bo'limidir.

to'liq havola Klaster **ō** ning kombinatsiya o'xshashligi **ō** dagi har qanday ikkita nuqtaning eng kichik o'xshashligidir: $w : \min_{d_i \in w} \min_{d_j \in w} SIM(d_i, d_j)$

GAAC klasterning kombinatsiya o'xshashligi w barcha juftlik o'xshashliklarining o'rtacha qiymatidir (bu yerda o'ziga o'xshashliklar o'rtachaga kiritilmagan), tenglama (6.3). Agar kombinatsiya o'xshashligining ushbu ta'riflaridan foydalansak, optimallik klasterlar to'plamini hosil qiluvchi jarayonning emas, balki klasterlar to'plamining xususiyatidir.

Endi **K** klasterlar soni bo'yicha bir bo'g'inli klasterlashning optimalligini isbotlashimiz mumkin. Ikkita juft hujjatlar bir xil o'xshashlikka ega bo'lmagan holatga dalil keltiramiz, lekin uni bog'langan holatga osonlik bilan kengaytirish mumkin. Dalilning induktiv asosi shundan iboratki, K=N klasterli klasterlar kombinatsiyasi o'xshashligi 1,0 ga ega, bu mumkin bo'lgan eng katta qiymatdir. Induksion gipoteza shuni ko'rsatadiki, K klasterli Ω_K yagona bo'g'inli klaster optimal hisoblanadi: $COMB - SIM(\Omega_K) \geq COMB - SIM(\Omega'_K)$ barcha Ω' uchun. Qarama-qarshilik uchun Ω_{K-1} klasterini olamiz deb faraz qilaylik Ω_K dagi ikkita eng o'xshash klasterni birlashtirish optimal emas va buning o'rniga Ω'_K, Ω'_{K-1} ni birlashtirishning boshqa ketma-ketligi K - 1 klasterlari bilan optimal klasterlashishga olib keladi. Ω'_{K-1} optimal va Ω_{K-1} $COMB - SIM(\Omega'_{K-1}) > COMB - SIM(\Omega_K) \geq COMB - SIM(\Omega'_K)$ (ΩK−1) kabi emas degan taxminni yozishimiz mumkin. 1-holat: **s** bilan bog'langan ikkita hujjat. $COMB - SIM(\Omega_{K-1})$ Ω_K da bir xil klasterda. Bir xil klasterda bo'lishi mumkin, agar birlashma ketma-ketligida **s** dan kichikroq o'xshashlik Ω_K ni hosil qilgan bo'lsa. Bu **s > COMB-** ni bildiradi. SIM Ω_K shunday qilib, COMB- SIM(Ω'_{K-1}) = s>COMB-SIM Ω_K COMB- SIM (Ω_K)> COMB-SIM (Ω'_{K-1})Qarama-qarshilik. 2-holat: **s = COMB- SIM** Ω_K bilan bog'langan ikkita hujjat **Ō** da bir xil klasterda

emas. Lekin **s = COMB- SIM** Ω_K shuning uchun bitta havolali birlashma qoidasi bu ikkalasini birlashtirishi kerak edi. Ω_K ni qayta ishlashda klasterlar .Qarama-qarshilik, Ω_{K-1} optimal hisoblanadi. Bitta havolali klasterlashdan farqli o'laroq, to'liq havolali klasterlash va **GAAC** optimal emas chunki bu misolda ko'rsatilgan:



Ikkala algoritm ham avval 1 (d_2 va d_3) masofaga ega ikkita nuqtani birlashtiradi va shuning uchun **{{d₁, d₂}, {d₃, d₄}}** ikki klasterli klasterni topa olmaydi. Lekin **{{d₁, d₂}, {d₃, d₄}}** to'liq havolali klasterlash va *GAC*ning optimallik mezonlari bo'yicha optimal hisoblanadi. Biroq, to'liq havolali klasterlash va **GAAC**ning birlashuv mezonlari taxminiy sferiklik desideratumini taxmin qiladi. Yagona bo'g'inli klasterlashning birlashma mezonidan ham yaxshiroq. Ko'pgina ilovalarda sharsimon klasterlarni xohlaymiz. Shunday qilib, bir bo'g'inli klasterlash optimalligi tufayli dastlab afzal ko'rinsa ham ko'plab hujjatlar klasterlash ilovalarida noto'g'ri mezonga nisbatan optimal hisoblanadi.

6.1-jadvalda ushbu bobda keltirilgan to'rtta **HAC** algoritmlarining xususiyatlari jamlangan. Hujjatlarni klasterlash uchun **GAAC** ni tavsiya qiladi chunki u odatda ilovalar uchun eng yaxshi xususiyatlarga ega klasterni ishlab chiqaradigan usuldir. Zanjirlanishdan chetga chiqishga sezuvchanlikdan va inversiyalardan aziyat chekmaydi. Ushbu tavsiyaga ikkita istisno mavjud. Birinchidan, vektor bo'lmagan tasvirlar uchun **GAAC** qo'llanilmaydi va klasterlash odatda to'liq havola usuli bilan amalga oshirilishi kerak. Ikkinchidan, ba'zi ilovalarda klasterlashning maqsadi to'liq ierarxiyani yoki butun hujjatlar to'plamining to'liq bo'limini yaratish emas. Masalan, birinchi voqeani aniqlash yoki yangilikni aniqlash - bu yangiliklar oqimida voqeaning birinchi sodir bo'lishini aniqlash vazifasi. Ushbu vazifaga yondashuvlardan biri qisqa vaqt ichida sim orqali yuborilgan va oldingi barcha hujjatlardan farqli bo'lgan hujjatlar ichida qattiq klasterni topishdir. Masalan, 2001-yil 11-sentabrda Jahon savdo markaziga qilingan hujumdan keyin bir necha daqiqada sim orqali yuborilgan hujjatlar shunday klasterni tashkil qiladi. Yagona bog'lamli klasterlashning o'zgarishlari bu vazifani yaxshi bajarishi mumkin chunki global tuzilma emas, balki vektor fazosining kichik qismlarining tuzilishi bu holatda muhim ahamiyatga ega. Misol.

Kombinatsiya o'xshashligining ikkita ta'rifining ekvivalentligini ko'rsating.

6.6. Bo'linuvchi klasterlash

Hozircha faqat ***aglomerativ klasterlash***ni ko'rib chiqdik, lekin ***klaster ierarxiyasi***ni yuqoridan pastga ham yaratish mumkin. ***Ierarxik klasterlash***ning bu varianti yuqoridan ***pastga klasterlash*** yoki ***bo'linuvchi klasterlash*** deb ataladi. Barcha hujjatlarni bitta klasterda yuqoridan boshlaymiz. Klaster tekis klasterlash algoritmi yordamida bo'linadi. Ushbu protsedura har bir hujjat o'zining yagona klasterida bo'lgunga qadar ***rekursiv*** ravishda qo'llaniladi. ***Yuqoridan pastga klasterlash pastdan yuqoriga klasterlash***dan kontseptual jihatdan murakkabroq chunki ularga "***kichik dastur***" sifatida ikkinchi tekis klasterlash algoritmi kerak. Agar ***to'liq ierarxiyani*** alohida hujjat varaqlarigacha yaratmasak samaraliroq bo'lish afzalligi bor. Yuqori darajalarning belgilangan soni uchun K-vositalari kabi samarali tekis algoritmdan foydalangan holda, yuqoridan pastga algoritmlar hujjatlar va klasterlar soni bo'yicha chiziqli bo'ladi. Shunday qilib, hech bo'lmaganda kvadratik bo'lgan **HAC** algoritmlariga qaraganda tezroq ishlaydi. Ba'zi hollarda bo'linuvchi algoritmlar pastdan yuqoriga algoritmlarga qaraganda aniqroq ierarxiyalarni ishlab chiqaradigan dalillar mavjud. 6.9-bo'limdagi ***K-o'rtachalarni ikkiga bo'lish*** bo'yicha havolalarga qarang. Pastdan yuqoriga yo'naltirilgan usullar dastlab global taqsimotni hisobga olmagan holda mahalliy tasvirlar asosida klasterlash qarorlarini qabul qiladi. Bu erta qarorlarni bekor qilib bo'lmaydi. Yuqoridan pastga klasterlash yuqori darajadagi taqsimlash qarorlarini qabul qilishda global taqsimot haqida to'liq ma'lumotdan foyda oladi.

method	combination similarity	time compl.	optimal?	comment
single-link	max inter-similarity of any 2 docs	$\Theta(N^2)$	yes	chaining effect
complete-link	min inter-similarity of any 2 docs	$\Theta(N^2 \log N)$	no	sensitive to outliers
group-average	average of all sims	$\Theta(N^2 \log N)$	no	best choice for most applications
centroid	average inter-similarity	$\Theta(N^2 \log N)$	no	inversions can occur

6.7. Klaster belgilari

Yassi klasterlash va ierarxik klasterlashning ko'pgina ilovalarida, xususan, tahlil vazifalari va foydalanuvchi interfeyslarida (5.1-jadval) foydalanuvchilar klasterlar bilan o'zaro aloqada bo'ladi. Bunday sozlamalarda klasterlarni belgilash kerak, shunda foydalanuvchilar klaster nima haqida ekanligini ko'rishlari mumkin. Differentsial klaster yorlig'i bir klasterdagi atamalar taqsimotini boshqa klasterlar bilan solishtirish orqali klaster belgilarini tanlaydi. Ular birinchi qismning 2.5-bo'limda taqdim etgan xususiyatlarni tanlash usullarining barchasi differentsial klaster belgilari uchun ishlatilishi mumkin. Xususan, o'zaro ma'lumot (MI) (birinchi qism 2.5.1-bo'lim) yoki shunga o'xshash ma'lumot olish va x^2-testi (2.5.2-bo'lim) klaster belgilaridan farqli ravishda bir klasterni tavsiflovchi klaster belgilarini aniqlaydi. Differensial testning nodir atamalar uchun kombinatsiyasi ko'pincha eng yaxshi yorliqlash natijalarini beradi chunki kamdan-kam atamalar umuman klasterni ifodalashi shart emas. 6.2-jadvalda K-ko'rsatkichlar klasteriga uchta yorliqlash usulini qo'llaymiz. Bu misolda MI va x^2 o'rtasida deyarli farq yo'q. Shuning uchun ikkinchisini o'tkazib yuboriladi.

Klaster-ichki yorliqlash boshqa klasterlarga emas, faqat klasterning o'ziga bog'liq bo'lgan yorliqni hisoblaydi. Klasterni centroidga eng yaqin hujjat sarlavhasi bilan belgilash klasterning ichki usullaridan biridir. Sarlavhalarni o'qish atamalar ro'yxatidan ko'ra osonroqdir. To'liq sarlavhada MI tomonidan tanlangan eng yaxshi 10 ta atamaga kirmagan muhim kontekst ham bo'lishi mumkin. Internetda sarlavha matni sarlavhaga o'xshash rol o'ynashi mumkin chunki sahifaga ishora qiluvchi sarlavha matn uni mazmunining qisqacha xulosasi bo'lib xizmat qilishi mumkin. 6.2-jadvalda 9-klaster sarlavhasi uning ko'pgina hujjatlari Checheniston mojarosiga tegishli ekanligini ko'rsatadi, bu faktni MI shartlari oshkor etmaydi. Biroq, bitta hujjat klasterdagi barcha hujjatlarni ifodalashi dargumon. Masalan, 4-klaster, uning tanlangan sarlavhasi noto'g'ri. Klasterning asosiy mavzusi neftdir. Dolli to'foni haqidagi maqolalar faqat neft narxiga ta'siri tufayli ushbu klasterda tugadi. Yorliq sifatida klasterning markaziy qismida yuqori vaznli atamalar ro'yxatidan ham foydalanish mumkin. Bunday yuqori vaznli atamalar (yoki undan ham yaxshiroq, iboralar, ayniqsa ismli iboralar) differensial usullarda

bo'lgani kabi farqlash uchun filtrlanmagan bo'lsa ham, ko'pincha bir nechta sarlavhalarga qaraganda klasterni ko'proq ifodalaydi. Biroq, iboralar ro'yxati foydalanuvchilar uchun yaxshi tayyorlangan sarlavhaga qaraganda ko'proq vaqt talab etadi.

Klaster-ichki usullari samarali lekin ular butun to'plamda tez-tez uchraydigan atamalarni faqat klasterda tez-tez uchraydigan atamalardan ajrata olmaydi. Yil yoki seshanba kabi atamalar klasterda eng ko'p uchraydigan atamalar bo'lishi mumkin ammo ular neft kabi ma'lum bir mavzuga ega bo'lgan klaster mazmunini tushunishda yordam bermaydi.

6.2-jadvalda centroid usuli MI (kuchlar, stol) dan ko'ra bir necha ko'proq ma'lumotga ega bo'lmagan atamalarni (000, sud, sent, sentyabr) tanlaydi, ammo har ikkala usulda tanlangan atamalarning aksariyati yaxshi tavsiflovchi hisoblanadi. Tanlangan shartlarni skanerlash orqali klasterdagi hujjatlar yaxshi tushuniladi.

6.2-jadval. **Avtomatik hisoblangan klaster belgilari**

	# docs	labeling method		
		centroid	mutual information	title
4	622	oil plant mexico production crude **power 000 refinery gas** bpd	plant oil production **barrels** crude bpd mexico **dolly capacity petroleum**	MEXICO: Hurricane Dolly heads for Mexico coast
9	1017	police security **russian** people military peace killed told **grozny** court	police killed military security peace told **troops forces rebels** people	RUSSIA: Russia's Lebed meets rebel chief in Chechnya
10	1259	00 000 tonnes traders futures wheat prices **cents september** tonne	**delivery** traders futures tonne tonnes **desk** wheat prices 000 00	USA: Export Business - Grain/oilseeds complex

Bu Reuters-RCV1-dagi dastlabki 10000 ta hujjatlarning **K** klasteridagi o'nta klasterdan uchtasi (4, 9 va 10) uchundir. Oxirgi uchta ustunda uchta yorliqlash usuli bo'yicha hisoblangan klaster xulosalari ko'rsatilgan. Markazda eng yuqori vaznli atamalar, o'zaro ma'lumotlar va klasterning markaziy qismiga (sarlavha) eng yaqin hujjat nomi keltirilgan. Birinchi ikkita usuldan faqat bittasi bilan tanlangan shartlar qalin qilib yozilgan.

Ierarxik klasterlash uchun klaster belgilarida qo'shimcha qiyinchiliklar paydo bo'ladi. Nafaqat daraxtdagi ichki tugunni uning

birodarlaridan, balki uning ota-onasi va bolalaridan ham farqlashimiz kerak. To'liq tugunlardagi hujjatlar ta'rifiga ko'ra ularning ota-ona tugunlarining a'zolaridir, shuning uchun ota-onani uning bolalaridan ajratib turadigan yorliqlarni topish uchun sodda differentsial usuldan foydalana olmaymiz. Biroq, umumiy yig'ish chastotasi va ma'lum bir klasterda tarqalganlik kombinatsiyasiga asoslangan murakkabroq mezonlar, atama bola tugun yoki ota-ona tugun uchun ko'proq ma'lumot beruvchi belgi ekanligini aniqlashi mumkin.

6.8. Amalga oshirish bo'yicha eslatmalar

Ko'p sonli nuqta mahsulotini hisoblashni talab qiladigan ko'pgina muammolar ***teskari indeks***dan foydalanadi. Bu OAK klasteri uchun ham amal qiladi. ***Invert indeks*** tufayli hisob-kitoblarni tejash, agar ko'p nol o'xshashliklar mavjud bo'lsa, katta bo'ladi - chunki ko'plab hujjatlar hech qanday shartlarni qabul qilmaydi yoki agressiv to'xtash ro'yxati qo'llaniladi. Past o'lchamlarda ko'proq tajovuzkor optimallashtirishlar mumkin, bu esa ko'pgina juftlik o'xshashliklarini hisoblashni keraksiz qiladi. Biroq, yuqori o'lchamlarda bunday algoritmlar ma'lum emas. **kNN** tasnifida bir xil muammoga duch keldik (3.7-bo'lim). **GAAC**-ni yuqori o'lchamdagi katta hujjatda ishlatganda, zich markazlardan qochishga ehtiyot bo'lishimiz kerak. Zich centroidlarni klasterlash uchun vaqt kerak bo'lishi mumkin $\Theta(MN^2 \log N)$, bu yerda M - lug'at hajmi, to'liq havolali klasterlash esa $\Theta(M_{ave} N^2 \log N)$ bu yerda M_{ave} - hujjat lug'atining o'rtacha hajmi. Shunday qilib, katta lug'atlar uchun to'liq havola klasterlash GAACning optimallashtirilmagan tatbiq etilishidan ko'ra samaraliroq bo'lishi mumkin. Bu muammoni 5-bobda ***K-vositalarni klasterlash*** kontekstida muhokama qildik va ikkita yechim taklif qildik: o'rniga ***siyrak medoidlar*** yordamida klasterlarni ifodalaydi. Ushbu optimallashtirishlar **GAAC** va ***centroid klasterlash***da ham qo'llanilishi mumkin.

Ushbu optimallashtirishlar bilan ham, **HAC** ***algoritmlari*** hammasi $\Theta(N^2)$ *yoki* $\Theta(N^2 \log N)$ va shuning uchun 1000000 yoki undan ortiq hujjatlarning katta to'plamlari uchun bajarib bo'lmaydi. Bunday katta to'plamlar uchun **HAC** faqat ***K-means*** kabi tekis klasterlash algoritmi bilan birgalikda ishlatilishi mumkin. Eslatib o'tamiz, ***K-vositalarni***

boshlash uchun urg'ular to'plami talab qilinadi (5.5-rasm). Agar bu urg'ular noto'g'ri tanlangan bo'lsa unda hosil bo'lgan klaster sifatsiz bo'ladi. Yuqori sifatli urg'ularni hisoblash uchun ***HAC algoritmi***dan foydalanishimiz mumkin. Agar ***HAC algoritmi*** o'lchami $\sqrt{N}$ bo'lgan hujjatlar to'plamiga qo'llanilsa, u holda K-ning umumiy ish vaqti HAC urg'usini hosil qilish degani D(N) ga teng. Buning sababi shundaki, kvadrat algoritmni $\sqrt{N}$ o'lchamdagi namunaga qo'llash umumiy murakkablik $\Theta(N)$ ga ega. Chiziqlilikni kafolatlash uchun $\Theta(N^2 \log N)$ algoritmiga tegishli tuzatish kiritilishi mumkin. Bu algoritm ***Buckshot algoritmi*** deb ataladi. U HACning determinizmi va yuqori ishonchliligini K-vositalari samaradorligini birlashtiradi.

Misol. Yagona bog'lamli klasterni grafikning minimal oraliq daraxtidan ham hisoblash mumkin. Minimal kengaytmali daraxt grafikning uchlarini eng kichik narxda bog'laydi, bunda xarajat grafikning barcha qirralari bo'ylab yig'indisi sifatida aniqlanadi. Holatlarimizda ularning chekka narxi ikki hujjat orasidagi masofadir. Ko'rsating, agar $\Delta k-1>\Delta k>$ bo'lsa. .. $>\Delta 1$ - minimal oraliqli daraxtning qirralari xarajatlari, keyin bu qirralarning ***bir bo'g'inli klasterni*** qurishda k-1 birlashuviga mos keladi.

Misol. Yagona bog'lamli klasterlash eng yaxshi birlashuvchi barqaror ekanligini va GAAC va centroid klasterlash eng yaxshi birlashuvchi barqaror emasligini ko'rsating.

Misol.

a. Ikki xil tildagi hujjatlar to'plamida 2-means klasterini ishga tushirishni ko'rib chiqing. Qanday natija kutgan bo'lardingiz?

b. ***HAC algoritmi***ni ishga tushirishda ham xuddi shunday natijani kutasizmi?

Misol. ***Reuters-21578-ni*** yuklab oling. Faqat xom, foiz va don sinflarida bo'lgan hujjatlarni saqlang. Ushbu uchta sinfdan bir nechtasiga a'zo bo'lgan hujjatlarni tashlang. Hujjatlarning (i) bitta bo'g'inli, (ii) to'liq havolasi, (iii) GAAC, (iv) markazlashtirilgan klasterini hisoblang. (v) K = 3 klasterni olish uchun har bir dendrogramni yuqoridan ikkinchi shoxchadan kesib oling. 4 ta klasterning har biri uchun ***Rand indeksi***ni hisoblang. Qaysi klasterlash usuli yaxshiroq ishlaydi?

Misol. Faraz qilaylik, HAC ishga tushirilishi K=7 bo'lgan klasterlashning oldindan tanlangan yaxshilik o'lchovi bo'yicha eng

yuqori qiymatga ega ekanligini topdi. K=7 bo'lgan barcha klasterlar orasida eng yuqori qiymatli klasterni topdikmi?

Misol. Chiziqda N nuqtadan iborat bitta bo'g'inli klasterni yaratish vazifasini ko'rib chiqing:

Faqat taxminan N ta o'xshashlikni hisoblashimiz kerakligini ko'rsating. Chiziqdagi nuqtalar to'plami uchun bitta bo'g'inli klasterlashning umumiy murakkabligi qanday?

Misol. Bitta bo'g'inli, to'liq bo'g'inli va guruhli o'rtacha klasterlar monoton ekanligini isbotlang.

Misol. N nuqta uchun K klasterlarga $\leq N^K$ turli tekis klasterlar mavjud (5.2-bo'lim). N ta hujjatning turli ierarxik klasterlari (yoki dendrogrammalari) soni qancha? Berilgan **K** va **N** uchun ko'proq tekis klasterlar yoki ko'proq ierarxik klasterlar mavjudmi?

6- bob bo'yicha foydalanilgan adabiyotlar

Papineni, Kishore.
2001.
Why inverse document frequency?
In *Proc. North American Chapter of the Association for Computational Linguistics*, pp. 1-8.

Pavlov, Dmitry, Ramnath Balasubramanyan, Byron Dom, Shyam Kapur, and Jignashu Parikh.
2004.
Document preprocessing for naive Bayes classification and clustering with mixture of multinomials.
In *Proc. KDD*, pp. 829-834.

Pelleg, Dan, and Andrew Moore.
1999.
Accelerating exact k-means algorithms with geometric reasoning.
In *Proc. KDD*, pp. 277-281. ACM Press.
DOI: *doi.acm.org/10.1145/312129.312248*.

Pelleg, Dan, and Andrew Moore.
2000.
X-means: Extending k-means with efficient estimation of the number of

clusters.
In *Proc. ICML*, pp. 727-734. Morgan Kaufmann.
Perkins, Simon, Kevin Lacker, and James Theiler.
2003.
Grafting: Fast, incremental feature selection by gradient descent in function space.
JMLR 3: 1333-1356.
Persin, Michael.
1994.
Document filtering for fast ranking.
In *Proc. SIGIR*, pp. 339-348. ACM Press.
Persin, Michael, Justin Zobel, and Ron Sacks-Davis.
1996.
Filtered document retrieval with frequency-sorted indexes.
JASIS 47 (10): 749-764.
Zobel, Justin, and Philip Dart.
1995.
Finding approximate matches in large lexicons.
Software Practice and Experience 25 (3): 331-345.
URL: *citeseer.ifi.unizh.ch/zobel95finding.html*.
Zobel, Justin, and Philip Dart.
1996.
Phonetic string matching: Lessons from information retrieval.
In *Proc. SIGIR*, pp. 166-173. ACM Press.
Zobel, Justin, and Alistair Moffat.
2006.
Inverted files for text search engines.
ACM Computing Surveys 38 (2).
Zobel, Justin, Alistair Moffat, Ross Wilkinson, and Ron Sacks-Davis.
1995.
Efficient retrieval of partial documents.
IP&M 31 (3): 361-377.
DOI: *dx.doi.org/10.1016/0306-4573(94)00052-5*.
Zukowski, Marcin, Sandor Heman, Niels Nes, and Peter Boncz.
2006.
Super-scalar RAM-CPU cache compression.
In *Proc. International Conference on Data Engineering*, p. 59. IEEE

Computer Society.
DOI: *dx.doi.org/10.1109/ICDE.2006.150*.

6- bob bo'yicha nazariy va amaliy test savollari

1. Ustunli indeks qanday saqlanadi?
   A) yozuv uchun barcha ustunlarni birga saqlash o'rniga, har bir ustun indeksdagi boshqa barcha qatorlar bilan alohida saqlanadi
   B) bu so'rovda so'ralgan barcha ustunlarni klasterlangan indeksni qo'shimcha izlamasdan qondiradigan ko'rsatkich
   C) Qo'shimcha yozish xarajatlari evaziga ma'lumotlar bazasi jadvalidagi ma'lumotlarni qidirish operatsiyalari tezligini yaxshilaydigan ma'lumotlar tuzilmasi
   D) Ichki foydalanish uchun ma'lumotlar bazasi tomonidan tuziladi
2. Qo'shimcha yozish xarajatlari evaziga ma'lumotlar bazasi jadvalidagi ma'lumotlarni qidirish operatsiyalari tezligini yaxshilaydigan ma'lumotlar tuzilmasi nima?
   A) Ma'lumotlar bazasi indeksi
   B) Kompozit indekslar
   C) Klasterli indeks
   D) klasterli bo'lmagan indeks
3. Indeks klasterlangan bo'ladi, agar _____
   A) faylning ma'lumotlar yozuvlari indeksning ma'lumotlar yozuvlari bilan bir xil tartibda tartibga solinsa
   B) u asosiy kalitni o'z ichiga olgan maydonlar to'plamida
   C) u nomzod kalitini tashkil etuvchi maydonlar to'plamida
   D) faylning ma'lumotlar yozuvlari indeksning ma'lumotlar yozuvlari bilan bir xil tartibda bo'lmaganda
4. Klasterlash indeksi qanday turdagi maydonlarda aniqlanadi?
   A) kalit bo'lmagan va buyurtma
   B) kalit bo'lmagan va tartibsiz
   C) kalit va buyurtma
   D) kalit va buyurtmasiz
5. FAT (fayllarni taqsimlash jadvali) asosidagi fayl tizimi qo'llanilmoqda va FATdagi har bir yozuvning umumiy yuki 4 bayt o'lchamda. Fayl tizimi saqlanadigan 100×10^6 bayt diskni va ma'lumotlar blokining hajmi 10^3 baytni hisobga olsak, ushbu diskda 10^6

bayt birliklarda saqlanishi mumkin bo‘lgan faylning maksimal hajmi _____ bo‘ladi.

A) 99,55 dan 99,65 gacha
B) 100,5 dan 101,4 gacha
C) 97,2 dan 98,5 gacha
D) 89,1 dan 91,2 gacha

6. Bloklarni faylga indeks ajratish sxemasida faylning mumkin bo‘lgan maksimal hajmi quyidagilarga bog‘liq:

A) indeks uchun ishlatiladigan bloklar soni va bloklarning o‘lchami
B) bloklarning o‘lchami va bloklar manzilining o‘lchami
C) bloklarning o‘lchami, indeks uchun ishlatiladigan bloklar soni va bloklar manzilining o‘lchami
D) A va B

7. Fayl shunday tashkil etilganki, ma’lumotlar yozuvlari tartibi Ba’zi indeksdagi ma’lumotlar kiritilishi tartibi bilan bir xil yoki unga yaqin bo‘ladi. Bu indeks qanday nomlanadi?

A) Klasterlangan
B) Zich
C) Siyrak
D) Klastersiz

8. T1 ma’lumotlar bazasi jadvali 2000 ta yozuvga ega va 80 ta disk blokini egallaydi. Boshqa T2 jadvali 400 ta yozuvga ega va 20 ta disk blokini egallaydi. Ushbu ikkita jadval ushbu ikkita jadvaldagi har bir juft yozuv uchun baholanishi kerak bo‘lgan belgilangan birlashma shartiga muvofiq birlashtirilishi kerak. Mavjud xotira bufer maydoni bir vaqtning o‘zida istalgan vaqtda T1 uchun bitta yozuv blokini va T2 uchun bitta blokni saqlashi mumkin. Ikkala jadvalda ham indeks mavjud emas. Agar tashqi siklda foydalanish uchun jadvalning eng to‘g‘ri tanlovi bilan birlashishni amalga oshirish uchun Nested-loop qo‘shilish algoritmi ishlatilsa, ma’lumotlarni o‘qish uchun zarur bo‘lgan bloklarga kirishlar soni nechta?

A) 32020
B) 40080
C) 800000
D) 100

9. T1 ma’lumotlar bazasi jadvali 2000 ta yozuvga ega va 80 ta disk blokini egallaydi. Boshqa T2 jadvali 400 ta yozuvga ega va 20 ta disk

blokini egallaydi. Ushbu ikkita jadval ushbu ikkita jadvaldagi har bir juft yozuv uchun baholanishi kerak bo‘lgan belgilangan birlashma shartiga muvofiq birlashtirilishi kerak. Mavjud xotira bufer maydoni bir vaqtning o‘zida istalgan vaqtda T1 uchun bitta yozuv blokini va T2 uchun bitta blokni saqlashi mumkin. Ikkala jadvalda ham indeks mavjud emas. Agar tashqi tsikldagi jadvalni yana eng to‘g‘ri tanlagan holda, Nested-loop join o‘miga blokirovka qilingan-loop qo‘shilishi ishlatilsa, ma’lumotlarni o‘qish uchun zarur bo‘lgan bloklarga kirishlar soni kamayadi.

A) 30400
B) 38400
C) 798400
D) 0

10. Quyidagilardan qaysi biri to‘g‘ri?

A) B+ daraxtlarida diapazon so‘rovlari tezroq
B) B-daraxtlar diskda ma’lumotlarni saqlash uchun, B + daraxtlari esa asosiy xotira uchun
C) B-daraxtlar birlamchi indekslar uchun, B+ daraxtlari esa ikkilamchi indekslar uchun
D) B+ daraxtining balandligi yozuvlar soniga bog‘liq emas.

Atama-hujjat matritsasi tushunchasi haqida avvalgi boblarda qisqacha aytib o'tilgan edi. Masalan, **M×N** matritsa **C** berilgan, uning har bir satri atamani va har bir ustuni to'plamdagi hujjatni ifodalaydi. Hatto oddiy o'lchamdagi to'plam uchun ham **C** ***atama-hujjat matritsasi*** bir necha o'n minglab qator va ustunlarga ega bo'lishi mumkin. 7.1.1-bo'limda birinchi navbatda ***matritsa dekompozitsiyasi*** deb nomlanuvchi chiziqli algebradan amallar sinfini ishlab chiqamiz. 7.2-bo'limda ***hujjat-hujjat matritsasiga past darajali yaqinlashish***ni qurish uchun matritsa taqsimlanishining maxsus shaklidan foydalaniladi. 7.3-bo'limda bunday past darajali yaqinlashuvlarni ***hujjatlarni indekslash*** va ***qidirishda*** qo'llanilishi ko'rib chiqiladi, bu usul ***yashirin semantik indeksatsiya*** deb ataladi. ***Yashirin semantik indeksatsiya*** ma'lumot olish uchun reyting va reytingda muhim kuch sifatida aniqlanmagan bo'lsa-da, u bir qator domenlarda, shu jumladan matnli hujjatlar to'plamida ham klasterlash uchun qiziqarli yondashuv bo'lib qolmoqda (5.6-bo'lim). Uning to'liq salohiyatini tushunish faol tadqiqot sohasi bo'lib qolmoqda. Chiziqli algebra bo'yicha yangilashni talab qilmaydigan o'quvchilar 7.1-bo'limni o'tkazib yuborishlari mumkin, garchi 7.1-misol ayniqsa tavsiya etiladi chunki u keyinchalik bobda foydalaniladigan xos qiymatlar xususiyatini ta'kidlaydi. **C** haqiqiy qiymatli yozuvlarga ega M×N matritsa bo'lsin. ***Atama-hujjat matritsasi*** uchun barcha yozuvlar aslida salbiy emas. Matritsaning darajasi - undagi chiziqli mustaqil qatorlar (yoki ustunlar) sonidir. Shunday qilib, daraja **(C) ≤ min {M,N}**. Diagonaldan tashqari barcha yozuvlari nolga teng bo'lgan kvadrat **r×r** matritsa ***diagonal matritsa*** deyiladi; uning darajasi nolga teng bo'lmagan diagonal yozuvlar soniga teng. Agar bunday diagonal matritsaning barcha **r** diagonal yozuvlari 1 ga teng bo'lsa, u **r** ***o'lchamning bir xillik matritsasi*** deb ataladi va **Ir** bilan ifodalanadi. Kvadrat **M×M** matritsasi **C** va barcha nolga teng bo'lmagan vector $C\vec{x} = \lambda\vec{x}$ (7.1) ni qanoatlantiruvchi 1 qiymatlari **C** ning xos qiymatlari deyiladi. λ xos qiymat uchun $N-vector$ $\vec{x}$ qanoatlantiruvchi (7.1) tenglama mos keladigan ***o'ng xos*** vektordir. Eng katta kattalikning xos qiymatiga mos keladigan xos vektor ***bosh xos vektor*** deyiladi. Xuddi shunday tarzda **C** ning **chap xos vektorlari** M-vektorlar y bo'lib, $\vec{y}^T C = \lambda\vec{y}^T$ **C** ning nolga teng bo'lmagan o'ziga xos qiymatlari soni eng ko'p (**C**) darajasiga ega.

Matritsaning xos qiymatlari xarakteristik tenglamani yechish yo'li bilan topiladi, u (7.1) tenglamani $(C - \lambda I_M)\vec{x} = 0$ ko'rinishida qayta yozish orqali olinadi. U holda **C** ning xos qiymatlari $|(C - \lambda I_M)| = 0,\ where\ |S|$ ning yechimlari hisoblanadi. $| = 0$, bu erda |S| kvadrat matritsaning determinantini bildiradi. Tenglama $|(C - \lambda I_M)| = 0$ dagi ***M-tartibli ko'phadli tenglama*** bo'lib, **C** ning xos qiymatlari bo'lgan ko'pi M ildizga ega bo'lishi mumkin. Bu xos qiymatlar **C** ning barcha yozuvlari haqiqiy bo'lsa ham, umuman olganda murakkab bo'lishi mumkin.

Endi quyidagi 7.2-bo'limda yagona qiymat taqsimlanishining markaziy g'oyasini o'rnatish uchun xos qiymatlar va xos vektorlarning yana bir qancha xususiyatlarini ko'rib chiqiladi. Birinchidan, ***matritsa-vektorni ko'paytirish*** va ***xususiy qiymatlar o'rtasidagi munosabati*** ko'rib chiqiladi.

7.1-misol: Matritsani ko'rib chiqing:

$$s = \begin{pmatrix} 30 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Shubhasiz, matritsa 3-darajali va uchta mos keladigan xos vektor bilan 3 ta noldan farqli $l_1 = 30$, $l_2 = 20$ va $l_3 = 1$ xos qiymatga ega.

$$\vec{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \vec{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad va \quad \vec{x}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Xususiy vektorlarning har biri uchun **S** ga ko'paytirish xos vektorni o'ziga xoslik matritsasining ko'paytmasiga ko'paytirgandek ishlaydi.

Karrali har bir xos vektor uchun har xil. Endi $\vec{v} = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}$ kabi ixtiyoriy vektorni ko'rib chiqing. Har doim $\vec{v}$ ni **S** ning uchta xos vektorining chiziqli birikmasi sifatida ifodalashimiz mumkin. Masalan:

$$\vec{v} = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} = 2\vec{x}_1 + 4\vec{x}_2 + 6\vec{x}_3$$

Aytaylik, $\vec{v}$ ni **S** ga ko'paytiramiz:

$$S\vec{v} = S(2\vec{x}_1 + 4\vec{x}_2 + 6\vec{x}_3)$$
$$= 2S\vec{x}_1 + 4S\vec{x}_2 + 6S\vec{x}_3$$
$$= 2\lambda_1\vec{x}_1 + 4\lambda_2\vec{x}_2 + 6\lambda_3\vec{x}_3$$
$$= 60\vec{x}_1 + 80\vec{x}_2 + 6\vec{x}_3$$

7.1-misol shuni ko'rsatadiki, $\vec{v}$ ixtiyoriy vektor bo'lsa-da, S ga ko'paytirish effekti **S** ning xos qiymatlari va xos vektorlari bilan aniqlanadi. Bundan tashqari, (7.3) tenglamadan intuitiv ravishda ko'rinib turibdiki, $S\vec{v}$ mahsuloti nisbatan ta'sir qilmaydi. **S** ning kichik xos qiymatlaridan kelib chiqadigan atamalar, bizning misolimizda $\lambda_3 = 1$, bo'lgani uchun (7.3) tenglamaning o'ng tomonidagi uchinchi hadning hissasi kichik. Haqiqatan ham, agar (7.3) tenglamadagi uchinchi xos vektordan olingan hissani butunlay e'tiborsiz qoldiradigan bo'lsak.

$\lambda_3 = 1$ bo'lsa, u holda $S\vec{v}$ mahsuloti hisoblanadi $\begin{pmatrix} 60 \\ 80 \\ 0 \end{pmatrix}$ to'g'ri mahsulot emas, balki $\begin{pmatrix} 60 \\ 80 \\ 0 \end{pmatrix}$ bu ikki vektor qo'llanilishi mumkin bo'lgan har xil ko'rsatkichlar (masalan, vektor farqining uzunligi) bo'yicha bir-biriga nisbatan yaqin.

Bu kichik xos qiymatlarning (va ularning xos vektorlarining) matritsa-vektor mahsulotiga ta'siri kichik ekanligini ko'rsatadi. 7.2-bo'limda matritsalarning taqsimlanishi va past darajali yaqinlashuvlarni o'rganishda ushbu sezgini ilgari suramiz. Buni amalga oshirishdan oldin, ular uchun alohida qiziqish uyg'otadigan matritsalarning maxsus shakllarining xos vektorlari va xos qiymatlarini tekshiramiz. Simmetrik S matritsa uchun alohida xos qiymatlarga mos keladigan xos vektorlar ortogonaldir. Bundan tashqari, agar S ham haqiqiy, ham simmetrik bo'lsa o'z qiymatlari hammasi haqiqiydir.

7.2-misol: Haqiqiy, simmetrik matritsani ko'rib chiqing

$$S = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

Xarakteristik tenglamadan $|S - \lambda I| = 0$, ularda kvadratik $(2-\lambda)^2 - 1 = 0$, uning yechimlari 3 va 1 xos qiymatlarni beradi. Tegishli xos vektorlar

$\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ va $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ ortogonaldir.

7.1. Matritsalarning taqsimlanilishi

Ushbu bo'limda kvadrat matritsa va uning xos vektorlaridan olingan matritsalar ko'paytmasiga ko'paytirish usullari ko'rib chiqiladi. Bu jarayonni ***matritsa taqsimlanilishi*** deb ataymiz. Ushbu bo'limdagilarga o'xshash matritsalarning parchalanishi 7.3-bo'limda matnni tahlil qilishning asosiy texnikamiz asosini tashkil qiladi, bu yerda kvadrat bo'lmagan muddatli hujjat matritsalarining parchalanishi ko'rib chiqiladi. Ushbu bo'limdagi kvadrat parchalanishlar soddaroq va o'quvchiga bunday parchalanishlar qanday ishlashini tushunishga yordam berish uchun yetarli matematik qat'iylik bilan ishlov berilishi mumkin. 7.2-bo'limdagi murakkabroq dekompozitsiyalarning batafsil matematik hosilasi ushbu kitob doirasidan tashqarida. Kvadrat matritsani maxsus shakldagi uchta matritsaning ko'paytmasiga taqsimlash bo'yicha ikkita teoremani berishdan boshlaymiz. Ulardan birinchisi, 7.1-teorema, kvadrat real qiymatli matritsaning asosiy faktorizatsiyasi uchta omilga beriladi. Ikkinchisi, 7.2-teorema, kvadrat simmetrik matritsalarga taalluqlidir va 7.3-teoremada tasvirlangan yagona qiymat taqsimlanishining asosidir. 7.1 teorema. (Matritsani diagonallashtirish teoremasi) **M** chiziqli mustaqil xos vektorli **S** kvadrat haqiqiy qiymatli **M×M** matritsa bo'lsin. Keyin o'ziga xos parchalanish mavjud

$$S = u\Lambda u^{-1}$$

Bu yerda **U** ustunlari **S** ning xos vektorlari va ^ diagonal matritsa bo'lib, diagonal yozuvlari **S** ning kamayish tartibida xos qiymatlari bo'ladi.

$$\begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \cdots & \\ & & & \lambda_M \end{pmatrix}, \lambda_i \geq \lambda_{i+1}$$

Agar xususiy qiymatlar aniq bo'lsa bu taqsimlanish noyobdir. 7.1 teorema qanday ishlashini tushunish uchun **U** ning ustunlar sifatida **S** ning xos vektorlariga ega ekanligini ta'kidlaymiz.

$$SU = S(\vec{u}_1\vec{u}_2....\vec{u}_M)$$
$$= (\lambda_1\vec{u}_1\ \lambda_2\vec{u}_2....\lambda_M\vec{u}_M)$$
$$= (\vec{u}_1\vec{u}_2....\vec{u}_M)\begin{pmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & ... & \\ & & & \lambda_M\end{pmatrix},$$
$$U = (\vec{u}_1\vec{u}_2....\vec{u}_M)$$

Shunday qilib, ularda $SU = U\Lambda,\ yoki\ \ S = U\Lambda U^{-1}$ mavjud.

Keyinchalik ular simmetrik kvadrat matritsaning xos vektorlaridan olingan matritsalar ko'paytmasiga chambarchas bog'liq parchalanishini aytamiz. Bu ularning matnni tahlil qilish uchun asosiy vositasi, yakka qiymatli dekompozitsiyani ishlab chiqish uchun yo'l ochadi (7.2-bo'lim).

7.2 teorema. (Simmetrik diagonallanish teoremasi) **S** kvadrat, simmetrik haqiqiy qiymatli **M×M** matritsa, **M** chiziqli mustaqil xos vektorlar bo'lsin. Keyin simmetrik diagonal taqsimlanish mavjud:

$$S = Q\Lambda Q^T,$$

bu yerda **Q** ustunlari **S** ning ortogonal va normallashtirilgan (birlik uzunligi, haqiqiy) xos vektorlari va l diagonal matritsa bo'lib, uning yozuvlari **S** ning xos qiymatlari bo'ladi. **Q** ning barcha yozuvlari haqiqiy va ularda $Q^{-1} = Q^T$. mavjud. Ular atama-hujjat matritsalariga past darajali yaqinlashishlarni yaratish uchun ushbu simmetrik diagonal parchalanishga asoslanadi.

Misol. Quyidagi **3 × 3** diagonal matritsaning darajasi qanday?

$$\begin{pmatrix}1 & 1 & 0\\ 0 & 1 & 1\\ 1 & 2 & 1\end{pmatrix}$$

Misol. l = 2 ning xos qiymati ekanligini ko'rsating:

$$C = \begin{pmatrix}6 & -2\\ 4 & 0\end{pmatrix}$$

Tegishli xos vektorni toping.

7.2. Term-hujjat matritsalari va singulyar qiymatlarning taqsimlanishi

Hozirgacha o'rganayotgan taqsimlanish kvadrat matritsalarga tegishli. Biroq, ularni qiziqtiradigan matritsa **M×N** ***atama-hujjat matritsasi*** **C** bo'lib, bu yerda (kamdan-kam uchraydigan tasodifni hisobga olmaganda) M != N. Bundan tashqari, **C** simmetrik bo'lishi ehtimoldan yiroq emas.

Shu maqsadda birinchi navbatda yagona qiymat dekompozitsiyasi deb nomlanuvchi simmetrik diagonal taqsimlanishning kengaytmasini tasvirlaymiz.

Keyin 7.3-bo'limda undan **C** ning taxminiy versiyasini yaratish uchun qanday foydalanish mumkinligini ko'rsatamiz. Singular qiymat taqsimlanishlari asosida yotgan matematikani to'liq ko'rib chiqish ushbu kitobning doirasiga kirmaydi; 7.3-teorema bayonotidan so'ng 7.1.1-bo'limdagi simmetrik diagonal taqsimlanishlar bilan singulyar qiymatlarning taqsimlanishini bog'laymiz.

C berilgan bo'lsa, **U** ustunlari CC^T ning ortogonal xos vektorlari bo'lgan U **M×M** matritsa va **V** ustunlari C^TC ning ortogonal xos vektorlari bo'lgan **N×N** matritsa bo'lsin. C^T orqali matritsaning transpozitsiyasini belgilang **C**. Teorema 7.3. **M×N** matritsasi **C** darajasi bo'lsin. Keyin, $C = U\sum V^T$ ko'rinishidagi **C** ning yagona qiymatli taqsimlanishi (qisqacha SVD) mavjud ,(7.9)

Bu yerda:

Xususiy qiymatlar $\lambda_1,..., \lambda_r of\ CC^T$ ning si C^TC ning xos qiymatlari bilan bir xil;

$\lambda_i \geq \lambda_{i+1}$ uchun $1 \leq i \leq r$, *let* $\sigma_i = \sqrt{\lambda_i}$, bo'lsin. U holda M × N matritsasi $\sum_{ii} = \sigma_i$ *for* $1 \leq i \leq r$, uchun $\sum$ aks holda nolni o'matish orqali tuziladi. **Sigma** qiymatlari **C** ning yagona qiymatlari deb ataladi.

7.3-teoremaning 7.2-teorema bilan bog'liqligini tekshirish ko'rsatma beradi. Ular 7.3-teoremaning umumiy isbotini olishdan ko'ra buni qiladi, bu kitobning doirasidan tashqarida. (7.9) tenglamani uning ko'chirilgan versiyasiga ko'paytirish orqali ularda

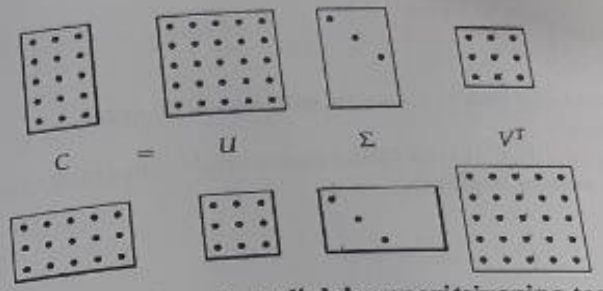
$$CC^T = U\sum V^T \; V\sum U^T = U\sum {}^2 U^T$$

Endi e'tibor bering, (7.10) tenglamada chap tomon kvadrat simmetrik matritsaning haqiqiy qiymatli matritsasi, o'ng tomoni esa 7.2 teoremadagi kabi uning simmetrik diagonal taqsimlanishini ifodalaydi. Chap tomondagi CC^T nimani anglatadi?

Bu har bir **M** shartga mos keladigan qator va ustunli kvadrat matritsadir. Matritsadagi yozuv (i,j) *i*th va *j*th hadlar o'rtasidagi o'zaro bog'liqlik ko'rsatkichi bo'lib, ularning hujjatlarda birgalikda kelishiga asoslanadi. Aniq matematik ma'no **C** ning atama vazniga asoslangan holda tuzilishiga bog'liq.

C atama-hujjat insidans matritsasi bo'lgan holatni ko'rib chiqaylik. Keyin CC^T dagi yozuv (i,j) i va j atamasi uchraydigan hujjatlar soni. SVD ning raqamli qiymatlarini yozishda $\sum$ ni diagonallardagi yagona qiymatlari bilan **r×r** matritsa sifatida ko'rsatish odatiy holdir chunki uning ushbu kichik matritsadan tashqaridagi barcha yozuvlari nolga teng. Shunga ko'ra, **S** ning ushbu o'tkazib yuborilgan qatorlariga mos keladigan **U** ning eng o'ngdagi **M-r** ustunlarini tashlab qo'yish odatiy holder.

Xuddi shunday **V** ning eng o'ngdagi **N−r** ustunlari ham kiritilmagan chunki ular V^T da $\sum$ dagi nollarning **N−r** ustunlariga ko'paytiriladigan qatorlarga mos keladi. SVD ning ushbu yozma shakli ba'zan qisqartirilgan SVD yoki kesilgan SVD deb nomlanadi va uni 7.9-mashqda yana uchratamiz. Bundan buyon bizning raqamli misollarimiz va mashqlarimiz ushbu qisqartirilgan shakldan foydalanadi.



7.1-rasm. **Singular-qiymatli dekompozitsiyaning tasviri**

Ushbu sxematik rasmda (7.9) tasvirlangan ikkita holatni ko'ramiz. Rasmning yuqori yarmida ularda **M>N** bo'lgan **C** matritsasi mavjud. Pastki yarmi **M<N** holatini ko'rsatadi.

7.3-misol: Endi 2-darajali **4×2** matritsaning singulyar-qiymatli taqsimlanishini tasvirlaymiz. Yagona qiymatlar 7.3-misol. Endi 2-darajali **4×2** matritsaning singulyar-qiymatli taqsimlanishini tasvirlaymiz. $\Sigma_{11} = 2.236$ and $\Sigma_{22} = 1$.

Yagona qiymatlar

(18.11)
$$C = \begin{pmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} -0.632 & 0.000 \\ 0.316 & -0.707 \\ -0.316 & -0.707 \\ 0.632 & 0.000 \end{pmatrix} \begin{pmatrix} 2.236 & 0.000 \\ 0.000 & 1.000 \end{pmatrix} \begin{pmatrix} -0.707 & 0.707 \\ -0.707 & -0.707 \end{pmatrix}.$$

7.1.1-bo'limda aniqlangan matritsalarni taqsimlashda bo'lgani kabi, matritsaning yagona qiymat dekompozitsiyasini turli xil algoritmlar yordamida hisoblash mumkin, ularning ko'pchiligi umumiy foydalanish mumkin bo'lgan dasturiy ta'minot bo'lgan. Bularga ko'rsatmalar 7.5-bo'limda keltirilgan.

7.4-misol.

$$C = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

to'plam uchun atama-hujjat insidans matritsasi bo'lsin.

CC^T birgalikda yuzaga keladigan matritsani hisoblang. Agar **C** atama-hujjat insidans matritsasi bo'lsa, CC^T diagonal yozuvlari qanday talqin qilinadi?

7.5-misol.

(7.12) tenglamadagi matritsaning SVD ekanligini tekshiring:

(18.13) $U = \begin{pmatrix} -0.816 & 0.000 \\ -0.408 & -0.707 \\ -0.408 & 0.707 \end{pmatrix}$, $\Sigma = \begin{pmatrix} 1.732 & 0.000 \\ 0.000 & 1.000 \end{pmatrix}$ and $V^T = \begin{pmatrix} -0.707 & -0.707 \\ 0.707 & -0.707 \end{pmatrix}$.

7.3- Teoremadagi barcha xususiyatlarni tekshirish orqali.

7.6-misol.

Faraz qilaylik, **C** ikkilik atama-hujjat insidans matritsasi bo'lsin. C^T C yozuvlari nimani anglatadi?

$$C = \begin{pmatrix} 0 & 2 & 1 \\ 0 & 3 & 0 \\ 2 & 1 & 0 \end{pmatrix}$$

yozuvlari muddatli chastotalar bo'lgan atama-hujjat matritsasi bo'lsin. Shunday qilib, 1 atama 2-hujjatda 2 marta va 3-hujjatda bir marta uchraydi. CC^T ni hisoblang. Uning yozuvlari bitta hujjatda ikkita atama eng ko'p uchraydigan joyda eng katta bo'lishiga e'tibor bering.

7.3. Past darajali taxminlar

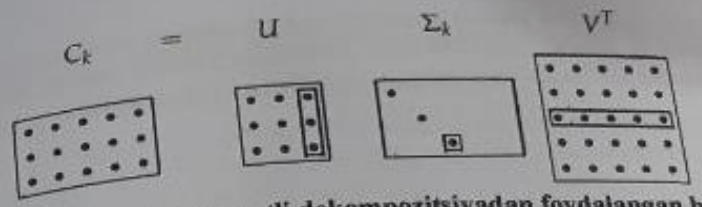
Matritsaga yaqinlashish muammosini aytib o'tamiz, bu birinchi navbatda ma'lumot olish bilan bog'liq emas.

Ushbu matritsa muammosining yechimini singulyar qiymatli dekompozitsiyalardan foydalangan holda tasvirlaymiz, so'ngra uning ma'lumot olish uchun qo'llanilishini ishlab chiqamiz.

M × N matritsasi **C** va musbat butun **k** sonini hisobga olib, ular eng ko'p **k** darajali **M×N** C_k matritsasini topmoqchimiz, shuning uchun $X = C - C_k$ matritsalar farqining Frobenius normasini minimallashtiramiz:

$$\| X \|_F = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{N} X^2_{ij}}$$

Shunday qilib, **X** ning Frobenius normasi C_k va **C** o'rtasidagi nomuvofiqlikni o'lchaydi. Bizning maqsadimiz C_k ni eng ko'p **k** darajaga ega bo'lishini cheklab, bu tafovutni minimallashtiradigan C_k matritsasini topishdir. Agar **r** **C** darajasi bo'lsa, aniq C_r = **C** va nomuvofiqlikning Frobenius normasi bu holda nolga teng. Agar **k** **r** dan ancha kichik bo'lsa, ular C_k ga past darajali yaqinlashish deb murojaat qiladi. Singular qiymat dekompozitsiyasidan past darajali matritsaga yaqinlashish masalasini hal qilish uchun foydalanish mumkin. Keyin undan atama-hujjat matritsalarini yaqinlashtirish uchun ilovani olamiz. Buning uchun quyidagi uch bosqichli protsedurani chaqiramiz:



7.2-rasm. **Singular-qiymatli dekompozitsiyadan foydalangan holda past darajali yaqinlashuvning tasviri**

Chiziqli katakchalar eng kichik yagona qiymatlarni "nollash" natijasida ta'sirlangan matritsa yozuvlarini ko'rsatadi.

1. Berilgan C ning SVD ni (7.9) ko'rsatilgan shaklda tuzing. U quyidagicha hisoblanadi: $C = U \sum V^T$

2. Σ diagonalidagi $\sum_k$ eng kichik singulyar qiymatlarni nolga almashtirish natijasida hosil bo'lgan $\sum_k$ matritsasini dan $\sum$ chiqaring.

3. Hisoblang va $C_k = U \sum_k V^T$ ni **C** darajasiga yaqinlashuv sifatida chiqaring. Σ_k darajasi eng ko'p **k**: bu $\mathbf{C_k}$ ning eng ko'pi **k** nolga teng bo'lmagan qiymatlarga ega ekanligidan kelib chiqadi. Keyinchalik, 7.1-misolning sezgisini eslaymiz: kichik o'z qiymatlarining matritsa mahsulotlariga ta'siri kichik. Shunday qilib, ushbu kichik o'z qiymatlarini nolga almashtirish mahsulotni sezilarli darajada o'zgartirmasligi va uni **C** ga "yaqin" qoldirishi mantiqiy ko'rinadi.

Ekkart va ***Yangga*** bog'liq bo'lgan quyidagi teorema shuni ko'rsatadiki, bu protsedura aslida ***Frobenius xatosi*** mumkin bo'lgan eng kichik **k** darajali matritsani beradi.

7.4- teorema:

$$\min_{Z \mid rank(Z) = k} \| C - F \|_F = \| C - C_k \|_F = \sigma_{k+1}$$

Yagona qiymatlar $\sigma_1 \geq \sigma_2 \geq \ldots$, kamayish tartibida ekanligini eslab, 7.4-teoremadan bilib olamizki, C_k **C** ga eng yaxshi **rank-k** yaqinlashuvi bo'lib, xatolikka yo'l qo'yadi (Frobenius normasi $C - C_k$ bilan o'lchanadi). σ_{k+1} gateng. Shunday qilib, k qanchalik katta bo'lsa, bu xatolik shunchalik kichik bo'ladi (va ayniqsa, k = r uchun xatolik $\sum$-$\sum$ dan nolga teng, shartr<M,N, keyin $\sigma_{r+1} = 0$ va shuning uchun $C_r = C$. $\sum$

dagi eng kichikr – k yagona qiymatlarni kesish jarayoni nima uchun past xatoning k darajali yaqinlashuvini yaratishga yordam berishi haqida qo'shimcha ma'lumot olish uchun C_k shaklini ko'rib chiqamiz:

$$C_k = U \sum_k V^T$$

$$= U \begin{pmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \sigma_k & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots \end{pmatrix}$$

$$= \sum_{i=1}^{k} \sigma_i \vec{u}_i \vec{v}_i^T$$

bu yerda $\vec{u}_i$ va $\vec{v}_i$ mos ravishda **U** va **V** ning i-ustunlari. Shunday qilib, $\vec{u}_i \vec{v}_i^T$ 1-darajali matritsadir, shuning uchun hozirgina C_k ni har bir alohida qiymat bilan tortilgan k-darajali matritsalarning yig'indisi sifatida ifodaladik. **i** ortishi bilan 1-darajali matritsaning hissasi $\vec{u}_i \vec{v}_i^T$ singular qiymatlarning qisqarishi ketma-ketligi bilan og'irlik qiladi σ_i.

7.8-misol. 7.13-misoldagi kabi SVD dan foydalanib, 7.12-misoldagi **C** matritsasiga C_2 darajali yaqinlashuvni hisoblang. Ushbu yaqinlashish xatosining Frobenius normasi nima?

7.9-misol.

Endi 7.8-misoldagi hisoblashni ko'rib chiqing. 7.2-rasmdagi sxema bo'yicha e'tibor bering, 1-darajali yaqinlashish uchun σ_1 skalyardir. **U** ning birinchi ustunini $\mathbf{U}_1$ va **V** ning birinchi ustunini $\mathbf{V}_1$ bilan belgilang. **C** ga 1-darajali yaqinlashuvni $U_1 \sigma_1 V_1^T = \sigma_1 U_1 V_1^T$ shaklida yozish mumkinligini ko'rsating.

7.10-misol.

7.9-misolni **k** darajali yaqinlashuvlar uchun umumlashtirish mumkin: u_k' va v_k' mos ravishda **U** va **V** ning faqat birinchi **k** ustunlarini saqlab qolgan holda hosil qilingan "qisqartirilgan" matritsalarni bildiradi. Shunday qilib, u_k' $M \times k$ matritsa, $v_k'^T$ esa $k \times N$ matritsadir. U holda, bizda

$$C_k = U_k' \sum_k^1 V_k'^T$$

bu yerda $\sum_k^1$ ning yagona qiymatlari s_1 ga ega bo'lgan $\sum_k$ ning kvadrat **k×k** submatritsasidir. Diagonalda $\sum_k$ $\sigma_1, \dots \sigma_k$ (7.20) dan foydalanishning asosiy afzalligi **U** va **V** dagi nollarning ko'plab ortiqcha ustunlarini yo'q qilish va shu bilan past darajali yaqinlashishga ta'sir qilmaydigan ustunlar bilan ko'paytirishni aniq yo'q qilishdir. **SVD** ning ushbu versiyasi ba'zan ***qisqartirilgan SVD*** yoki ***kesilgan SVD*** deb nomlanadi va past darajali yaqinlashuvni hisoblash uchun hisoblash jihatidan soddaroq tasvirlanadi.

7.4. Yashirin semantik indekslash

Endi SVD yordamida **C** atama-hujjat matritsasini quyi darajadagi birga yaqinlashtirishni muhokama qilamiz. **C** ga past darajali yaqinlashish to'plamdagi har bir hujjat uchun yangi taqdimotni beradi. Ushbu past darajali vakillikka ham so'rovlar yuboramiz, bu ularga ushbu past darajadagi taqdimotda so'rov hujjatlari o'xshashlik ballarini hisoblash imkonini beradi. Bu jarayon yashirin semantik indekslash (umuman qisqartirilgan ***LSI***) deb nomlanadi. Lekin birinchi navbatda, bunday yaqinlashishni qo'llab quvvatlaymiz. Birinchi qismning 6.3-bo'limda kiritilgan hujjatlar va so'rovlarning vektor maydoni tasvirini eslang. Ushbu vektor fazosining tasviri bir qator afzalliklarga ega, shu jumladan so'rovlar va hujjatlarni vektor sifatida bir xilda ko'rib chiqish, kosinus o'xshashligi asosida induktsiya qilingan ballni hisoblash, turli atamalarni turlicha baholash qobiliyati va uni klasterlash va tasniflash kabi ilovalarga hujjatlarni qidirishdan tashqari kengaytirish. Vektor fazosining tasviri tabiiy tillarda yuzaga keladigan ikkita klassik muammolarni hal qila olmasligidan aziyat chekadi: **sinonimiya** va ***polisemiya***. ***Sinonimiya*** ikki xil so'z (aytaylik, avtomobil va avtomobillar) bir xil ma'noga ega bo'lgan holatni anglatadi. Chunki vektor fazosining tasviri avtomobil va avtomobil kabi sinonimik atamalar o'rtasidagi munosabatni aks ettira olmaydi - vektor fazodagi har bir alohida o'lchamga ko'ra. Demak, so'rov $\vec{q}$ (aytaylik, avtomobil) va avtomobil va avtomobilni o'z ichiga olgan hujjat $\vec{d}$ hisoblangan o'xshashlik $\vec{q}\,\vec{d}$ foydalanuvchi qabul qiladigan haqiqiy o'xshashlikni kam baholaydi. Boshqa tomondan, ko'p ma'nolilik a zaryad kabi atama bir nechta ma'noga ega, shuning uchun hisoblangan

o'xshashlik $\vec{q}\,\vec{d}$ foydalanuvchi sezadigan o'xshashlikni oshirib yuboradi. Terimlarning latent semantik assotsiatsiyasini aniqlash va bu muammolarni hal qilish uchun atamalarning birgalikda kelishidan foydalana olamizmi?

Hatto oddiy o'lchamdagi to'plam uchun ham, C atama-hujjat matritsasi, ehtimol, bir necha o'n minglab qator va ustunlarga, shuningdek, o'n minglab darajaga ega bo'lishi mumkin. Yashirin semantik indekslashda (ba'zan latent semantik tahlil (LSA) deb ataladi) ular SVD dan hujjat matritsasi atamasining dastlabki darajasidan ancha kichik bo'lgan k qiymati uchun past darajali C_k yaqinlashuvini yaratish uchun foydalaniladi. C. Ushbu bo'limda keyinroq keltirilgan eksperimental ishda, k odatda past yuzliklarda bo'lish uchun tanlangan. Shunday qilib, ular har bir qatorni/ustunni (mos ravishda atama/hujjatga mos keladigan) k o'lchamli fazoga joylashtiramiz; bu fazo CC^T va C^TC ning k asosiy xos vektorlari (eng katta xos qiymatlarga mos) bilan belgilanadi. E'tibor bering, C_k matritsasining o'zi k dan qat'i nazar, baribir $M \times N$ matritsa hisoblanadi. Keyinchalik, ular vektorlar o'rtasidagi o'xshashlikni hisoblash uchun, ular asl tasvirni qilganimizdek, yangi k o'lchovli LSI tasviridan foydalaniladi. So'rov vektori $\vec{q}$ o'zgartirish orqali uning LSI fazosida ko'rinishiga keltiriladi.

$$\vec{q}_k = \sum_k^{-1} U_k^T \vec{q}$$

Endi ular so'rov va hujjat, ikkita hujjat yoki ikkita atama o'rtasidagi o'xshashlikni hisoblash uchun 6.3.1-bo'lim (120-bet)dagi kabi kosinus o'xshashliklaridan foydalanishimiz mumkin. E'tibor bering, (18.21) tenglama hech qanday tarzda $\vec{q}$ so'rov bo'lishiga bog'liq emas; bu shunchaki atamalar fazosida vektor. Bu shuni anglatadiki, agar ularda hujjatlar to'plamining LSI ko'rinishi mavjud bo'lsa, to'plamda bo'lmagan yangi hujjat

$$\vec{q}_k = \sum_k^{-1} U_k^T \vec{q}$$

tenglama yordamida ushbu ko'rinishga "buklanishi" mumkin. Bu ularga LSI taqdimotiga hujjatlarni bosqichma-bosqich qo'shish imkonini beradi. Albatta, bunday qo'shimcha qo'shimcha yangi qo'shilgan hujjatlarning birgalikdagi hodisalarini qamrab olmaydi (va hatto ulardagi har qanday

yangi shartlarni e'tiborsiz qoldiradi). Shunday qilib, LSI taqdimotining sifati ko'proq hujjatlar qo'shilishi bilan yomonlashadi va oxir-oqibat LSI taqdimotini qayta hisoblashni talab qiladi. C_k ni C ga yaqinlashtirishning sodiqligi ularni kosinus o'xshashliklarining nisbiy qiymatlari saqlanib qolishiga umid qilishga olib keladi: agar so'rov dastlabki fazoda hujjatga yaqin bo'lsa, uch o'lchovli fazoda nisbatan yaqin bo'lib qoladi. Lekin buning o'zi unchalik qiziq emas, ayniqsa siyrak so'rov vektori $\vec{q}$ past o'lchamli fazoda zich so'rov vektoriga $\vec{q}$, aylanayotganini hisobga olsak. Bu $\vec{q}$ ni o'zining mahalliy ko'rinishida qayta ishlash xarajatlari bilan solishtirganda sezilarli hisoblash narxiga ega.

Misol: C = atama-hujjat matritsasini ko'rib chiqish.

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

Uning yakka qiymatli dekompozitsiyasi quyidagi uchta matritsaning mahsulotidir. Avval ularda U bor, bu misolda:

	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
voyage	−0.70	0.35	0.15	−0.58	0.16
trip	−0.26	0.65	−0.41	0.58	−0.09

SVD atama-hujjat matritsasiga qo'llanilganda, U SVD atama matritsasi sifatida tanilgan. Yagona qiymatlar $\sum$ =

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

Nihoyat, ularda V^T ga ega bo'lib, atama-hujjat matritsasi kontekstida SVD hujjat matritsasi sifatida tanilgan:

	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

By "zeroing out" all but the two largest singular values of Σ, we obtain $\Sigma_2 =$

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

From this, we compute $C_2 =$

	d_1	d_2	d_3	d_4	d_5	d_6
1	−1.62	−0.60	−0.44	−0.97	−0.70	−0.26
2	−0.46	−0.84	−0.30	1.00	0.35	0.65
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

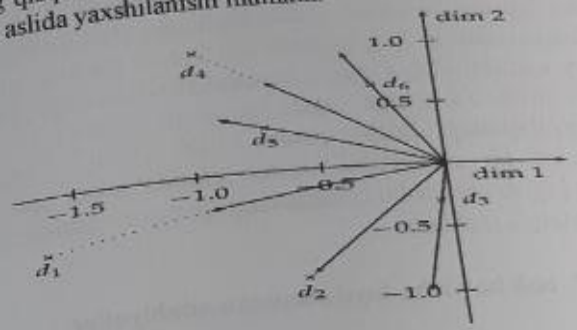
E'tibor bering, past darajali yaqinlashish, dastlabki C matritsasidan farqli o'laroq, salbiy yozuvlarga ega bo'lishi mumkin. Yuqoridagi misolda c_i va $\sum_i$ ni tekshirish shuni ko'rsatadiki, bu matritsalarning har birining oxirgi 3 qatori butunlay nol bilan to'ldirilgan. Bu (7.18) tenglamadagi SVD mahsuloti $U\sum V^T$ ni $\sum_i$ va V^T ko'rinishlarida faqat ikkita qator bilan amalga oshirish mumkinligini ko'rsatadi. Bu matritsalarni ularning kesilgan $\sum_2$ va $(V')^T$ versiyalari bilan almashtirish mumkin. Masalan, bu misoldagi kesilgan SVD hujjat matritsasi $(V')^T$:

	d_1	d_2	d_3	d_4	d_5	d_6
1	−1.62	−0.60	−0.44	−0.97	−0.70	−0.26
2	−0.46	−0.84	−0.30	1.00	0.35	0.65

7.3-rasmda. $(V')^T$ **dagi hujjatlar ikki o'lchovli tasvirlangan**

Shuni ham yodda tutingki, C_2 C ga nisbatan zichroqdir. Umuman olganda, **C** ning past darajali yaqinlashuvini cheklangan optimallashtirish muammosi sifatida ko'rishimiz mumkin: C_k eng ko'p **k** darajasiga ega bo'lgan cheklovni hisobga olgan holda quyidagi ko'rsatkichni izlaymiz.

$C - C_k$ xatosi uchun kam ***Frobenius normasi*** bilan **C** ni o'z ichiga olgan atamalar va hujjatlar. Shartlarni/hujjatlarni **k** o'lchovli bo'shliqqa siqib chiqarishga majbur bo'lganda, SVD o'xshash hodisalar bilan atamalarni birlashtirishi kerak. Shunday qilib, bu sezgi shuni ko'rsatadiki, o'lchamning qisqarishidan faqat olish sifati juda ko'p zarar ko'rmasligi kerak, balki aslida yaxshilanishi mumkin.



7.3-rasm. **Hujjatlarning ikki o'lchamga qisqartirilgan grafigi**

Dumais (1995) SVD ni hisoblash uchun keng tarqalgan *Lanczos* algoritmidan foydalangan holda TREC hujjatlari va vazifalari bo'yicha LSI bilan tajriba o'tkazdgan. 1990-yillarning boshlarida o'n minglab hujjatlar bo'yicha LSI hisob-kitoblari bir mashinada taxminan bir kun davom etgan. Ushbu tajribalarda o'rtacha TREC ishtirokchisining aniqligi yoki undan yuqori aniqlikka erishildi. Taxminan 20% TREC mavzularida ularning tizimi eng yaxshisi bo'lgan va xabarlarga ko'ra, taxminan 350 o'lchamdagi LSI uchun standart vektor bo'shliqlaridan o'rtacha bir oz yaxshiroq. Bu yerda LSI bo'yicha ba'zi xulosalar birinchi bo'lib ularning ishlari tomonidan taklif qilingan va keyinchalik ko'plab boshqa tajribalar bilan tasdiqlangan.

- SVD ning hisoblash qiymati sezilarli. Ushbu maqolani yozish paytida bir milliondan ortiq hujjat bilan muvaffaqiyatli tajriba o'tkazmaganligini bilamiz. Bu LSI ning keng tarqalishiga eng katta to'siq bo'ldi. Ushbu to'siqni hal qilish usullaridan biri to'plamdagi hujjatlarning tasodifiy tanlab olingan kichik to'plamida LSI taqdimotini qurishdir, shundan so'ng qolgan hujjatlar (7.21) tenglamada batafsil tavsiflanganidek bajriladi.
- **k** ni kamaytirsak, eslash kutilganidek ortib boradi.

• Ajablanarlisi shundaki, past yuzliklarda **k** qiymati ba'zi so'rovlar ko'rsatkichlarida aniqlikni oshirishi mumkin. Bu shuni ko'rsatadiki, **k** ning mos qiymati uchun LSI sinonimiyaning Ba'zi muammolarini hal qiladi.

• LSI so'rovlar va hujjatlar o'rtasida bir-biriga zid bo'lmagan ilovalarda eng yaxshi ishlaydi. Tajribalar, shuningdek, LSI an'anaviy indekslar va ballarni hisoblash samaradorligiga mos kelmagan ba'zi rejimlarni hujjatlashtirdi. Eng muhimi LSI vektor fazosini qidirishning ikkita asosiy kamchiliklarga ega: inkorlarni ifodalashning yaxshi usuli yo'q (nemis tilini o'z ichiga olgan hujjatlarni toping) va mantiqiy shartlarni qo'llashning hech qanday usuli yo'q. Qisqartirilgan bo'shliqning har bir o'lchamini klaster sifatida talqin qilish orqali klasterlash va hujjatning ushbu o'lchamdagi qiymatini ushbu klasterdagi kasr a'zoligi sifatida izohlash.

7- bob bo'yicha foydalanilgan adabiyotlar

Anh, Vo Ngoc, Owen de Kretser, and Alistair Moffat.
2001.
Vector-space ranking with effective early termination.
In *Proc. SIGIR*, pp. 35-42. ACM Press.

Anh, Vo Ngoc, and Alistair Moffat.
2005.
Inverted index compression using word-aligned binary codes.
IR 8 (1): 151-166.
DOI: *dx.doi.org/10.1023/B:INRT.0000048490.99518.5c*.

Anh, Vo Ngoc, and Alistair Moffat.
2006a.
Improved word-aligned binary compression for text indexing.
IEEE Transactions on Knowledge and Data Engineering 18 (6): 857-861.

Anh, Vo Ngoc, and Alistair Moffat.
2006b.
Pruned query evaluation using pre-computed impacts.
In *Proc. SIGIR*, pp. 372-379. ACM Press.
DOI: *doi.acm.org/10.1145/1148170.1148235*.

Anh, Vo Ngoc, and Alistair Moffat.

2006c.
Structured index organizations for high-throughput text querying.
In *Proc. SPIRE*, pp. 304-315. Springer.

Apté, Chidanand, Fred Damerau, and Sholom M. Weiss.
1994.
Automated learning of decision rules for text categorization.
TOIS 12 (1): 233-251.

Carmel, David, Doron Cohen, Ronald Fagin, Eitan Farchi, Michael Herscovici, Yoelle S. Maarek, and Aya Soffer.
2001.
Static index pruning for information retrieval systems.
In *Proc. SIGIR*, pp. 43-50. ACM Press.
DOI: *doi.acm.org/10.1145/383952.383958*.

Carmel, David, Yoelle S. Maarek, Matan Mandelbrod, Yosi Mass, and Aya Soffer.
2003.
Searching XML documents via XML fragments.
In *Proc. SIGIR*, pp. 151-158. ACM Press.
DOI: *doi.acm.org/10.1145/860435.860464*.

Caruana, Rich, and Alexandru Niculescu-Mizil.
2006.
An empirical comparison of supervised learning algorithms.
In *Proc. ICML*.

Castro, R. M., M. J. Coates, and R. D. Nowak.
2004.
Likelihood based hierarchical clustering.
IEEE Transactions in Signal Processing 52 (8): 2308-2321.

Cavnar, William B., and John M. Trenkle.
1994.
N-gram-based text categorization.
In *Proc. SDAIR*, pp. 161-175.

Chakrabarti, Soumen.
2002.
Mining the Web: Analysis of Hypertext and Semi Structured Data.
Morgan Kaufmann.

7- bob bo‘yicha nazariy va amaliy test savollari

1. B+ daraxtida qidiruv kaliti qiymati 12 bayt uzunlikda, blok hajmi 1024 bayt va blok ko‘rsatkichi 6 bayt bo‘lsa, daraxtning har bir barg bo‘lmagan tuguniga joylashtirish mumkin bo‘lgan kalitlarning maksimal soni shunday bo‘ladi. _____ .

A) 56
B) 58
C) 54
D) 57

2. Mijozlarni tavsiflovchi jadvalni ko‘rib chiqing:
Mijozlar (xo‘jayin nomi, ismi, jinsi, reytingi)*Reyting* qiymati jinsi saqlangan qator 1 5 va faqat ikki qiymatlari (erkak va ayol) biror aniq son. “Qancha erkak mijozning reytingi 5” degan so‘rovni ko‘rib chiqing? So‘rovga mos keladigan eng yaxshi indekslash mexanizmi qanday?

A) Bit-xaritalashtirilgan xeshlash
B) B + daraxt
C) Kengaytiriladigan xeshlash
D) Chiziqli xeshlash

3. Quyidagi so‘rovni ko‘rib chiqing:
SELECT E.eno, COUNT(*)
Xodimlardan E
E.eno tomonidan gurux
Agar *eno*-da indeks mavjud bo‘lsa, so‘rovga faqat agar indeksini skanerlash orqali javob berish mumkin

A) indeks xesh yoki B + daraxti va klasterli yoki klastersiz bo‘lishi mumkin
B) indeks faqat xesh va klasterlangan
C) indeks faqat B + daraxt va klasterli
D) indeks xesh yoki B + daraxti va klasterli bo‘lishi mumkin

4. Qor parchalari sxemasi bilan bog‘liq bo‘lgan quyidagilardan qaysi biri to‘g‘ri?

A) O‘lchov jadvallari normallashtirilgan
B) Har bir o‘lchov bitta o‘lchovli jadval bilan ifodalanadi
C) Davom ettirish ishlari kamroq
D) Bu yulduz sxemasining kengaytmasi emas

5. Trigger bu – nima?

A) Ma’lumotlar bazasini o‘zgartirishning yon ta’siri sifatida tizim tomonidan avtomatik ravishda bajariladigan bayonot
B) Har qanday DBMSni ishga tushirish imkonini beruvchi bayonot
C) Ilova dasturini disk raskadrovka qilishda foydalanuvchi tomonidan bajariladigan bayonot
D) Tizim ma’lumotlar bazasi foydalanuvchisining haqiqiyligini tekshirish sharti

6. B+ daraxtidagi barg tugunining tartibi u ushlab turishi mumkin bo‘lgan juftliklarning maksimal soni (qiymat, ma’lumotlarni qayd qilish ko‘rsatkichi). Blok hajmi 1K bayt, ma’lumotlarni yozib olish ko‘rsatkichi 7 bayt, qiymat maydoni 9 bayt va blok ko‘rsatkichi 6 bayt uzunlikda ekanligini hisobga olsak, barg tugunining tartibi qanday?

A) 63
B) 64
C) 67
D) 68

7. Berilgan blokda 3 ta yozuv yoki 10 ta asosiy ko‘rsatkich bo‘lishi mumkin. Ma’lumotlar bazasi n ta yozuvni o‘z ichiga oladi, keyin ma’lumotlar faylini va zich indeksni saqlash uchun qancha blok kerak?

A) 13n/30
B) n/3
C) n/10
D) n/30

8. Quyida berilgan ma’lumotlar asosida B+ daraxtining barg(p_{barg}) va bargsiz(p) tugunlari tartibini hisoblang. Qidiruv kaliti maydoni = 12 bayt Yozuv ko‘rsatkichi = 10 bayt Blok ko‘rsatkichi = 8 bayt Blok hajmi = 1 KB.

A) $p_{barg} = 46$ & $p = 50$
B) $p_{barg} = 47$ & $p = 52$
C) $p_{barg} = 51$ & $p = 46$
D) $p_{barg} = 52$ & $p = 47$

9. Fayl kalitini yozuv joyiga aylantiruvchi formula bilan aniqlangan yozuvning jismoniy joylashuvi nima?

A) Xeshlangan fayl
B) B-Tree fayli
C) Indekslangan fayl
D) Ketma-ket fayl

10. O'rnatilgan ko'rsatgich taqdim etadi _____.
A) Ikkinchi darajali kirish yo'li
B) Jismoniy yozuvli kaliti
C) Teskari indeks
D) Asosiy kalit

VIII BOB. VEB-QIDIRUV ASOSLARI

Ushbu va keyingi ikki bobda veb-qidiruv tizimlari ko'rib chiqiladi. 8.1–8.4-bo'limlar o'quvchiga Internetni xaotik, tez o'zgaruvchan va (ma'lumot olish nuqtai nazaridan) ushbu kitobda hozirgacha o'rganilgan "an'anaviy" to'plamlardan juda farq qiladigan kuchlarni baholashga yordam berish uchun ba'zi ma'lumotlar va tarixni taqdim etadi. 8.5–8.6 bo'limlar veb-qidiruv tizimlari tomonidan indekslangan hujjatlar sonini baholash va veb-indekslardagi ikki nusxadagi hujjatlarni o'chib ketishi bilan bog'liq muammolar ko'rib chiqiladi. Ushbu ikki oxirgi bo'lim keyingi ikki bob uchun asosiy ma'lumotlar bo'lib xizmat qiladi.

8.1. Ma'lumot va uning tarixi

Internet ko'p jihatdan misli ko'rilmagan, miqyosi bo'yicha misli ko'rilmagan ma'lumotlar manbasi hisoblanadi. Uni yaratishda deyarli to'liq muvofiqlashtirishning yo'qligi va uning ishtirokchilarining kelib chiqishi va motivlarining xilma-xildir. Bu "an'anaviy" hujjatlarni qidirishdan ko'ra ancha qiyinroq. 1940-yillarda ***Vannevar Bush*** tomonidan o'ylab topilgan va birinchi marta 1970-yillarda ishchi tizimlarda amalga oshirilgan gipermatn ixtirosi Butunjahon Internet tarmog'i shakllanishidan ancha oldin sodir bo'ladi (Web 1990-yillarda). Internetdan foydalanish juda katta o'sishni ko'rsatdiki, u hozirda oddiy, ochiq mijoz-server dizayniga tayanib, insoniyatning yaxshi qismini ishtirokchi sifatida da'vo qilmoqda: (1) server mijoz bilan protokol (http yoki gipermatn) orqali bog'lanadi. HTML (gipermatn belgilash tili uchun) deb nomlangan oddiy belgilash tilida kodlangan turli xil foydali yuklarni (matn, tasvirlar va vaqt o'tishi bilan - audio va video fayllar kabi yanada boyroq vositalar) asinxron ravishda olib yuradigan engil va sodda bo'lgan uzatish protokoli); (2) mijoz - odatda brauzer, grafik foydalanuvchi muhitidagi dastur - tushunmaydigan narsani e'tiborsiz qoldirishi mumkin. Ko'rinishidan zararsiz ko'rinadigan ushbu xususiyatlarning har biri Internetning rivojlanishiga katta hissa qo'shgan, shuning uchun ularni batafsilroq ko'rib chiqish maqsadga muvofiqdir.

Asosiy operatsiya quyidagicha: mijoz (masalan, brauzer) veb-serverga http so'rovini yuboradi. Brauzer `http://www.stanford.edu/home/atoz/contact.html` kabi URL manzilini (Universal Resource Locator uchun) belgilaydi.

Ushbu URL misolida http qatori ma'lumotlarni uzatish uchun ishlatiladigan protokolga ishora qiladi www.stanford.edu qatori domen sifatida tanilgan va veb-sahifalar ierarxiyasining ildizini bildiradi (odatda veb-server asosidagi fayl tizimi ierarxiyasini aks ettiradi). Ushbu misolda /home/atoz/contact.html ushbu ierarxiyadagi contact.html faylga ega bo'lgan yo'l bo'lib, bu so'rovga javoban www.stanford.edu veb-server tomonidan qaytariladigan ma'lumotlarni o'z ichiga oladi. HTML-kodlangan contact.html fayli giperhavolalar va tarkibni (bu holda, Stenford universiteti uchun aloqa ma'lumotlari), shuningdek ushbu kontentni brauzerda ko'rsatish uchun formatlash qoidalarini o'z ichiga oladi. Shunday qilib, http so'rovi ularga kontentni olish imkonini beradi. Hujjatlarni skanerlash va indekslashda foydali bo'lishi mumkin bo'lgan fayl sahifa deyiladi.

Birinchi brauzerlarning dizaynerlari URL mazmunidagi HTML belgilash teglarini ko'rishni osonlashtirdi. Ushbu oddiy qulaylik yangi foydalanuvchilarga keng ko'lamli ta'lim yoki tajribasiz o'zlarining HTML kontentini yaratishga imkon berdi. Balki o'zlariga yoqqan misol mazmunidan bilib oladi. Shunday qilib, brauzerlarning ikkinchi xususiyati veb-kontentni yaratish va ulardan foydalanishning tez tarqalishini qo'llab-quvvatladi. Brauzerlar tushunmagan narsalarini e'tiborsiz qoldiradi. Bu HTMLning ko'plab mos kelmaydigan dialektlarini yaratishga olib kelmadi. U targ'ib qilgan narsa havaskor kontent yaratuvchilari edi.

Oddiy sintaksis xatosi "tizimni buzadi" deb qo'rqmasdan o'zlarining yangi yaratilgan veb-sahifalari bilan erkin tajriba o'tkazishlari va ulardan o'rganishlari mumkin edi. Internetda nashr qilish bir nechta o'qitilgan dasturchilar bilan chegaralanib qolmagan, balki o'nlab va oxir-oqibat yuzlab million shaxslar uchun ochiq bo'lgan ommaviy faoliyatga aylandi. Aksariyat foydalanuvchilar va ko'pgina ma'lumotlarga bo'lgan ehtiyojlar uchun Internet tezda kamdan-kam uchraydigan kasalliklardan tortib metro jadvallarigacha bo'lgan barcha narsalar haqida ma'lumot berish va iste'mol qilishning eng yaxshi usuliga aylandi.

Ma'lumotni Internetda ommaviy nashr qilish, agar bu ma'lumotlar boyligi boshqa foydalanuvchilar tomonidan topilmasa va iste'mol qilinmasa, foydasizdir. Veb-ma'lumotni "kashf qilinadigan" qilishning dastlabki urinishlari ikkita keng toifaga bo'lingan: (1) ***Altavista***, ***Excite*** va ***Infoseek*** kabi to'liq matnli qidiruv tizimlari va (2) ***Yahoo!*** kabi toifalardagi veb-sahifalar bilan to'ldirilgan taksonomiyalar! Birinchisi

foydalanuvchiga oldingi boblarda keltirilganlarga asoslangan teskari indekslar va tartiblash mexanizmlari tomonidan qo'llab-quvvatlanadigan kalit so'zlarni qidirish interfeysini taqdim etdi. Ikkinchisi foydalanuvchiga toifa belgilarining ierarxik daraxtini ko'rib chiqish imkonini berdi. Bu dastlab veb-sahifalarni topish uchun qulay va intuitiv metafora bo'lsa-da, uning bir qator kamchiliklari bor: birinchidan, veb-sahifalarni ***taksonomiya daraxti*** tugunlariga to'g'ri tasniflash ko'pincha qo'lda tahrirlash jarayoni bo'lib, uni masshtablash qiyin. Internetning o'lchamini aytish mumkinki, taksonomiyada faqat "yuqori sifatli" veb-sahifalarga ega bo'lishimiz kerak, har bir toifa uchun faqat eng yaxshi veb-sahifalar joylashadi.

Biroq, ularni kashf qilish va ularni aniq va izchil ravishda taksonomiyaga tasniflash insonning katta kuchini talab qiladi. Bundan tashqari, foydalanuvchi taksonomiya daraxti tugunlariga tasniflangan veb-sahifalarni samarali kashf etishi uchun foydalanuvchining ma'lum bir mavzu bo'yicha qanday kichik daraxt(lar)ni izlashi haqidagi fikri tasnifni amalga oshiruvchi muharrirlarnikiga mos kelishi kerak. Taksonomiya hajmi oshgani sayin, bu tezda qiyinlashadiuo Yahoo! Taksonomiya daraxti juda erta 1000 ta aniq tugunlardan oshib ketdi. Ushbu qiyinchiliklarni hisobga olgan holda, vaqt o'tishi bilan taksonomiyalarning mashhurligi pasayib ketdi, garchi variantlar (masalan, About.com va Open Directory loyihasi kabi) har bir toifa uchun veb-sahifalarni to'playdigan va izohlaydigan mavzu bo'yicha mutaxassislar tomonidan paydo bo'lgan bo'lsa ham. Veb-qidiruv mexanizmlarining birinchi avlodi oldingi boblardagi kabi klassik qidiruv usullarini veb-domenga o'tkazdi, bu esa masshtab muammosiga e'tibor qaratishga olib keldi. Eng qadimgi veb-qidiruv tizimlari o'n millionlab hujjatlarni o'z ichiga olgan indekslar bilan kurashishga majbur bo'ldi, bu umumiy foydalanishdagi har qanday oldingi ma'lumot qidirish tizimidan bir necha baravar kattaroq edi. Ushbu miqyosda indekslash, so'rovlarga xizmat ko'rsatish va tartiblash iste'molchi qidiruvi ilovasida hali kuzatilmagan miqyosda yuqori darajada mavjud tizimlarni yaratish uchun o'nlab mashinalarni birlashtirishni talab qildi. Veb-qidiruv mexanizmlarining birinchi avlodi ushbu muammolarni hal qilishda muvaffaqiyat qozondi, shu bilan birga Internetning muhim qismini doimiy ravishda indekslash, ikkinchidan past javob vaqtlari bilan so'rovlarga xizmat ko'rsatish bilan birga amalga oshirildi. Biroq, veb-qidiruv natijalarining sifati va dolzarbligi 8.2-bo'limda muhokama qilinadigan Internetda kontent yaratishning o'ziga xos xususiyatlari

tufayli ko'p narsani talab qildi. Bu qidiruv natijalarining sifatini ta'minlash uchun yangi reyting va spamga qarshi kurash usullarini ixtiro qilishni talab qildi. Klassik ma'lumot qidirish usullari (masalan, ushbu kitobda ilgari yoritilgan) veb-qidiruv uchun zarur bo'lib qolsa-da, ular hech qanday tarzda yetarli emas. Muhim jihat (10-bobda batafsil ishlab chiqilgan) shundan iboratki, klassik usullar hujjatning so'rovga mosligini o'lchasa-da, hujjatning vakolatliligini qaysi veb-saytda joylashganligi kabi belgilar asosida o'lchash zarurati saqlanib qolmoqda.

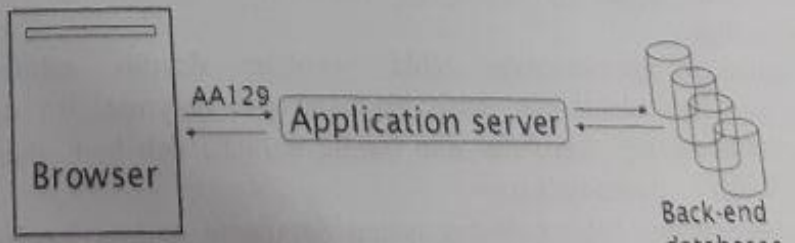
8.2. Veb xususiyatlari

Vebning jadal ko'rinishidagi ma'lumotlarini o'sishiga olib kelgan muhim xususiyat - mualliflikni markaziy nazoratisiz markazlashtirilmagan kontentni nashr etish bo'ldi. Bu - veb-qidiruv tizimlari uchun ushbu tarkibni indekslash va olish uchun eng katta muammo bo'lib chiqdi. Veb-sahifa mualliflari o'nlab tillarda va minglab dialektlarda tarkib yaratdi, shuning uchun stemming va boshqa lingvistik operatsiyalarning ko'p turli shakllarini talab qiladi. Nashr qilish endi o'n millionlab odamlar uchun ochiq bo'lganligi sababli, veb-sahifalar ko'p muhim jihatlarda dahshatli miqyosda ma'lumotlarni taqdim etish imkonini berdi. Birinchidan, kontent yaratish endi muharrirlik bo'yicha o'qitilgan yozuvchilarga tegishli emas edi. Bu kontent yaratishning ulkan demokratlashuvini ifodalagan bo'lsa-da, u grammatika va uslubning katta o'zgarishiga olib keldi. Darhaqiqat, veb-nashr qaysidir ma'noda sayyoraviy miqyosda ish stoli nashrining eng yaxshi va eng yomonini ochib berdi, shuning uchun sahifalar tezda ranglar, shriftlar va tuzilishdagi vahshiy o'zgarishlar bilan to'lib ketdi. Ba'zi veb-sahifalar, jumladan, ba'zi yirik korporatsiyalarning professional tarzda yaratilgan uy sahifalari, butunlay tasvirlardan iborat edi (u bosilganda, matn mazmuni yanada boy bo'ldi) va shuning uchun indekslanadigan matn yo'q edi.

Veb-sahifalardagi matnning mazmuni haqida nima deyish mumkin? Internetda kontent yaratishni demokratlashtirish deyarli har qanday mavzu bo'yicha fikrlashning yangi darajasini anglatardi. Bu shuni anglatadiki, tarmoq haqiqat, yolg'on, qarama-qarshilik va taxminlarni o'z ichiga oladi. Savol tug'iladi: qaysi veb-sahifalarga ishonish kerak? Soddalashtirilgan yondashuvda, ba'zi noshirlar ishonchli, boshqalari esa

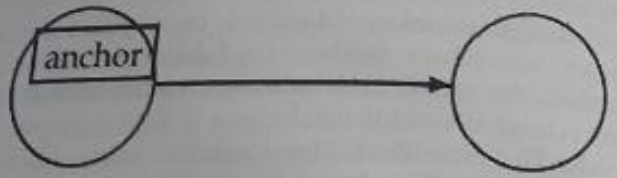
ishonchli emas, deb bahslashish mumkin - qidiruv tizimi har bir veb-sayt yoki veb-sahifaga bunday ishonch o'lchovini qanday belgilashi haqida savol tug'iladi. 10-bobda ushbu savolni tushunishga yondashuvlar ko'rib chiqiladi.

Aniqroq aytganda, universal, foydalanuvchidan mustaqil ishonch tushunchasi bo'lmasligi mumkin. Mazmuni bir foydalanuvchi uchun ishonchli bo'lgan veb-sahifa boshqa foydalanuvchi uchun bunday bo'lmasligi mumkin. An'anaviy (veb bo'lmagan) nashrlarda bu muammo emas: foydalanuvchilar ishonchli manbalarni o'zlari tanlaydi. Shunday qilib, bir o'quvchi ***The New York Times*** xabarini ishonchli deb topishi mumkin, boshqasi esa ***The Wall Street Journalni*** afzal ko'rishi mumkin. Ammo qidiruv tizimi foydalanuvchining ko'p kontentdan xabardor bo'lishi uchun yagona samarali vosita bo'lsa, bu muammo ahamiyatli bo'ladi. "Internet qanchalik katta?" degan savol tug'iladi. Oson javob yo'q (8.5-bo'limga qarang), "qidiruv tizimi indeksida qancha veb-sahifa bor" degan savol aniqroq, garchi bu savolda ham muammolar mavjud. 1995 yil oxiriga kelib, Altavista taxminan 30 million statik veb-sahifalarni skanerlash va indekslash haqida xabar berdi. Statik veb-sahifalar mazmuni o'sha sahifa uchun so'rovdan ikkinchisiga o'zgarmaydigan sahifalardir. Buning uchun har hafta bosh sahifasini qo'lda yangilaydigan professor statik veb-sahifaga ega, ammo aeroportning parvoz holati sahifasi dinamik hisoblanadi. Dinamik sahifalar odatda 8.1-rasmda ko'rsatilganidek, ma'lumotlar bazasidagi so'rovga javoban dastur serveri tomonidan mexanik ravishda yaratiladi. Bunday sahifaning belgilaridan biri URL manzilida "?" belgisidir. 1995 yilda statik veb-sahifalar soni har bir necha oyda ikki baravar ko'payib borayotganligi sababli, Altavista kabi dastlabki veb-qidiruv tizimlari veb-sahifalarni skanerlash va indekslash uchun doimiy ravishda apparat va tarmoqli kengligi qo'shishga majbur bo'ldi.



8.1-rasm. **Dinamik ravishda yaratilgan veb-sahifa**

Brauzer veb-ilovaga AA129 reysi bo'yicha parvoz ma'lumotlari so'rovini yuboradi, bu ma'lumotni backend ma'lumotlar bazalaridan olinadi, so'ngra brauzerga qaytariladigan dinamik veb-sahifani yaratadi.



8.2-rasm. **Havola bilan birlashtirilgan veb-grafaning ikkita tugunlari**

8.2.1. Veb-grafik

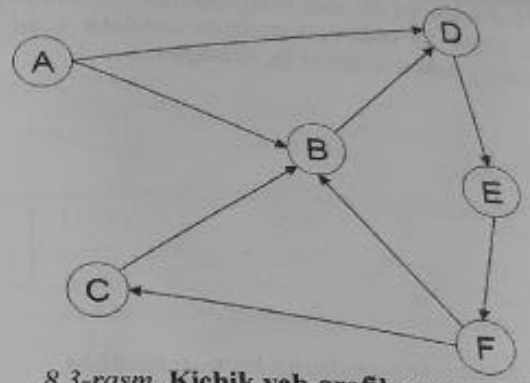
Statik HTML-sahifalardan tashkil topgan statik Webni va ular orasidagi giperhavolalarni yo'naltirilgan grafik sifatida ko'rishimiz mumkin, unda har bir veb-sahifa tugun va har bir giperhavola yo'naltirilgan manzildir. 8.2-rasmda veb-grafaning har biri veb-sahifaga mos keladigan, **A** dan **B** gacha bo'lgan giperhavolaga ega ikkita **A** va **B** tugunlari ko'rsatilgan.

Ular veb-grafik sifatida barcha bunday tugunlar va yo'naltirilgan manzillarning to'plamiga murojaat qiladi. 8.2-rasmda (veb-sahifalardagi aksariyat havolalarda bo'lgani kabi) **A** sahifadagi giperhavolaning kelib chiqishi bilan bog'liq ba'zi matn mavjudligi ham ko'rsatilgan. Bu matn odatda <**a**>(langar) tegining href atributiga kiritilgan.

Bu **A** sahifaning HTML kodidagi giperhavolani kodlaydi va langar matn deb ataladi. Biror kishi taxmin qilishi mumkinki, bu yo'naltirilgan grafik kuchli bog'liq emas: sahifalar juftligi borki, ulardan biri giperhavolalar orqali juftlikning bir sahifasidan ikkinchisiga o'tish mumkin emas.

Sahifadagi giperhavolalar ichki havolalar deyilib, sahifadan tashqaridagi havolalarni esa tashqi havolalar deb ataymiz. Bir qator tadqiqotlarda sahifaga havolalar soni (uning darajasi deb ham ataladi) o'rtacha 8 dan 15 gacha bo'lgan.

Xuddi shunday veb-sahifaning tashqi darajasini undagi havolalar soni deb belgilanadi. Bu tushunchalar 8.3-rasmda keltirilgan.

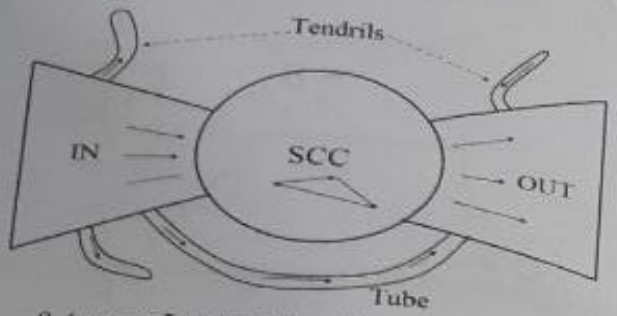


8.3-rasm. **Kichik veb-grafik namunasi**

Ushbu misolda ularda **A-F** etiketli oltita sahifa mavjud. **B** sahifasi 3-darajali va 1- darajaga ega. Ushbu misol grafik kuchli bog'lanmagan: **B-F** sahifalarining hech biridan **A** sahifasiga yo'l yo'q.

Ushbu havolalar tasodifiy taqsimlanmaganligi haqida ko'plab dalillar mavjud. Birinchidan, veb-sahifaga havolalar sonini taqsimlash, agar har bir veb-sahifa o'z havolalarining manzillarini tasodifiy bir xilda tanlasa, kutilgan ***Puasson taqsimotiga*** mos kelmaydi. Aksincha, bu taqsimot kuch qonuni sifatida keng tarqalgan bo'lib, unda **i** darajasidagi veb-sahifalarning umumiy soni $1/i^{\alpha}$ ga proportsionaldir. Odatda tadqiqotlar tomonidan bildirilgan **a** qiymati **2,1** ni tashkil qiladi. Bundan tashqari, bir nechta tadqiqotlar veb-sahifalarni bog'laydigan yo'naltirilgan grafikning kamon shakliga ega ekanligini ko'rsatdi: veb-sahifalarning uchta asosiy toifasi bor, ular ba'zan IN, OUT va SCC deb nomlanadi. Veb-surfer giperhavolalar orqali IN dagi istalgan sahifadan SCC ning istalgan sahifasiga o'tishi mumkin. Xuddi shunday, surfer SCCdagi sahifadan OUTdagi istalgan sahifaga o'tishi mumkin. Nihoyat, surfer SCCdagi istalgan sahifadan SCCdagi istalgan boshqa sahifaga o'tishi mumkin. Biroq, SCCdagi sahifadan IN dagi biron bir sahifaga yoki OUT dagi sahifadan SCCdagi sahifaga (yoki, demak, IN) o'tish mumkin emas. Shunisi e'tiborga loyiqki, bir nechta tadqiqotlarda **IN** va **OUT** hajmi taxminan teng, **SCC** esa biroz kattaroqdir. Ko'pchilik veb-sahifalar ushbu uchta to'plamdan biriga kiradi. Qolgan sahifalar **SCC** tashqarisidagi kichik sahifalar to'plamidan iborat bo'lib, ular to'g'ridan-to'g'ri **IN** dan

OUT ga olib boruvchi va **IN** dan hech qayerga olib bormaydigan yoki hech qayerdan **OUT** ga olib bormaydigan novdalar hosil qiladi. 8.4-rasmda Internetning ushbu tuzilishi tasvirlangan.



8.4-rasm. **Internetning kamon tuzilishi**

Bu yerda bitta naycha va uchta paycha ko'rsatilmoqda.

8.2.2. Spam

Veb-qidiruv tarixining boshida veb-qidiruv tizimlari reklama beruvchilarni bo'lajak xaridorlar bilan bog'lashning muhim vositasi ekanligi ayon bo'ldi. ***Maui golf*** ko'chmas mulkini qidirayotgan foydalanuvchi nafaqat ***Maui orolidagi golf*** maydonlarida uy-joy haqida yangiliklar yoki o'yin-kulgilarni izlamaydi, balki uning o'rniga bunday mulkni sotib olishga intiladi. Shunday qilib, bunday mulk sotuvchilari va ularning agentlari ushbu so'rov bo'yicha yuqori o'rinni egallagan veb-sahifalarni yaratish uchun kuchli rag'batga ega. Baholash atama chastotalariga asoslangan qidiruv tizimida ***Maui golf*** ko'chmas mulkining ko'p marta takrorlangan veb-sahifasi yuqori o'rinni egallaydi. Bu spamning birinchi avlodiga olib keldi, bu (veb-qidiruv kontekstida) tanlangan kalit so'zlar bo'yicha qidiruv natijalarida yuqori ko'rinishga ega bo'lish maqsadida veb-sahifa tarkibini manipulyatsiya qilishdir. Ushbu takrorlashlar bilan foydalanuvchilarni bezovta qilmaslik uchun murakkab spamchilar murojaat qilishdi. Bu takrorlangan atamalarni fon bilan bir xil rangda ko'rsatish kabi hiyla-nayranglarga boy edi. Ushbu so'zlar inson foydalanuvchisi uchun ko'rinmas bo'lishiga qaramay, qidiruv tizimining indeksatori veb-sahifaning HTML ko'rinishidagi ko'rinmas so'zlarni tahlil qiladi va bu so'zlarni sahifada mavjud sifatida indekslaydi.

8.5-rasm. **Spamerlar tomonidan qo'llaniladigan yashirin murojaat**

Spamning ildizida Internetda kontent yaratish motivlarining xilma-xilligidan kelib chiqadi. Xususan, ko'plab veb-kontent yaratuvchilari tijorat maqsadlariga ega va shuning uchun qidiruv tizimining natijalarini manipulyatsiya qilishdan foyda olishadi. Siz bu sariq sahifalarda telefon raqamlarini ro'yxatga olish uchun katta shriftlardan foydalanadigan kompaniyadan farqi yo'q deb bahslashishingiz mumkin. Lekin bu odatda kompaniyaga ko'proq xarajat olib keladi va shuning uchun adolatli mexanizmdir. Sariq sahifalar toifasida ro'yxatga olinadigan **A** ning uzun qatori bilan boshlangan kompaniya nomlaridan foydalanish yanada mosroq o'xshashlikdir. Aslida, sariq sahifalarning kattaroq/quyuqroq shriftlar uchun to'lovchi kompaniyalar modeli veb-qidiruvda takrorlangan: ko'plab qidiruv tizimlarida o'z veb-sahifasini qidiruv tizimi indeksiga kiritish uchun pul to'lash mumkin - bu model pulli hisoblandi. Turli xil qidiruv tizimlarida pulli inklyuziyaga ruxsat berish yoki bunday to'lovning qidiruv natijalaridagi reytingga ta'siri bor-yo'qligi bo'yicha turli xil siyosatlar mavjud. Tez orada qidiruv tizimlari spamni aniqlashda ma'lum kalit so'zlarning ko'p sonli takrorlanishini aniqlash uchun yetarlicha murakkab bo'ldi. Spamerlar ***spam*** texnikalarining yanada boy to'plami bilan javob berishdi, ulardan eng yaxshisini hozir tasvirlab beramiz.

Ushbu usullardan birinchisi 8.5-rasmda ko'rsatilgan plashdir. Bu yerda ***spamerning veb-serveri*** http so'rovi veb-qidiruv tizimining brauzeridan (qidiruv tizimining veb-sahifalarni to'playdigan qismi, 9-bobda tasvirlangan) yoki inson foydalanuvchi brauzeridan kelishiga qarab turli sahifalarni qaytaradi. Birinchisi, veb-sahifani noto'g'ri kalit so'zlar ostida qidiruv tizimi tomonidan indekslanishiga olib keladi. Foydalanuvchi ushbu kalit so'zlarni qidirganda va sahifani ko'rishni tanlaganida, u qidiruv tizimi tomonidan indekslanganidan butunlay boshqacha tarkibga ega veb-sahifani oladi. Qidiruv indekslarini bunday

aldash an'anaviy ma'lumot olish dunyosida noma'lum. Bu sahifa nashriyoti va veb-qidiruv tizimlari o'rtasidagi munosabatlar to'liq hamkorlikda emasligidan kelib chiqadi. Eshik sahifasi tanlangan qidiruv kalit so'zlari bo'yicha yuqori o'rinni egallash uchun sinchkovlik bilan tanlangan matn va metama'lumotlarni o'z ichiga oladi.

Kirish sahifasi tanlangan qidiruv kalit so'zlari bo'yicha yuqori o'rinni egallash uchun sinchkovlik bilan tanlangan matn va metama'lumotlarni o'z ichiga oladi. Brauzer kirish sahifasini so'raganda, u ko'proq tijoriy xarakterdagi kontentni o'z ichiga olgan sahifaga yo'naltiriladi. Murakkab spam yuborish usullari sahifa bilan bog'liq metama'lumotlarni, shu jumladan (ular 21-bobda ko'rib chiqiladigan sabablarga ko'ra) veb-sahifaga havolalarni o'z ichiga oladi. Spam-xabar jo'natish tabiatan iqtisodiy asosli faoliyat ekanligini hisobga olsak, uning atrofida o'z veb-sahifalari tanlangan kalit so'zlar bo'yicha yuqori o'rinni egallashga intilayotgan mijozlarga maslahat xizmatlarini ko'rsatish uchun qidiruv tizimini optimallashtiruvchilar sanoati paydo bo'ldi. Veb-qidiruv mexanizmlari o'zlarining xususiy tartiblash usullarini ochishga va moslashishga urinishlarni oqlamadi, haqiqatan ham o'zlari toqat qilmaydigan SEO xatti-harakatlari shakllari bo'yicha siyosatni e'lon qiladi (ma'lumki, Ba'zi SEO-larning qidiruv so'rovlarini ularni burgalik uchun o'chirib qo'yishadi). Muqarrar ravishda, bunday SEOlar (har bir veb-qidiruv tizimining tartiblash usullarining xususiyatlarini asta-sekin aniqlaydigan) va veb-qidiruv tizimlari (javobga moslashadigan) o'rtasidagi to'xtovsiz kurashdir. Darhaqiqat, ushbu jang atrofida qarama-qarshilik ma'lumotlarini izlashning tadqiqot sub-sohasi paydo bo'ldi. O'z veb-sahifalarining matnini o'zgartiradigan spamerlarga qarshi kurashish uchun Internetning havolalar tuzilmasidan foydalanish - bu havola tahlili deb nomlanuvchi usuldir. Havola tahlilini keng miqyosda qo'llashi ma'lum bo'lgan birinchi veb-qidiruv tizimi (10-bobda batafsil ma'lumot beriladi) Google edi, garchi hozirda barcha veb-qidiruv tizimlari undan foydalanmoqda (shunga mos ravishda, spamerlar endi uni buzish uchun katta kuch sarflamoqdalar - bu havola spam sifatida tanilgan).

8.1-mashq. Agar i darajali sahifalar soni $1/i^{2.1}$ ga proportsional bo'lsa, tasodifiy tanlangan veb-sahifaning 1-darajali bo'lish ehtimoli qanday?

8.2-mashq. Agar i darajali sahifalar soni $1/i^{2.1}$ ga mutanosib bo'lsa, veb-sahifaning o'rtacha darajasi qancha?

8.3-mashq. Agar i darajali sahifalar soni $1/i^{2.1}$ ga mutanosib bo'lsa, eng katta darajaga cheksizlikka o'tishi bilan i darajali sahifalar ulushi o'zgarmaydimi yoki kamayadimi? 2.1 dan boshqa ko'rsatkich darajasi uchun javobingiz qanday o'zgaradi?

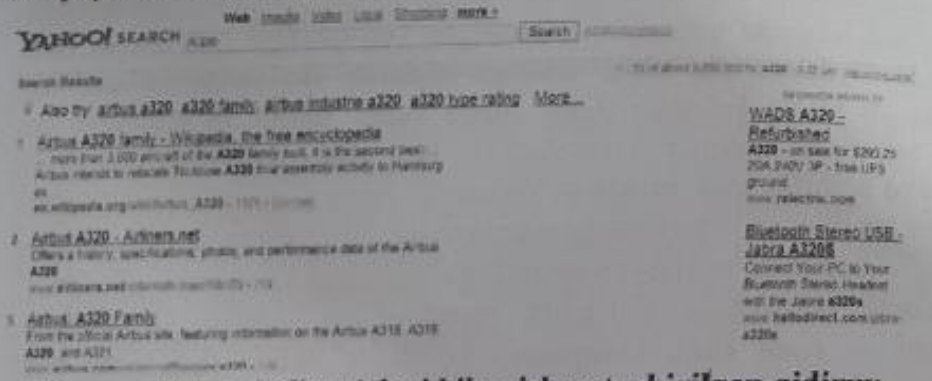
8.4-mashq. Veb-grafik suratidagi barcha tugunlarning o'rtacha darajasi 9 ga teng. Ushbu suratdagi barcha tugunlarning o'rtacha tashqi darajasi haqida nima deyish mumkin?

8.3. Reklama iqtisodiy model sifatida

Internet tarixining boshida kompaniyalar mashhur veb-saytlar (MSN, America Online, Yahoo! va CNN kabi yangiliklar va ko'ngilochar saytlar) veb-sahifalarida grafik banner reklamalaridan foydalanganlar. Ushbu reklamalarning asosiy maqsadi brendlash edi: tomoshabinga reklama joylashtirgan kompaniyaning brendi haqida ijobiy his-tuyg'ularni etkarish. Odatda, bu reklama CPM ko'rsatkichlari har bir mil uchun xarajat (CPM) asosida baholanadi: kompaniyaning banner reklamasini 1000 marta ko'rsatish narxi. Ba'zi veb-saytlar o'zlarining reklama beruvchilari bilan shartnomalar tuzdi, bunda reklama narxi uning ko'rsatilishi soniga (shuningdek, taassurotlar deb ham ataladi) emas, balki foydalanuvchi tomonidan necha marta bosilganiga qarab baholanadi. Ushbu narxlash modeli CPC bo'lib, klik boshiga xarajat (CPC) modeli sifatida tanilgan. Bunday hollarda, reklama ustiga bosish foydalanuvchini reklama beruvchi tomonidan o'rnatilgan veb-sahifaga olib boradi va u yerda foydalanuvchi xarid qilishga undaydi. Bu yerda reklamaning maqsadi savdo belgisini ilgari surish emas. Brend va tranzaksiyaga yo'naltirilgan reklama o'rtasidagi bu farq allaqachon translyatsiya va bosma nashrlar kabi an'anaviy ommaviy axborot vositalari kontekstida keng tan olingan. Internetning interaktivligi CPC billing modeliga imkon berdi - bosishlar veb-sayt tomonidan o'lchanishi va nazorat qilinishi va reklama beruvchiga to'lanishi mumkin edi. Ushbu yo'nalishdagi kashfiyot *Goto* ismli kompaniya bo'lib, u *Yahoo* tomonidan sotib olinishidan oldin o'z nomini *Overture* ga o'zgartirdi! *Goto* an'anaviy ma'noda qidiruv tizimi emas edi. Aksincha, har bir so'rov muddati uchun u q so'rovida o'z veb-sahifasini ko'rsatishni istagan kompaniyalarning takliflarini qabul qildi. q so'roviga javoban *Goto* q uchun taklif qilgan barcha reklama beruvchilarning o'z takliflari bo'yicha tartiblangan

sahifalarini qaytaradi. Bundan tashqari, foydalanuvchi qaytarilgan natijalardan birini bosganida, tegishli reklama beruvchi ***Goto***-ga to'lovni amalga oshiradi (dastlabki amalga oshirishda bu to'lov reklama beruvchining **q** uchun taklifiga teng edi).

Goto modelining bir nechta jihatlarini ta'kidlash kerak. Birinchidan, ***Goto*** qidiruv interfeysiga **q** so'rovini kiritgan foydalanuvchi **q** so'roviga oid qiziqish va niyatni faol ifodalagan. Misol uchun, ***golf*** klublarini yozayotgan foydalanuvchi golf haqidagi yangiliklarni ko'rayotganga qaraganda, to'plamni tez orada sotib olishi mumkin. Ikkinchidan, ***Goto*** faqat foydalanuvchi reklamaga qiziqish bildirganida kompensatsiya oldi - bu foydalanuvchi reklamani bosganidan dalolat beradi. Birgalikda bular reklama beruvchilarni iste'molchilar bilan bog'lash uchun kuchli mexanizm yaratdi va ***Goto/Overture*** yillik daromadini tez orada yuzlab million dollarga oshirdi. Qidiruv tizimining ushbu uslubi turlicha homiylik qidiruvi yoki qidiruv reklamasi sifatida tanildi. Ushbu ikki turdagi qidiruv tizimlarini hisobga olgan holda - ***Google*** va ***Altavista*** kabi "sof" qidiruv tizimlari homiylik qidiruv tizimlariga nisbatan - keyingi mantiqiy qadam birlashtirish edi. Ularni bitta foydalanuvchi tajribasiga aylantirish kerak edi. Joriy qidiruv tizimlari aynan shu modelga amal qiladi: ular foydalanuvchi qidiruviga asosiy javob sifatida sof qidiruv natijalarini (odatda algoritmik qidiruv natijalari deb nomlanadi) hamda algoritmik natijalarning o'ng tomonida alohida va alohida ko'rsatiladigan homiy qidiruv natijalarini taqdim etadi.



8.6-rasm. **So'rov kalit so'zlari bilan ishga tushirilgan qidiruv reklamasi**

Bu 8.6-rasmda ko'rsatilgan. Homiylik ostidagi qidiruv natijalarini olish va so'rovga javoban ularni tartiblash endi oddiy ***Goto*** sxemasiga

qaraganda ancha murakkablashdi hamda jarayon axborotdan g'oyalarni birlashtirishni o'z ichiga oldi.

Bu yerda ***A320*** so'rovi ***Airbus samolyoti*** haqidagi algoritmik qidiruv natijalarini hamda reklama beruvchilar ushbu so'rov bo'yicha so'raganlarga sotmoqchi bo'lgan ***A320 raqamli samolyotga*** tegishli bo'lmagan tovarlarning reklamalarini qaytaradi. Samolyot uchun reklamalarning yo'qligi kam sonli sotuvchilar A320 samolyotlarini Internetda sotishga harakat qilishini aks ettiradi.

Izlash va mikroiqtisodiyot va darslikning doirasidan tashqarida. Reklama beruvchilar uchun qidiruv tizimlari bu reytingni qanday amalga oshirishini va marketing kampaniyasi byudjetlarini turli kalit so'zlarga va turli homiyligidagi qidiruv tizimlariga qanday taqsimlashni tushunish qidiruv tizimi **G** marketingi (**SEM**) deb nomlanuvchi kasbga aylandi. Ba'zi ishtirokchilarning tizimni o'z manfaati uchun buzishga urinishlari kuchaymoqda edi. Bu juda ko'p shakllarga ega bo'lishi mumkin va ulardan biri spamni bosish deb nomlanadi. Hozirda spamni bosishning umume'tirof etilgan ta'rifi yo'q. Bu (nomidan ko'rinib turibdiki) qidiruv foydalanuvchilaridan bo'lmagan homiylik qilingan qidiruv natijalariga bosishlarga ishora qiladi. Misol uchun, ayyor reklama beruvchi raqobatchining homiyligidagi qidiruv e'lonlarini qayta-qayta bosish (robot-klik generatoridan foydalanish orqali) orqali raqobatchining reklama byudjetini tugatishga urinishi mumkin. Qidiruv mexanizmlari reklama beruvchi mijozlaridan bunday bosishlar uchun haq to'lamaslik uchun ular kuzatayotgan sekin urishlarning qaysi biri spam bosish namunasi ekanligini aniqlash muammosiga duch kelishadi.

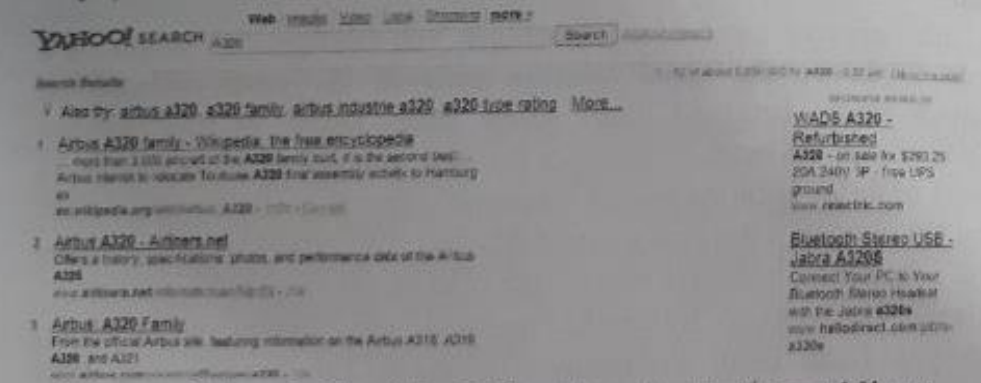
8.5-mashq.

Goto usuli so'rovga mos keladigan reklamalarni taklif bo'yicha tartiblaydi: eng yuqori narx taklif qilgan reklama beruvchi yuqori o'rinni egallaydi, ikkinchi o'rinni egalladi va hokazo. Eng ko'p taklif qiluvchi reklama beruvchi so'rovga aloqador bo'lmagan reklama joylashtirganda nima noto'g'ri bo'lishi mumkin? Nima uchun ahamiyatsiz reklamaga ega bo'lgan reklama beruvchi bunday tarzda yuqori narx qo'yishi mumkin?

8.6-mashq. Aytaylik, takliflardan tashqari, ularda har bir reklama beruvchi uchun ularning bosish tezligi mavjud edi: foydalanuvchilar o'z reklamalarini bosganliklari sonining reklama ko'rsatilgan soniga nisbati. Yuqoridagi 8.5-mashqdagi muammoning oldini olish uchun ushbu ma'lumotlardan foydalanadigan ***Goto sxemasini*** o'zgartirishni taklif qiling.

sahifalarini qaytaradi. Bundan tashqari, foydalanuvchi qaytarilgan natijalardan birini bosganida, tegishli reklama beruvchi *Goto*-ga to'lovni amalga oshiradi (dastlabki amalga oshirishda bu to'lov reklama beruvchining **q** uchun taklifiga teng edi).

Goto modelining bir nechta jihatlarini ta'kidlash kerak. Birinchidan, *Goto* qidiruv interfeysiga **q** so'rovini kiritgan foydalanuvchi **q** so'roviga oid qiziqish va niyatni faol ifodalagan. Misol uchun, *golf* klublarini yozayotgan foydalanuvchi golf haqidagi yangiliklarni ko'rayotganga qaraganda, to'plamni tez orada sotib olishi mumkin. Ikkinchidan, *Goto* faqat foydalanuvchi reklamaga qiziqish bildirganida kompensatsiya oldi - bu foydalanuvchi reklamani bosganidan dalolat beradi. Birgalikda bular reklama beruvchilarni iste'molchilar bilan bog'lash uchun kuchli mexanizm yaratdi va *Goto/Overture* yillik daromadini tez orada yuzlab million dollarga oshirdi. Qidiruv tizimining ushbu uslubi turlicha homiylik qidiruvi yoki qidiruv reklamasi sifatida tanildi. Ushbu ikki turdagi qidiruv tizimlarini hisobga olgan holda - *Google* va *Altavista* kabi "sof" qidiruv tizimlari homiylik qidiruv tizimlariga nisbatan - keyingi mantiqiy qadam birlashtirish edi. Ularni bitta foydalanuvchi tajribasiga aylantirish kerak edi. Joriy qidiruv tizimlari aynan shu modelga amal qiladi: ular foydalanuvchi qidiruviga asosiy javob sifatida sof qidiruv natijalarini (odatda algoritmik qidiruv natijalari deb nomlanadi) hamda algoritmik natijalarning o'ng tomonida alohida va alohida ko'rsatiladigan homiy qidiruv natijalarini taqdim etadi.



8.6-rasm. **So'rov kalit so'zlari bilan ishga tushirilgan qidiruv reklamasi**

Bu 8.6-rasmda ko'rsatilgan. Homiylik ostidagi qidiruv natijalarini olish va so'rovga javoban ularni tartiblash endi oddiy *Goto* sxemasiga

qaraganda ancha murakkablashdi hamda jarayon axborotdan g'oyalarni birlashtirishni o'z ichiga oldi.

Bu yerda ***A320*** so'rovi ***Airbus samolyoti*** haqidagi algoritmik qidiruv natijalarini hamda reklama beruvchilar ushbu so'rov bo'yicha so'raganlarga sotmoqchi bo'lgan ***A320 raqamli samolyotga*** tegishli bo'lmagan tovarlarning reklamalarini qaytaradi. Samolyot uchun reklamalarning yo'qligi kam sonli sotuvchilar A320 samolyotlarini Internetda sotishga harakat qilishini aks ettiradi.

Izlash va mikroiqtisodiyot va darslikning doirasidan tashqarida. Reklama beruvchilar uchun qidiruv tizimlari bu reytingni qanday amalga oshirishini va marketing kampaniyasi byudjetlarini turli kalit so'zlarga va turli homiyligidagi qidiruv tizimlariga qanday taqsimlashni tushunish qidiruv tizimi **G** marketingi (**SEM**) deb nomlanuvchi kasbga aylandi. Ba'zi ishtirokchilarning tizimni o'z manfaati uchun buzishga urinishlari kuchaymoqda edi. Bu juda ko'p shakllarga ega bo'lishi mumkin va ulardan biri spamni bosish deb nomlanadi. Hozirda spamni bosishning umume'tirof etilgan ta'rifi yo'q. Bu (nomidan ko'rinib turibdiki) qidiruv foydalanuvchilaridan bo'lmagan homiylik qilingan qidiruv natijalariga bosishlarga ishora qiladi. Misol uchun, ayyor reklama beruvchi raqobatchining homiyligidagi qidiruv e'lonlarini qayta-qayta bosish (robot-klik generatoridan foydalanish orqali) orqali raqobatchining reklama byudjetini tugatishga urinishi mumkin. Qidiruv mexanizmlari reklama beruvchi mijozlaridan bunday bosishlar uchun haq to'lamaslik uchun ular kuzatayotgan sekin urishlarning qaysi biri spam bosish namunasi ekanligini aniqlash muammosiga duch kelishadi.

8.5-mashq.

Goto usuli so'rovga mos keladigan reklamalarni taklif bo'yicha tartiblaydi: eng yuqori narx taklif qilgan reklama beruvchi yuqori o'rinni egallaydi, ikkinchi o'rinni egalladi va hokazo. Eng ko'p taklif qiluvchi reklama beruvchi so'rovga aloqador bo'lmagan reklama joylashtirganda nima noto'g'ri bo'lishi mumkin? Nima uchun ahamiyatsiz reklamaga ega bo'lgan reklama beruvchi bunday tarzda yuqori narx qo'yishi mumkin?

8.6-mashq. Aytaylik, takliflardan tashqari, ularda har bir reklama beruvchi uchun ularning bosish tezligi mavjud edi: foydalanuvchilar o'z reklamalarini bosganliklari sonining reklama ko'rsatilgan soniga nisbati. Yuqoridagi 8.5-mashqdagi muammoning oldini olish uchun ushbu ma'lumotlardan foydalanadigan *Goto sxemasini* o'zgartirishni taklif qiling.

8.4. Qidiruv foydalanuvchisining tajribasi

Veb-qidiruv foydalanuvchilarini ham tushunishimiz juda muhimdir. Bu yana an'anaviy ma'lumot qidirishdan sezilarli o'zgarish bo'lib, foydalanuvchilar odatda uslubi va tuzilishini yaxshi tushunadigan yaxshi mualliflik to'plamiga nisbatan so'rovlarni ifodalash san'ati bo'yicha hech bo'lmaganda biroz tayyorgarlikka ega professionallar edi. Aksincha, veb-qidiruv foydalanuvchilari odatda veb-kontentning mukammalligi, so'rov tillari sintaksisi va so'rovlarni iboralash san'ati haqida bilmaydilar (yoki ahamiyat bermaydilar). Darhaqiqat, asosiy vosita (veb-qidiruv paydo bo'lganligi sababli) milliardlab odamlarga bunday og'ir talablarni qo'ymasligi kerak. Bir qator tadqiqotlar shuni ko'rsatdiki, veb-qidiruvdagi kalit so'zlarning o'rtacha soni 2 dan 3 gacha bo'ladi. Sintaksis operatorlari (mantiqiy birikmalar, joker belgilar va boshqalar) kamdan-kam qo'llaniladi, bu yana tomoshabinlar tarkibining natijasi – "normal" odamlar, axborot yaratuvchilar emas.

Ma'lumki, veb-qidiruv tizimi qancha ko'p foydalanuvchi trafikini jalb qilsa, homiylik qidiruvidan shunchalik ko'p daromad oladi. Qanday qilib qidiruv tizimlari o'zlarini farqlaydi va trafikni oshiradi? Bu yerda ***Google*** o'zining raqobatchilari hisobiga o'sishiga yordam beradigan ikkita tamoyilni aniqladi: (1) birinchi natijalarni eslab qolishdan ko'ra, dolzarblikka, xususan, aniqlikka e'tibor berish; (2) engil foydalanuvchi tajribasi, ya'ni qidiruv so'rovi sahifasi ham, qidiruv natijalari sahifasi ham tartibsiz va deyarli to'liq matnli, juda oz grafik elementlarga ega. Birinchisining ta'siri shunchaki foydalanuvchilarning o'zlari qidirayotgan ma'lumotni topish vaqtini tejash edi. Ikkinchisining ta'siri foydalanuvchiga juda sezgir bo'lgan yoki har qanday holatda qidiruv so'rovi yoki natijalar sahifasini yuklash vaqti bilan bog'liq bo'lmagan foydalanuvchi tajribasini taqdim etishdir.

8.4.1. Foydalanuvchi so'rovining ehtiyojlari

Umumiy veb-qidiruv so'rovlarini guruhlash mumkin bo'lgan uchta keng toifa mavjud: (1) axborot, (2) navigatsiya va (3) tranzaksiya. Endi ular ushbu toifalarni tushuntiramiz. Ba'zi so'rovlar ushbu toifalarning bir nechtasiga to'g'ri kelishi aniq bo'lishi kerak, boshqalari esa ulardan tashqarida bo'ladi. Axborot so'rovlari ***leykemiya*** yoki ***Provans*** kabi keng

mavzu bo'yicha umumiy ma'lumotni qidiradi. Odatda qidirilayotgan barcha ma'lumotlarni o'z ichiga olgan bitta veb-sahifa yo'q. Haqiqatan ham, axborot so'rovlari bo'lgan foydalanuvchilar odatda bir nechta veb-sahifalardagi ma'lumotlarni o'zlashtirishga harakat qilishadi. ***Lufthansa*** aviakompaniyasining ta'kidlashicha, navigatsiya so'rovlari foydalanuvchi o'ylagan yagona obyektning veb-sayti yoki bosh sahifasini qidiradi. Bunday hollarda, foydalanuvchining taxmini shundaki, birinchi qidiruv natijasi ***Lufthansa*** bosh sahifasi bo'lishi kerak. Foydalanuvchini ***Lufthansa*** atamasi bo'lgan ko'plab hujjatlar qiziqtirmaydi. Bunday foydalanuvchi uchun foydalanuvchi qoniqishining eng yaxshi o'lchovi 1 deb qabul qilingan.

Tranzaksiya so'rovi foydalanuvchining ***Internetda tranzaktsiyani*** amalga oshirishi uchun muqaddima bo'lib xizmat qiladi - masalan, mahsulotni sotib olish, faylni yuklab olish yoki bron qilish. Bunday hollarda, qidiruv tizimi bunday operatsiyalar uchun shakl interfeyslarini ta'minlaydigan natijalar ro'yxati xizmatlarini qaytarishi kerak. Ushbu toifalarning qaysi biriga tegishli ekanligini aniqlash qiyin bo'lishi mumkin.

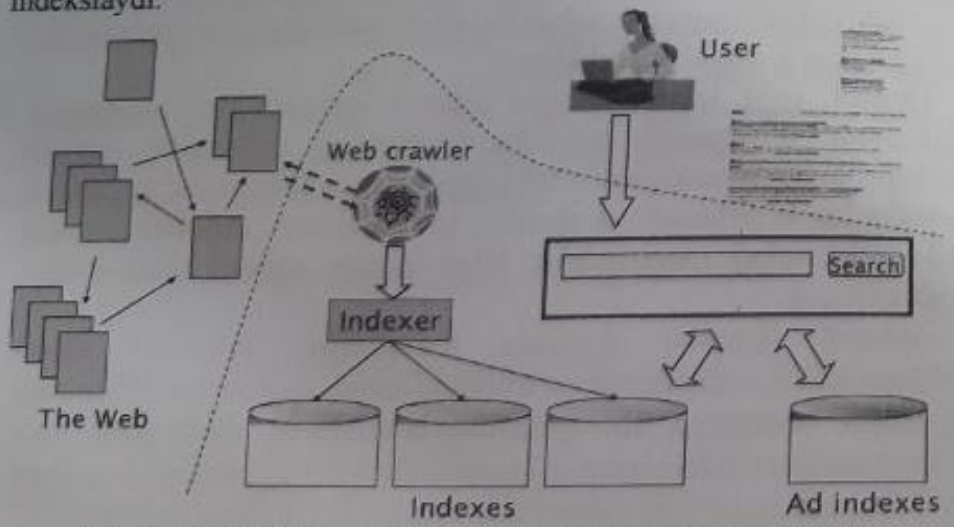
Kategoriya nafaqat algoritmik qidiruv natijalarini, balki so'rovning homiylik qilingan qidiruv natijalariga mosligini ham boshqaradi (chunki so'rov sotib olish niyatini ko'rsatishi mumkin). Navigatsiya so'rovlari uchun ba'zilar qidiruv tizimi faqat bitta natijani yoki hatto maqsadli veb-sahifani to'g'ridan-to'g'ri qaytarishi kerakligini ta'kidladi. Shunga qaramay, veb-qidiruv tizimlari tarixan qaysi biri ko'proq veb-sahifalarni indekslashi uchun maqtanish uchun kurashda qatnashgan. Foydalanuvchi haqiqatan ham g'amxo'rlik qiladimi?

Ehtimol, yo'q, lekin ommaviy axborot vositalari turli xil qidiruv tizimlarining o'lchamlari bo'yicha taxminlarni (ko'pincha statistik jihatdan himoyalanmagan) ta'kidlaydi. Foydalanuvchilar ushbu hisobotlardan ta'sirlanadi va shuning uchun qidiruv tizimlari ularning indeks o'lchamlari raqobatchilar bilan qanday solishtirishiga e'tibor berishlari kerak.

Axborot (kamroq darajada tranzaksiya) so'rovlari uchun foydalanuvchi qidiruv tizimining to'liqligi haqida qayg'uradi. 8.7-rasmda veb-qidiruv tizimining, shu jumladan brauzerning, shuningdek, veb-sahifa va reklama indekslarining kompozit tasviri ko'rsatilgan. Shaklning egri chiziq ostidagi qismi qidiruv tizimining ichki qismidir.

Birinchi taxminga ko'ra, qamrovlilik indeks hajmi bilan o'sib boradi, garchi qidiruv tizimi qaysi sahifalarni indekslashi muhim emas - ba'zi sahifalar boshqalarga qaraganda ko'proq ma'lumotga ega. Qidiruv tizimi tomonidan indekslangan Internetning ulushi haqida fikr yuritish ham qiyin chunki dinamik veb-sahifalarning cheksiz soni mavjud, masalan, http://www.yahoo.com/any_string xato emas, yaroqli HTML-sahifani qaytaradi va foydalanuvchiga *Yahoo!*-da bunday sahifa yo'qligi haqida xushmuomalalik bilan xabardor qiladi! Bunday "yumshoq 404 xatosi" veb-serverlar cheksiz miqdordagi haqiqiy veb-sahifalarni yaratishi mumkin bo'lgan ko'plab usullarning bir misolidir. Haqiqatan ham, ulardan ba'zilari qidiruv tizimining skanerini (9-bobda tasvirlangan qidiruv tizimi indeksi uchun veb-sahifalarni muntazam ravishda to'playdigan komponent) *spamerning* veb-saytida qolishiga va ushbu saytning ko'plab sahifalarini indekslashiga sabab bo'lish uchun ishlab chiqilgan zararli o'rgimchak tuzoqlaridir. Quyidagi aniqroq savolni berishimiz mumkin: Ikkita qidiruv tizimini hisobga olsak, ularning indekslarining nisbiy o'lchamlari qanday? Hatto bu savol noaniq bo'lib chiqadi, chunki:

1. Qidiruv tizimi so'rovlarga javoban mazmuni (to'liq yoki qisman) indekslanmagan veb-sahifalarni qaytarishi mumkin. Birinchidan, qidiruv tizimlari odatda veb-sahifadagi dastlabki bir necha ming so'zni indekslaydi.



8.7-rasm. **Web qidiruv tizimining turli komponentlari**

Ba'zi hollarda, qidiruv tizimi o'zi indekslagan sahifalar bilan bog'langan, lekin **p** ni o'zi indekslamagan **p** sahifasidan xabardor. 10-bobda keltirilganidek, qidiruv natijalarida **p**-ni mazmunli ravishda qaytarish hali ham mumkin.

2. Qidiruv mexanizmlari odatda o'z indekslarini turli darajalar va bo'limlarda tashkil qiladi, ularning hammasi ham har bir qidiruvda tekshirilmaydi (1- qismning 7.2.1-bo'limdagi darajali indekslarni eslang). Masalan, veb-sayt ichidagi veb-sahifa indekslanishi mumkin, ammo umumiy veb-qidiruvlarda olinmaydi. Biroq, foydalanuvchi ushbu veb-sayt bilan aniq cheklab qo'ygan qidiruv natijasida olinadi (bunday saytga xos qidiruv ko'pchilik veb-qidiruv tizimlari tomonidan taklif etiladi).

Shunday qilib, qidiruv tizimi indekslari indekslangan sahifalarning bir nechta sinflarini o'z ichiga oladi, shuning uchun indeks hajmining yagona o'lchovi mavjud emas. Ushbu muammolarga qaramay, $\mathbf{E}_1$ va $\mathbf{E}_2$ qidiruv tizimlarining indeks o'lchamlari nisbatlarini qo'pol baholash uchun bir qator usullar ishlab chiqilgan. Ushbu usullarning asosiy gipotezasi shundan iboratki, har bir qidiruv tizimi mustaqil va bir xilda tasodifiy tanlangan Internetning bir qismini indekslaydi. Bu ba'zi shubhali taxminlarni o'z ichiga oladi: birinchidan, har bir qidiruv tizimi quyi to'plamni tanlaydigan Internet uchun cheklangan o'lcham bor, ikkinchidan, har bir vosita mustaqil, bir xil tanlangan kichik to'plamni tanlaydi. 9-bobda sudralib yurish haqidagi muhokamadan ma'lum bo'lishicha, bu haqiqatdan uzoqdir. Ammo, agar ushbu taxminlar bilan boshlasak, unda qo'lga olish-qayta olish usuli deb nomlanuvchi klassik baholash usulini qo'llashimiz mumkin.

Aytaylik, $\mathbf{E}_1$ indeksidan tasodifiy sahifani tanlab, uning $\mathbf{E}_2$ indeksida va nosimmetrik ekanligini tekshirib ko'ramiz, $\mathbf{E}_2$ dan tasodifiy sahifa $\mathbf{E}_1$ da ekanligini tekshirib ko'ring. Ushbu tajribalar ularga **x** va **y** kasrlarini beradi, ularning taxminimizcha, $\mathbf{E}_1$ sahifalarining **x** qismi $\mathbf{E}_2$ da, $\mathbf{E}_2$ sahifasining **y** qismi esa $\mathbf{E}_1$ da. Keyin, ruxsat berish $|\mathbf{E}_i|$ $\mathbf{E}_i$ qidiruvi indeksining o'lchamini bildiradi, ular quyidagicha hisoblanadi:

$$x\,|E_1| \approx y\,|E_2|$$

undan foydalanadigan shaklga egamiz:

$$\frac{|E_1|}{|E_2|} \approx \frac{y}{x}.$$

Agar ularning E_1 va E_2 Internetning mustaqil va yagona tasodifiy kichik to'plamlari haqidagi taxminimiz to'g'ri bo'lsa va tanlama jarayonimiz xolis bo'lsa, (8.1) tenglama ularga $|E_1|/|E_2|$ uchun xolis baho berishi kerak. Bu yerda ikkita rejani ajratamiz. Yoki o'lchov qidiruv tizimlaridan birining indeksiga kirish huquqiga ega bo'lgan shaxs tomonidan amalga oshiriladi (aytaylik, E_1 xodimi), yoki o'lchov ikkala qidiruv tizimining ichki qismiga kirish huquqiga ega bo'lmagan mustaqil tomon tomonidan amalga oshiriladi. Birinchi holda, bitta indeksdan tasodifiy hujjatni tanlashimiz mumkin. Oxirgi holat yanada qiyinroq. Qidiruv tizimi tashqarisidan bitta qidiruv tizimidan tasodifiy sahifani tanlab, so'ngra tasodifiy sahifa boshqa qidiruv tizimida mavjudligini tekshiring. Namuna olish bosqichini amalga oshirish uchun butun (ideallashtirilgan, cheklangan) Internetdan tasodifiy sahifa yaratishimiz mumkin va uni har bir qidiruv tizimida mavjudligini sinab ko'rish mumkin. Afsuski, tasodifiy bir xilda veb-sahifani tanlash qiyin muammo. Har biriga xos bo'lgan belgilarni ko'rsatib, bunday namunaga erishish uchun bir nechta urinishlarni qisqacha bayon qilsh kerak. Shundan so'ng ko'p izlanishlar asosida qurilgan texnikani batafsil bayon qilsh mumkin. Tasodifiy qidiruvlar - veb-qidiruvlarning qidiruv jurnalidan boshlanadi. Bu jurnaldan tasodifiy qidiruvni E_1 ga va natijalardan tasodifiy sahifaga yuboriladi. Bunday jurnallar qidiruv tizimidan tashqarida keng tarqalmaganligi sababbi va amalga oshirishligi barcha qidiruvlarni jurnalga kiritishga rozi bo'lgan ishchi guruhidan (tadqiqot markazidagi olimlarning aytishlaricha) barcha qidiruv so'rovlarini tuzoqqa tushirishdir. Ushbu yondashuv bir qator muammolarga, jumladan, ishchi guruh tomonidan amalga oshirilgan qidiruv turlaridan nomaqbullikka ega. Bundan tashqari, bunday tasodifiy qidiruv natijalaridan E_1 ga tasodifiy hujjat E_1. E_2 dan tasodifiy hujjat bilan bir xil emas. Tasodifiy IP-manzillar: Ikkinchi yondashuv tasodifiy IP-manzillarni yaratish va tasodifiy manzilda joylashgan veb-serverga so'rov yuborish va shu serverdagi barcha sahifalarni yig'ishdir.

Tasodifiy qidiruvlar veb-qidiruvlarning qidiruv jurnalidan boshlanadi. Bu jurnaldan tasodifiy qidiruvni E_1 ga va natijalardan tasodifiy sahifaga yuboriladi. Bunday jurnallar qidiruv tizimidan tashqarida keng tarqalmaganligi va amalga oshirishini sababbi barcha qidiruvlarni jurnalga kiritishga rozi bo'lgan ishchi guruhdan (tadqiqot markazidagi olimlarning aytishlaricha) barcha qidiruv so'rovlarini tuzoqqa tushirishdir. Ushbu yondashuv bir qator muammolarga,

jumladan, ishchi guruh tomonidan amalga oshirilgan qidiruv turlaridan nomaqbullikka ega. Bundan tashqari, bunday tasodifiy qidiruv natijalaridan E_1 ga tasodifiy hujjat E_1 dan tasodifiy hujjat bilan bir xil emas.

Tasodifiy IP manzillar: Ikkinchi yondashuv tasodifiy IP manzillarni yaratish va tasodifiy manzilda joylashgan veb-serverga so'rov yuborish va shu serverdagi barcha sahifalarni yig'ishdir. Bu yerda noto'g'ri fikrlar ko'plab xostlar bitta IP-ni ishlatib ko'rishi (virtual hosting deb nomlanuvchi amaliyot tufayli) yoki tajriba o'tkazilayotgan xostdan http so'rovlarini qabul qilmasligini o'z ichiga oladi. Qolaversa, bu usul bir necha sahifali ko'plab saytlardan biriga kirib, hujjatning ehtimollarini chalg'itadi. Agar veb-saytlardagi sahifalar sonining taqsimlanishini tushunsak, bu ta'sirni to'g'rilashimiz mumkin.

Tasodifiy yurishlar: Agar veb-grafik kuchli bog'langan yo'naltirilgan grafik bo'lsa, ixtiyoriy veb-sahifadan boshlab tasodifiy yurishni amalga oshirishimiz mumkin edi. Bu yurish barqaror holat taqsimotiga yaqinlashadi (bu haqida ko'proq ma'lumot olish uchun 10-bob, 10.2.1-bo'limga qarang), printsipial jihatdan aniq ehtimollik bilan veb-sahifani tanlashimiz mumkin. Bu usul ham bir qator noaniqliklarga ega. Birinchidan, Internet kuchli bog'lanmagan, shuning uchun hatto turli xil tuzatish qoidalariga qaramay, har qanday sahifadan boshlab barqaror holat taqsimotiga erisha olamiz, deb bahslashish qiyin. Ikkinchidan, tasodifiy yurishning ushbu barqaror holatga kelishi uchun zarur bo'lgan vaqt noma'lum va tajriba uzunligidan oshib ketishi mumkin.

Shubhasiz, bu yondashuvlarning har biri mukammallikdan uzoqdir. Endi to'rtinchi tanlov usulini, tasodifiy so'rovlarni tasvirlaymiz. Ushbu yondashuv ikki sababga ko'ra e'tiborga loyiqdir: u tobora takomillashtirilgan bir qator hisob-kitoblar uchun muvaffaqiyatli qurilgan va aksincha, noto'g'ri talqin qilinishi va beparvolik bilan amalga oshirilishi mumkin bo'lgan yondashuv bo'lib chiqdi, bu noto'g'ri o'lchovlarga olib keladi. Maqsad, qidiruv tizimi indeksidan tasodifiy so'rov qo'yish orqali sahifani (deyarli) tasodifiy ravishda tanlashdir. ***Webster*** lug'atidan tasodifiy atamalar to'plamini tanlash bu g'oyani amalga oshirishning yaxshi usuli emasligi aniq bo'lishi kerak. Birinchidan, barcha lug'at atamalari bir xil darajada tez-tez uchramaydi, shuning uchun bu yondashuv hujjatlarni qidiruv tizimidan tasodifiy bir xilda tanlashga olib kelmaydi. Boshqacha qilib aytganda, veb-hujjatlarda Webster kabi standart lug'atda uchramaydigan juda ko'p atamalar

mavjud. Oddiy lug'atda bo'lmagan lug'at atamalari muammosini hal qilish uchun ishni namunaviy veb-lug'atni to'plashdan boshlaymiz kerak. Buni Internetning cheklangan qismini skanerlash yoki *Yahoo!* (bu usul bilan eng dastlabki tajribalarda qilinganidek). Ushbu lug'atdan tasodifiy tanlangan ikki yoki undan ortiq so'z bilan bog'liq so'rovni ko'rib chiqing. Operatsion tarzda ular quyidagicha harakat qiladi: ular E_1 da tasodifiy kon'yunktiv so'rovdan foydalaniladi va topilgan 100 ta natijadan tasodifiy **p** sahifa tanlanadi. Keyin **p** ning E_2 da mavjudligini tekshiramiz, **p** da 6-8 past chastotali atamalarni tanlaymiz va ularni E_2 uchun kon'yunktiv so'rovda ishlatamiz. Tajribani ko'p marta takrorlash orqali taxminni yaxshilashimiz mumkin. Namuna olish jarayonida ham, sinov jarayonida ham bir qator muammolar mavjud.

1. Ular namunamiz uzunroq hujjatlarga qarama-qarshidir.

2. E_1 ning eng yaxshi 100 ta natijalaridan tanlab olish E_1 reyting algoritmidan chetga chiqishga olib keladi. E_1 ning barcha natijalaridan tanlab olish tajribani sekinlashtiradi. Bu, ayniqsa, ko'pchilik veb-qidiruv tizimlari haddan tashqari robot so'rovlariga qarshi himoya vositalarini o'rnatganligi sabablidir.

3. Tekshiruv bosqichida bir qator qo'shimcha noaniqliklar kiritiladi: masalan, E_2 8 so'zli kon'yunktiv so'rovlarni to'g'ri bajarmasligi mumkin.

4. E_1 yoki E_2 test so'rovlariga javob berishdan bosh tortishi mumkin va ularni sof so'rovlar sifatida emas, balki robot spam sifatida ko'rib chiqadi.

5. Ulanish vaqti tugashi kabi operatsion muammolar bo'lishi mumkin.

Ushbu muammolarning ayrimlarini bartaraf etish uchun ushbu asosiy paradigma asosida bir qator tadqiqotlar olib borildi. Hali mukammal yechim yo'q, biroq statistik nuqtai nazarlarni tushunish uchun murakkablik darajasi ortib bormoqda. Asosiy g'oya - har bir hujjat uchun noto'g'rilikning kattaligini taxmin qilish orqali tarafdorlarni bartaraf etish. Bundan standart statistik tanlama usullari xolis namunalarni yaratishi mumkin. Tekshirish bosqichida yangi ish qo'shma so'rovlardan ibora va boshqa yaxshi bajarilgan so'rovlarga o'tadi. Va nihoyat, yangi tajribalar tasodifiy so'rovlardan tashqari boshqa tanlab olish usullaridan ham foydalanadi. Ulardan eng mashhuri hujjatlardan olingan virtual grafikda tasodifiy yurish orqali hujjat tanlab olinadigan tasodifiy yurish namunasi hisoblanadi. Ushbu grafikda tugunlar hujjatlardir. Ikkita hujjat ikki yoki undan ortiq umumiy so'zlarni ulashsa, chekka hududlar bilan

bog'lanadi. Grafik hech qachon instantsiyalanmaydi, balki uning ustida tasodifiy yurish **d** hujjatidan ikkinchisiga o'tish orqali **d** dagi kalit so'zlar juftligini tanlash, qidiruv tizimida so'rovni bajarish va natijalardan tasodifiy hujjatni tanlash orqali amalga oshirilishi mumkin. Tafsilotlarni 8.7-bo'limdagi havolalarda topish mumkin.

Mashq 8.7. Ikkita veb-qidiruv tizimi **A** va **B** har biri o'z indekslaridan tasodifiy ravishda ko'p sonli sahifalarni yaratadi. **A** sahifalarining **30%** **B** indeksida, **50%** **B** sahifasi esa **A** indeksida mavjud. **A** indeksidagi sahifalar soni **B** ga nisbatan qancha?

8.6. Takrorlangan hujjatlar va ularni indekslash

8.5-bo'limda indeks hajmini muhokama qilishda e'tibordan chetda qoldirgan jihat - bu takrorlash: Internetda bir xil tarkibning bir nechta nusxalari mavjud. Ba'zi hisob-kitoblarga ko'ra, Internetdagi sahifalarning 40% ga yaqini boshqa sahifalarning dublikatidir. Ularning aksariyati qonuniy nusxalardir, masalan, ba'zi ma'lumotlar omborlari ortiqcha va kirish ishonchliligini ta'minlash uchun shunchaki aks ettiriladi. Qidiruv mexanizmlari saqlash va qayta ishlash xarajatlarini kamaytirish uchun bir xil tarkibning bir nechta nusxalarini indekslashdan qochishga harakat qiladi.

Dublikatlarni aniqlashning eng oddiy usuli bu har bir veb-sahifa uchun ushbu sahifadagi belgilarning qisqacha (aytaylik, 64-bit) dayjesti bo'lgan barmoq izini hisoblashdir. Keyin ikkala veb-sahifaning barmoq izlari teng bo'lganda, sahifalarning o'zi teng yoki yo'qligini tekshiramiz va agar shunday bo'lsa, ulardan birini ikkinchisining nusxasi deb e'lon qiladi. Ushbu soddalashtirilgan yondashuv Internetda juda muhim va keng tarqalgan hodisani qo'lga kirita olmaydi: yaqin takrorlash.

Ko'pgina hollarda, bir veb-sahifaning mazmuni boshqasiniki bilan bir xil bo'ladi, bir nechta belgilardan tashqari, masalan, sahifa oxirgi marta o'zgartirilgan sana va vaqtni ko'rsatadigan belgi. Bunday holatlarda ham ikkita sahifani faqat bitta nusxasini indekslash uchun yetarlicha yaqin deb e'lon qilishni xohlaymiz.

Milliardlab sahifalar miqyosida amalga oshirib bo'lmaydigan vazifa bo'lgan barcha juft veb-sahifalarni to'liq taqqoslashdan so'ng, bunday yaqin dublikatlarni qanday aniqlashimiz va filtrlashimiz mumkin? Endi ikki nusxadagi veb-sahifalarni aniqlash muammosini hal qilishni

tasvirlaymiz. Javob **shingling** deb nomlanuvchi texnikada yotadi. Hujjatdagi **d** musbat butun son **k** va hadlar ketma-ketligini hisobga olib, **d** ning **k**-shingllarini **d** dagi **k** hadning barcha ketma-ketliklari to'plami sifatida aniqlanadi.

Misol tariqasida quyidagi matnni ko'rib chiqing: atirgul - bu atirgul. Ushbu matn uchun 4- shingles (**k = 4** deyarli takrorlanadigan veb-sahifalarni aniqlashda ishlatiladigan odatiy qiymat) atirgul **a**, atirgul atirguldir.

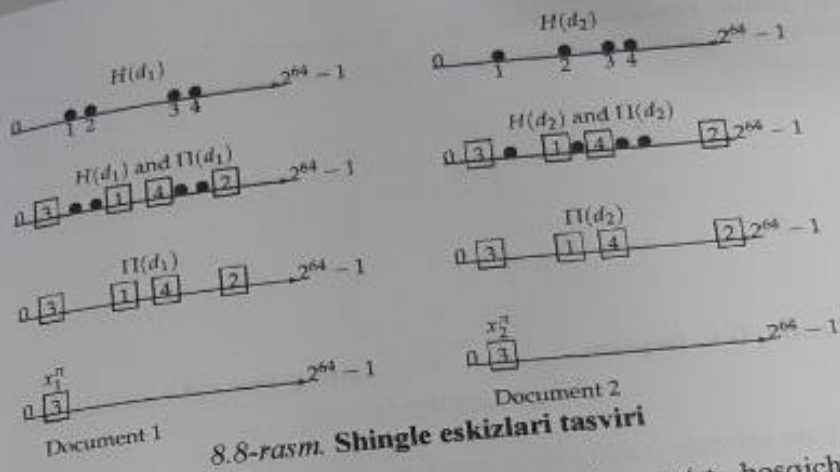
Ushbu shingillarning dastlabki ikkitasi matnda ikki marta uchraydi. Intuitiv ravishda ikkita hujjat ikki nusxada bo'ladi, agar ulardan yaratilgan shingillalar to'plami deyarli bir xil bo'lsa. Endi bu sezgini aniq qilish talab qilinadi, so'ngra barcha veb-sahifalar uchun shingillalar to'plamini samarali hisoblash va taqqoslash usulini ishlab chiqamiz.

S(dj) **dj** hujjatining shingillar to'plamini bildirsin. ***Jakkard koeffitsientini*** eslaylik, u **S(d1)** va **S(d2)** to'plamlar o'rtasidagi o'xshashlik darajasini $S(d_1)\ as\ |S(d_1)\cap S(d_2)|\ /\ |S(d_1)\cup S(d_2)|$; buni **J(S(d1), S(d2)** bilan belgilang. **d1** va **d2** o'rtasidagi yaqin takrorlanish uchun testimiz ushbu ***Jaccard koeffitsientini*** hisoblashdir. Agar u oldindan belgilangan chegaradan oshsa (aytaylik, 0,9), ularni dublikatlarga yaqin deb e'lon qiladi va bittasini indeksatsiyadan olib tashlaymiz.

Biroq, bu masala soddalashtirilganga o'xshaydi: hali ham ***Jaccard koeffitsientlarini*** juftlik bilan hisoblashimiz kerak.

Bunga yo'l qo'ymaslik uchun xeshlash usulidan foydalaniladi. Birinchidan, har bir shingilni katta maydondagi xesh qiymatiga, masalan, 64 bitga aylantiramiz. **j = 1, 2** uchun **H(dj) S(dj)** dan olingan 64 bitli xesh qiymatlarining mos keladigan to'plami bo'lsin. **H()** to'plamlari katta ***Jaccard*** o'zaro o'xshashligi bo'lgan hujjat juftlarini aniqlash uchun endi quyidagi hiyla-nayrangni ishlatamiz. π 64 bitli butun sonlardan 64 bitli butun sonlarga tasodifiy almashtirish bo'lsin. $\prod(d_j)$ da almashtirilgan xesh qiymatlar to'plamini $h \in H(d_j)$ bilan belgilang. Shuning uchun har bir $h \in H(d_j)$ uchun mos keladigan $\pi(h) \in \prod(d_j)$. qiymat mavjud. x_j^π dagi eng kichik butun son $\prod(d_j)$ bo'lsin. Keyin teorema 8.1.

$$J(S(d_1), S(d_2)) = P(x_1^\pi = x_2^\pi)$$



8.8-rasm. **Shingle eskizlari tasviri**

Ikkita hujjat shingil eskizini hisoblashning to'rt bosqichidan o'tayotganini ko'ramiz. Birinchi bosqichda (yuqori qator) **H(d1)** va **H(d2)** (doiralar) olish uchun har bir hujjatdan har bir shingilga 64 bitli xeshni qo'llaymiz. Keyinchalik, **H(d1)** va **H(d2)** ni almashtirish uchun **n** tasodifiy almashtirishni qo'llaymiz, **n(d1)** va **n(d2)** (kvadratchalar) olamiz. Uchinchi qatorda faqat $\prod(d_1)$ va $\prod(d_2)$ pastki qatorda esa har bir hujjat uchun x_1^π va x_2^π minimal qiymatlari ko'rsatilgan.

Isbot. Dalilni biroz umumiyroq tarzda keltiramiz: elementlari umumiy olingan to'plamlar oilasini ko'rib chiqing. To'plamlarni koinotdagi har bir element uchun bitta qatorli **A** matritsasining ustunlari sifatida ko'ring. **j**-ustun ko'rsatadigan S_j to'plamda **i** element mavjud bo'lsa, $a_{ij} = 1$ element. **n A** qatorlarining tasodifiy almashtirilishi bo'lsin; $\prod$ ustunga **n** qo'llanilishi natijasida hosil bo'lgan ustunni $\prod(S_j)$ bilan belgilanadi. Nihoyat, x_j^π ustuni 1 ga ega bo'lgan birinchi qatorning indeksi bo'lsin. Keyin har qanday ikkita j_1, j_2, ustunlari uchun,

$$P(x_{j_1}^\pi = x_{j2}^\pi) = J(S_{j1}, S_{j2})$$

hisoblanadi. Agar buni isbotlay olsak, teorema quyidagicha bo'ladi.

8.9-rasmda ko'rsatilganidek, ikkita **j1, j2** ustunlarini ko'rib chiqing. S_{j1} va S_{j2} yozuvlarining tartiblangan juftliklari qatorlarni to'rt turga bo'ladi: bu ustunlarning ikkalasida 0 ga ega bo'lganlar, S_{j1} da 0 va S_{j2} da 1, S_{j1} da 1 va S_{j2} da 0 bo'lgan va nihoyat, bu ikkala ustunda 1 ga ega

bo'lganlar hisoblanadi. Darhaqiqat, 8.9-rasmning birinchi to'rt qatori ushbu to'rt turdagi qatorlarning barchasiga misoldir.

S_{j_1}	S_{j_2}
0	1
1	0
1	1
0	0
1	1
0	1

8.9-rasm. **Ikki to'plam S_{j_1} va S_{j_2} va ularning Jaccard koeffitsienti 2/5 ga tengligi jadvali**

Ikkala ustunda 0 dan iborat qatorlar sonini C_{00} bilan belgilang, C_{01} ikkinchi, C_{10} uchinchi va C_{11} to'rtinchi. Keyin,

$$J(S_{j1}, S_{j2}) = \frac{C_{11}}{C_{01} + C_{10} + C_{11}}$$

(8.2) tenglamaning o'ng tomoni $P(x_{j_1}^{\pi} = x_{j_2}^{\pi})$ ga teng ekanligini ko'rsatish orqali isbotlashni yakunlash uchun birinchi nolga teng bo'lmagan yozuv topilmaguncha satr indeksini oshirishda j_1, j_2 ustunlarini skanerlashni ko'rib chiqing. Ikkala ustunda P tasodifiy almashtirish bo'lgani uchun bu eng kichik satr ikkala ustunda 1 ga ega bo'lish ehtimoli (8.2) tenglamaning aynan o'ng tomonidir. Shunday qilib, shingil to'plamlarining ***Jaccard koeffitsienti*** uchun testimiz ehtimollikdir: ular turli hujjatlardan $\mathbf{x_p}^i$ hisoblangan qiymatlarni solishtiramiz. Agar juftlik mos kelsa, ularda dublikatlarga yaqin nomzodi bor. Jarayonni 200 ta tasodifiy almashtirish **p** uchun mustaqil ravishda takrorlang (adabiyotda tavsiya etilgan tanlov). x_i^{π} ning 200 ta natijaviy qiymatlari to'plamini π ning $\psi(d_i)$ eskizi deb ataymiz. Keyin har qanday d_i, d_j hujjatlar juftligi uchun ***Jakkard koeffitsientini*** $|\psi_i \cap \psi_j|$ 200 bo'lishini taxmin qilishimiz mumkin. Agar bu oldindan belgilangan chegaradan oshsa, d_i, d_j o'xshashligini e'lon qilinadi. Qanday qilib barcha **i, j** juftliklari uchun $|\psi_i \cap \psi_j|$/ 200 ni tezda hisoblashimiz mumkin? Haqiqatan ham, hujjatlar sonida kvadratik bo'lgan noaniqlik sodir

bo'lmasdan, o'xshash hujjatlarning barcha juftlarini qanday ifodalaymiz? Birinchidan, bir xil hujjatlarning bir nusxasidan tashqari hammasini olib tashlash uchun barmoq izlaridan foydalaniladi. Hujjatlarda tez-tez uchraydigan shinglelarni yo'q qilish uchun takrorlash haqida hech narsa aytmasdan, umumiy HTML teglari va butun sonlarni shingle hisoblashdan olib tashlashimiz mumkin. Keyinchalik o'xshash hujjatlarni o'z ichiga olgan klasterlarni yaratish uchun birlashmani topish algoritmidan foydalaniladi. Buning uchun hal qiluvchi bosqichni bajarishimiz kerak: eskizlar to'plamidan $\mathbf{d_i}$ va $\mathbf{d_j}$ o'xshash bo'ladigan **i, j** juftliklari to'plamiga o'tish.

Shu maqsadda eskizlari umumiy a'zolarga ega bo'lgan har qanday hujjat juftligi uchun umumiy shingillalar sonini hisoblaymiz. $< x_i^{\pi}, d_i >$ ro'yxatini x_i^{π} juftliklari bo'yicha tartiblashdan boshlaymiz. Har bir x_i^{π} uchun endi barcha **i, j** juftlarini hosil qilishimiz mumkin, ular uchun ularning ikkala eskizida x_i^{π} mavjud. Ulardan har bir **i, j** juftligi uchun nolga teng bo'lmagan eskiz bilan o'zaro bog'liq bo'lgan umumiy x_i^{π} qiymatlari sonini hisoblashimiz mumkin. Oldindan o'rnatilgan chegarani qo'llash orqali qaysi **i, j** juftlarining eskizlari bir-biriga o'xshashligini bilib olamiz. Misol uchun, agar chegara 80% bo'lsa, har qanday **i, j** uchun hisoblash kamida 160 bo'lishi kerak. Bunday juftlarni aniqlaganimizda, hujjatlarni deyarli takroriy "sintaktik klasterlarga" guruhlash uchun union-find dasturini ishga tushiramiz. Bu mohiyatan 6.2-bo'limda kiritilgan yagona bo'g'inli klasterlash algoritmining variantidir. Bitta oxirgi hiyla **i, j** juftliklari uchun $|\mathbf{p_{si}} \cap \mathbf{p_{sj}}|$/200 ni hisoblashda zarur bo'lgan bo'sh joyni qisqartiradi, bu esa printsipial jihatdan hujjatlar sonida kvadratik bo'shliqni talab qilishi mumkin. Eskizlarida bir nechta umumiy shingillalarga ega bo'lgan **i, j** juftliklarini e'tibordan chetda qoldirish uchun har bir hujjatga eskizni quyidagicha qayta ishlaymiz: eskizdagi x_i^{π} ni saralaymiz, so'ngra ushbu tartiblangan ketma-ketlikni shingillalar to'plamini har bir hujjatni yaratish uchun ajratamiz. Agar ikkita xujjatda umumiy super shingil bo'lsa, $|\psi_i \cap \psi_j|$ 200 ning aniq qiymatini hisoblashga o'tamiz. Bu yana evristik, lekin eskizning bir-biriga mos kelishi hisobini to'playdigan **i,j** juftlik sonini qisqartirishda juda samarali bo'lishi mumkin.

8.8-mashq. **A** va **B** veb-qidiruv tizimlari har biri Internetning bir xil o'lchamdagi tasodifiy kichik to'plamini skanerlaydi. Ko'rib chiqilgan

sahifalarning Ba'zilari dublikatdir - turli URL manzillarida bir-birining aniq matnli nusxalaridir. Dublikatlar **A** va **B** tomonidan skanerlangan sahifalar orasida bir xilda taqsimlangan deb faraz qilaylik. Bundan tashqari, dublikat aynan ikki nusxaga ega bo'lgan sahifadir, deb faraz qiling - hech bir sahifada ikkitadan ortiq nusxa yo'q. **A** sahifalarni takroriy yo'q qilmasdan indekslaydi, **B** esa har bir takroriy sahifaning faqat bitta nusxasini indekslaydi. Ikki tasodifiy kichik to'plamlar takroriy yo'q qilishdan oldin bir xil o'lchamga ega. Agar **A** ning indekslangan URL manzillarining 45% **B** indeksida, **B** ning indekslangan URL manzillarining 50% **A** indeksida mavjud bo'lsa, Webning qaysi qismi dublikati bo'lmagan sahifalardan iborat?

8.9-mashq. 8.8-rasmda ko'rsatilgan jarayondan foydalanish o'rniga, ikkita S_1 va S_2 to'plamlari orasidagi o'zaro bog'lanishning ***Jakkard koeffitsientini*** baholash uchun quyidagi jarayonni ko'rib chiqing. S_1 va S_2 bo'lgan koinot elementlarining tasodifiy kichik to'plamini tanlaymiz. Chizilgan bo'lishi ham mumkin. Bu isbotda **A** matritsa satrlarining tasodifiy kichik to'plamini tanlashga to'g'ri keladi. Ushbu tasodifiy kichik to'plamlarning ***Jakkard koeffitsientini*** to'liq hisoblaymiz. Nima uchun bu taxmin S_1 va S_2 uchun ***Jaccard koeffitsientining xolis baholovchisi*** hisoblanadi?

8.10-mashq. Nima uchun bu baholovchini amalda qo'llash juda qiyin bo'lishini tushuntiring.

8- bob bo'yicha foydalanilgan adabiyotlar

Sakai, Tetsuya.
2007.
On the reliability of information retrieval metrics based on graded relevance.
IP&M 43 (2): 531-548.

Salton, Gerard.
1971a.
Cluster search strategies and the optimization of retrieval effectiveness.
In *The SMART Retrieval System - Experiments in Automatic Document Processing* Salton (1971b), pp. 223-242.
Salton, Gerarded.).

1971b.
The SMART Retrieval System - Experiments in Automatic Document Processing.
Prentice Hall.

Salton, Gerard.
1975.
Dynamic information and library processing.
Prentice Hall.

Salton, Gerard.
1989.
Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.
Addison Wesley.

Salton, Gerard.
1991.
The Smart project in automatic document retrieval.
In *Proc. SIGIR*, pp. 356-358. ACM Press.

Salton, Gerard, James Allan, and Chris Buckley.
1993.
Approaches to passage retrieval in full text information systems.
In *Proc. SIGIR*, pp. 49-58. ACM Press.
DOI: *doi.acm.org/10.1145/160688.160693*.

8- bob bo'yicha nazariy va amaliy test savollari

1. Fayl tizimlaridan foydalanishning quyidagi kamchiliklarini bartaraf etish uchun ma'lumotlar bazasi ilovalari to'g'ridan-to'g'ri fayl tizimining ustiga qurilgan: **(i)** ma'lumotlarning ortiqcha va nomuvofiqligi **(ii)** ma'lumotlarga kirishdagi qiyinchilik **(iii)** ma'lumotlarni izolyatsiyasi **(iv)** yaxlitlik muammolari
   A) (i), (ii), (iii) va (iv)
   B) (i), (ii) va (iii)
   C) (i) va (iv)
   D) (i)
2. Qaysi RAID darajasi ikki marta taqsimlangan paritet bilan blok darajasidagi chiziqni beradi?
   A) RAID 6
   B) RAID 2

C) RAID 5
D) RAID 10

3. Ta'minlash uchun qanday hollarda disklarning reyd konfiguratsiyasi qo'llaniladi?
A) B va C
B) Xatolarga chidamlilik
C) Yuqori tezlik
D) Yuqori ma'lumotlar zichligi

4. Fayl bloklarining indekslangan sxemasida faylning mumkin bo'lgan maksimal hajmi quyidagilarga bog'liq:
A) Indeks uchun ishlatiladigan bloklar soni va indeks hajmi
B) Bloklarning o'lchami va manzilning o'lchami
C) Indeks hajmi
D) Blok hajmi

5. Fayl tizimida disk bloklarini ajratish uchun _____ ajratish usulida faylga bloklarni kiritish va o'chirish oson.
A) Bog'langan
B) Indeks
C) Qo'shni
D) Bit xaritasi

6. Operatsion nazorat qilish uchun zarur bo'lgan ma'lumotlarni yig'ish darajasi qanday?
A) Batafsil
B) Agregat
C) Sifatli
D) To'g'ri javob yo'q

7. Katalogni fayl nomlarini katalog yozuvlariga aylantiruvchi sifatida ko'rish mumkin.
A) Belgilar jadvali
B) Bo'lim
C) Joyni almashtirish
D) kesh

8. Quyidagilardan qaysi biri zich indeks hisoblanadi?
A) Ikkilamchi indeks
B) Birlamchi indeks
C) Klasterlar indeksi
D) Ikkilamchi kalit bo'lmagan indeks

9. Ieraxik ma'lumotlar bazasida _____ manzilini aniqlash uchun xeshlash funksiyasidan foydalaniladi.
A) Ildiz
B) To'qnashuv
C) Begona kalit
D) Yozuvlar

10. Har bir munosabatni alohida operatsion tizim faylida saqlaydigan ma'lumotlar bazasi tizimlari o'zlari maxsus sxemani belgilash o'rniga operatsion tizimning avtorizatsiya sxemasidan foydalanishlari mumkin. Bunday holda, qaysi biri noto'g'ri?
A) Avtorizatsiya jarayoni operatsion tizim darajasida amalga oshirilganligi sababli ma'lumotlar bazasidagi operatsiyalar tezlashadi.
B) Administrator grant variantini ko'proq nazorat qiladi
C) Avtorizatsiyalarni yangilash, o'chirish va kiritish o'rtasida farqlash qiyin.
D) Faylda bir nechta munosabatni saqlay olmaydi.

IX BOB. VEB SKANERLASH VA INDEKSLAR

9.1. Umumiy ko‘rinish

Veb skanerlash - ularni indekslash va qidiruv tizimini qo‘llab-quvvatlash uchun Internetdan sahifalarni yig‘ish jarayonidir. Tekshirishning maqsadi iloji boricha ko‘proq foydali veb-sahifalarni va ularni o‘zaro bog‘laydigan havolalar tuzilmasi bilan tez va samarali to‘plashdir. 8-bobda millionlab muvofiqlashtirilmagan shaxslar tomonidan yaratilgan Internetning murakkabliklarini o‘rganib chiqdik. Ushbu bobda Internetni skanerlashda yuzaga keladigan qiyinchiliklarni o‘rganamiz. Ushbu bobning diqqat markazida veb-brauzer sifatida 8.7-rasmda ko‘rsatilgan komponent bo‘ladi va uni ba’zan o‘rgimchak deb ham atashadi. Ushbu bobning maqsadi to‘liq miqyosli tijorat veb-qidiruv tizimi uchun brauzerni qanday yaratishni tasvirlash emas. Buning o‘rniga talaba loyihasi miqyosidan tortib, muhim tadqiqot loyihalarigacha bo‘lgan umumiy bo‘lgan bir qator masalalarga e’tibor qaratamiz. Veb-brauzerlar uchun desiderata ro‘yxatidan boshlaymiz (9.1- bo‘lim) va keyin 9.2-bo‘limda ushbu muammolarning har biri qanday hal qilinishini muhokama qilinadi. Ushbu bobning qolgan qismida ushbu xususiyatlarni qondiradigan tarqatilgan veb-brauzerning arxitekturasi va ba’zi amalga oshirish tafsilotlari tasvirlangan. 9.3-bo‘limda veb-miqyosda amalga oshirish uchun ko‘plab mashinalar bo‘ylab indekslarni taqsimlash muhokama qilinadi.

9.1.1. Brauzer taqdim etishi kerak bo‘lgan xususiyatlar

Veb-brauzerlar uchun talablarni ikkita toifaga ajratamiz. Veb-brauzerlar taqdim etishi kerak bo‘lgan xususiyatlar. Barqarorlik: Internetda o‘rgimchak tuzoqlarini yaratuvchi serverlar mavjud bo‘lib, ular brauzerlarni ma’lum bir domendagi cheksiz sonli sahifalarni olishda tiqilib qolishga undaydigan veb-sahifalar generatorlaridir. *Crawlerlar* bunday tuzoqlarga chidamli bo‘lishi uchun mo‘ljallangan bo‘lishi kerak. Bunday tuzoqlarning hammasi ham zararli emas. Ba’zilari noto‘g‘ri veb-sayt ishlab chiqishning tasodifiy yomon ta’siridir. Xushmuomalalik: Veb-serverlarda brauzerning ularga tashrif buyurish tezligini tartibga soluvchi

ham yashirin, ham aniq siyosatlar mavjud. Ushbu xushmuomalalik siyosatlariga rioya qilish kerak.

9.1.2. Brauzer taqdim etishi kerak bo‘lgan xususiyatlar

Tarqalgan: **Crawler** bir nechta mashinalar bo‘ylab taqsimlangan tarzda bajarish qobiliyatiga ega bo‘lishi kerak.

Kengaytirilishi mumkin: brauzer arxitekturasi qo‘shimcha mashinalar va tarmoqni kengligi qo‘shish orqali skanerlash tezligini oshirishga imkon berishi kerak.

Ishlash va samaradorlik: skanerlash tizimi turli xil tizim resurslaridan, jumladan protsessor, saqlash va tarmoq o‘tkazish qobiliyatidan samarali foydalanishi kerak.

Sifat: barcha veb-sahifalarning muhim qismi foydalanuvchi so‘rovlari ehtiyojlarini qondirish uchun yomon yordam ekanligini hisobga olsak, brauzer birinchi navbatda “foydali” sahifalarni olishga moyil bo‘lishi kerak.

Yangilik: ko‘pgina ilovalarda brauzer uzluksiz rejimda ishlashi kerak: u ilgari olingan sahifalarning yangi nusxalarini olishi kerak. Misol uchun, qidiruv tizimining qidiruvi qidiruv tizimining indeksida har bir indekslangan veb-sahifaning juda dolzarb ko‘rinishini o‘z ichiga olishiga ishonch hosil qilishi mumkin. Bunday doimiy skanerlash uchun brauzer ushbu sahifaning o‘zgarish tezligiga yaqin bo‘lgan chastotali sahifani skanerlay olishi kerak.

Kengaytiriladigan: **Crawlerlar** ko‘p jihatdan kengaytiriladigan bo‘lishi kerak - yangi ma’lumotlar formatlari, yangi olish protokollari va boshqalar bilan ishlash uchun. Bu brauzer arxitekturasining modulli bo‘lishini talab qiladi.

9.2. Skaynerlash

Har qanday gipermatnli skanerning asosiy ishi (Internet, intranet yoki boshqa gipermatnli hujjatlar to‘plami uchun) quyidagicha. Brauzer fayllar to‘plamini tashkil etuvchi bir yoki bir nechta URL bilan boshlanadi. U ushbu to‘plamdan URLni tanlaydi, so‘ngra veb-sahifani o‘sha URL manzilidan oladi. Keyin olingan sahifa matnini ham, sahifadagi havolalarni ham ajratib olish uchun tahlil qilinadi (ularning har biri

boshqa URL manziliga ishora qiladi). Olingan matn matn indeksatoriga beriladi (birinchi qismning 4 va 5-boblarda tasvirlangan). Olingan havolalar (URLlar) keyin URL chegarasiga qo'shiladi, u har doim tegishli sahifalari brauzer tomonidan olib tashlanmagan URL manzillaridan iborat. Dastlab, URL chegarasi to'plamini o'z ichiga oladi. Sahifalar olinayotganda, tegishli URL manzillar URL chegarasidan o'chiriladi. Butun jarayonni veb-grafikdan o'tish sifatida ko'rish mumkin (8-bobga qarang). Uzluksiz skanerlashda olingan sahifaning URL manzili kelajakda yana olish uchun chegaraga qo'shiladi. Veb-grafikaning oddiy ko'rinadigan rekursiv o'tishi amaliy veb-skanerlash tizimiga qo'yiladigan ko'plab talablar bilan murakkablashadi: brauzer yuqori sifatli sahifalarni olishda tarqatilgan, kengaytiriladigan, samarali, xushmuomala, mustahkam va kengaytiriladigan bo'lishi kerak. Ushbu masalalarni har birining ta'siri ko'rib chiqiladi. Ularning davolashimiz bir qator tadqiqot va tijorat brauzerlarining asosini tashkil etgan ***Mercator*** brauzerining dizayniga amal qiladi. Malumot nuqtasi sifatida, bir oylik skanerlashda milliard sahifani (hozirgi statik Internetning kichik bir qismi) olish har soniyada bir necha yuz sahifani olishni talab qiladi. Ushbu olish tezligiga erishish uchun umumiy brauzer tizimidagi bir nechta to'siqlarni hal qilish uchun ko'p tarmoqli dizayndan qanday foydalanish ko'rib chiqiladi. Ushbu batafsil tavsifga o'tishdan oldin, har qanday noprofessional brauzer qondirishi kerak bo'lgan ba'zi bir asosiy xususiyatlarga ega brauzerlarni yaratishga urinishi mumkin bo'lgan o'quvchilar uchun yana bir bor takrorlaymiz:

1. Bir vaqtning o'zida har qanday xost uchun faqat bitta ulanish ochiq bo'lishi kerak.

2. Xostga ketma-ket so'rovlar o'rtasida bir necha soniya kutish vaqti bo'lishi kerak.

3. 9.2.1-bo'limda tavsiflangan xushmuomalalik cheklovlariga rioya qilish kerak.

9.2.1. Crawler arxitekturasi

Yuqorida keltirilgan oddiy sxema 9.1-rasmda ko'rsatilganidek, bir-biriga mos keladigan bir nechta modullarni talab qiladi.

1. Joriy skanerlashda hali olinmagan URL manzillarini o'z ichiga olgan URL chegarasi (doimiy skanerlashda URL avvaldan olingan

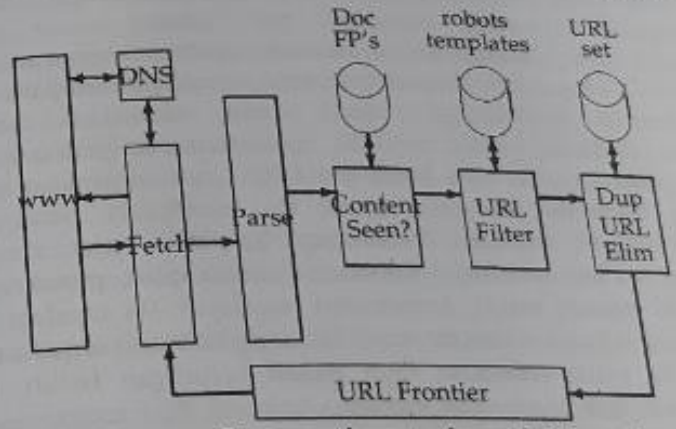
bo'lishi mumkin, lekin qayta olish uchun chegarada joylashgan bo'lishi kerak). Buni 9.2.3-bo'limda batafsilroq tasvirlab beramiz.

2. URL orqali ko'rsatilgan sahifani olish uchun veb-serverni aniqlaydigan DNS rezolyutsiyasi moduli aniqlanadi. Bu 9.2.2-bo'limda batafsil bayon qilinadi.

3. URL manzilidagi veb-sahifani olish uchun http protokolidan foydalanadigan qabul qilish moduli ishlab chiqilishi kerak.

4. Olingan veb-sahifadan matn va havolalar to'plamini ajratib oluvchi tahlil qilish moduli ishlab chiqiladi.

5. Chiqarilgan havola allaqachon URL chegarasida yoki yaqinda olinganligini aniqlaydigan takroriy o'chirish moduli ishlab chiqiladi.



9.1-rasm. **Brauzerning asosiy arxitekturasi**

Tekshiruv 9.1-rasmdagi mantiqiy sikl bo'ylab aylanib o'tadigan birdan potentsial yuzlargacha oqimlar tomonidan amalga oshiriladi. Ushbu oqimlar bitta jarayonda ishga tushirilishi yoki taqsimlangan tizimning turli tugunlarida ishlaydigan bir nechta jarayonlarga bo'linishi mumkin. Ular URL chegarasi joyida va bo'sh emas deb faraz qilishdan boshlaymiz va URL chegarasini amalga oshirish tavsifini 9.2.3-bo'limga qoldiramiz. Bitta URL manzilini olish, turli tekshiruvlar va filtrlardan o'tish, so'ng nihoyat (doimiy skanerlash uchun) URL chegarasiga qaytarilish sikli orqali kuzatib boramiz. Ko'rib chiquvchi oqim chegaradan URL olish va odatda http protokolidan foydalangan holda ushbu URL manzilidagi veb-sahifani olish bilan boshlanadi. Keyin

olingan sahifa vaqtinchalik do‘konga yoziladi va u yerda bir qator operatsiyalar bajariladi. Keyinchalik, sahifa tahlil qilinadi va matn va undagi havolalar chiqariladi. Matn (har qanday teg ma’lumotlari bilan - masalan, qalin harflar bilan atamalar) indekslovchiga uzatiladi. Bog‘lanish ma’lumotlari, shu jumladan langar matni, shuningdek, 10-bobda tavsiflangan usullarda tartiblashda foydalanish uchun indeksatorga uzatiladi. Bundan tashqari, har bir ajratilgan havola havolani URL chegarasiga qo‘shish kerakligini aniqlash uchun bir qator testlardan o‘tadi. Birinchidan, oqim xuddi shu tarkibga ega veb-sahifa boshqa URL manzilida ko‘rilganligini tekshiradi. Buning uchun eng oddiy dastur nazorat summasi kabi oddiy barmoq izini ishlatishi mumkin (9.1-rasmda ta’riflanganidek, yanada murakkab test barmoq izlari o‘miga shingillalardan foydalanadi. Keyin, bir nechta testlardan biriga asoslanib, olingan URL chegaradan chiqarib tashlanishi kerakligini aniqlash uchun URL filtridan foydalaniladi. Misol uchun, skanerlash muayyan domenlarni (masalan, barcha .com URL manzillarini) chiqarib tashlashga harakat qilishi mumkin - bu holda test URL .com domenidan bo‘lsa, shunchaki filtrlaydi. Shunga o‘xshash test eksklyuziv emas, balki inklyuziv bo‘lishi mumkin. Internetdagi ko‘plab xostlar o‘z veb-saytlarining ma’lum qismlarini Robotlarni istisno qilish protokoli deb nomlanuvchi standart ostida skanerlashni taqiqlaydi. Bu saytdagi URL ierarxiyasining ildiziga ***robots.txt*** nomli faylni joylashtirish orqali amalga oshiriladi. Bu yerda ***robots.txt*** fayli misoli keltirilgan bo‘lib, unda “***searchengine***” deb nomlangan robotdan tashqari, fayl ierarxiyasidagi o‘rni /yoursite/temp/ bilan boshlanadigan hech qanday robot URL manziliga tashrif buyurishi kerak emasligini ko‘rsatadi.

```
User – agent : *
Disallow : / yoursite / temp /

User – agent : searchengi ne
Disallow :
```

Ko‘rib chiqilayotgan URL robot cheklovlaridan o‘tishini tekshirish uchun **robots.txt** fayli veb-saytdan olinishi kerak va shuning uchun URL chegarasiga qo‘shilishi mumkin. Chegaraga qo‘shiladigan har bir URLda sinovdan o‘tkazish uchun uni qaytadan olish o‘rniga, xost uchun faylning yaqinda olingan nusxasini olish uchun keshdan foydalanish mumkin. Bu,

ayniqsa, juda muhim chunki sahifadan olingan havolalarning ko‘pchiligi sahifa olingan xostga to‘g‘ri keladi va shuning uchun xostning **robots.txt** faylida tekshirilishi mumkin. Shunday qilib, havolani chiqarish jarayonida filtrlashni amalga oshirgan holda, **robots.txt** fayllarini sinab ko‘rishimiz kerak bo‘lgan xostlar oqimida ayniqsa yuqori joylashuvga ega bo‘lar edik, bu esa yuqori kesh tezligiga olib keladi. Afsuski, bu veb-boshqaruvchilarning xushmuomalalik talablarini buzadi. URL manzili (ayniqsa, sifatsiz yoki kamdan-kam o‘zgaruvchan hujjatga ishora qiluvchi) bir necha kun yoki hatto haftalar davomida chegarada bo‘lishi mumkin. Agar bunday URL manzilini chegaraga qo‘shishdan oldin robotlar filtrlashdan o‘tadigan bo‘lsa, uning **robots.txt** fayli URL chegaradan olib tashlangan va olingan vaqtgacha o‘zgarishi mumkin edi. Shunday qilib, veb-sahifani olishga urinishdan oldin darhol robotlarni filtrlashni amalga oshirishimiz kerak. Ma’lum bo‘lishicha, **robots.txt** fayllari keshini saqlash hali ham juda samarali. URL chegarasidan ajratilgan URL-manzillar oqimida ham yetarli joylashuv mavjud.

Keyinchalik, URL quyidagi ma’noda normallashtirilishi kerak: ko‘pincha veb-sahifadagi havolani kodlash **p** sahifaga nisbatan ushbu havolaning maqsadini ko‘rsatadi. Shunday qilib, ***en.wikipedia.org/wiki/Main_Page*** sahifasining HTML-da shunday kodlangan nisbiy havolasi mavjud:

```
page en.wikipedia org / wiki / Main _ Page :

< a href =" / wiki / Wikipedia :General _ disclaimer " title =" Wikipedia :General
disclaimer "> Disclaimer s < / a >
```

Nihoyat, URL takroriy yo‘q qilish uchun tekshiriladi: agar URL allaqachon chegarada bo‘lsa yoki (uzluksiz skanerlashda) allaqachon skanerlangan bo‘lsa, uni chegaraga qo‘shmaymiz. URL chegaraga qo‘shilsa, unga ustuvorlik beriladi, buning asosida u oxir-oqibat olish uchun chegaradan olib tashlanadi. Ushbu ustuvor navbatning tafsilotlari 9.2.3-bo‘limda keltirilgan. Ba’zi uy ishlarini bajarish odatda maxsus oqim tomonidan amalga oshiriladi. Bu oqim odatda jim turadi, bundan tashqari u bir necha soniyada bir necha soniyalarda skanerlash jarayoni statistikasini (skanerlangan URL manzillar, chegara o‘lchami va h.k.) qayd qilish uchun uyg‘onadi, skanerlashni to‘xtatish yoki (bir necha

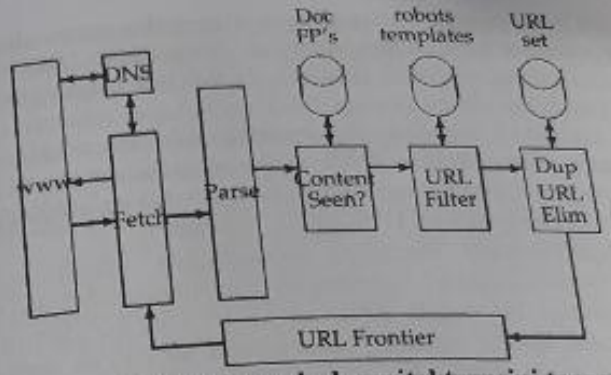
soatda bir marta skanerlash) tekshiruvini o'tkazishni hal qiladi. Tekshirish punktida skaner holatining surati (aytaylik, URL chegarasi) diskda saqlanadi. Ko'zdan kechiruvchi katastrofik nosozlik bo'lsa, skanerlash eng so'nggi nazorat punktidan qayta boshlanadi.

Brauzerni tarqatish

Skanerdagi oqimlar har xil jarayonlar ostida ishlashi mumkinligini aytib o'tdik, ularning har biri taqsimlangan skanerlash tizimining boshqa tugunida ham ishlaydi. Bunday taqsimlash masshtablash uchun zarur. Bundan tashqari, har bir tugun o'z "yaqinidagi" xostlarni taraydigan geografik taqsimlangan brauzer tizimida ham qo'llanilishi mumkin. Ko'zdan kechirilayotgan xostlarni brauzer tugunlari orasida bo'lish xesh funksiyasi yoki maxsus moslashtirilgan siyosat yordamida amalga oshirilishi mumkin. Misol uchun, Yevropa domenlariga e'tibor qaratish uchun Yevropada brauzer tugunini topishimiz mumkin, garchi bu bir necha sabablarga ko'ra ishonchli bo'lmasa-da - paketlarning Internet orqali o'tadigan marshrutlari har doim ham geografik yaqinlikni va har qanday holatda ham xost domenini aks ettirmaydi. Har doim ham uning jismoniy joylashishini aks ettirmaydi. Tarqalgan brauzerning turli tugunlari URL manzillarini qanday bog'laydi va almashadi? G'oya 9.1-rasmdagi oqimni har bir tugunda takrorlashdan iborat, bunda bitta muhim farq bor: URL filtridan so'ng jo'natish uchun xost ajratgichdan foydalaniladi.

URL uchun mas'ul bo'lgan brauzer tugunida omon qolgan har bir URL bilan ishlash mumkin. Shunday qilib, tekshirilayotgan xostlar to'plami tugunlar orasida bo'linadi. Ushbu o'zgartirilgan oqim 9.2-rasmda ko'rsatilgan. Xost splitterning chiqishi taqsimlangan tizimdagi bir-birining tugunlarining ***Duplicate URL Eliminator*** blokiga kiradi. "Ko'rilgan tarkib?" 9.2-rasmdagi taqsimlangan arxitektura moduli bir qancha omillar bilan murakkablashadi:

1. URL chegarasi va dublikatni yo'q qilish modulidan farqli o'laroq, hujjat barmoq izlari/shingles xost nomiga qarab bo'linib bo'lmaydi. Bir xil (yoki juda o'xshash) kontentning turli veb-serverlarda paydo bo'lishiga hech narsa to'sqinlik qilmaydi. Shunday qilib, barmoq izlari/shingillalar to'plami barmoq izi/shingilning Ba'zi xususiyatlariga asoslangan holda tugunlar bo'ylab bo'linishi kerak (aytaylik, barmoq izi modulini tugunlar sonini olish orqali).



9.2-rasm. **Asosiy skanerlash arxitekturasini tarqatish**

Bu joylashuv nomuvofiqligi natijasida eng ko'p "Ko'rilgan kontent?" testlar masofaviy protsedura chaqiruviga olib keladi (garchi ommaviy qidirish so'rovlarini olish mumkin bo'lsa ham).

2. Hujjatlarning barmoq izlari/shingillalar oqimida juda kam joylashadi. Shunday qilib, mashhur barmoq izlarini keshlash yordam bermaydi (chunki mashhur barmoq izlari yo'q).

3. Hujjatlar vaqt o'tishi bilan o'zgarib turadi va shuning uchun uzluksiz skanerlash kontekstida ularning eskirgan barmoq izlarini/shillinglarini kontent ko'rilgan to'plam(lar)dan o'chirib tashlashimiz kerak. Buni amalga oshirish uchun URL chegarasida hujjatning barmoq izini/shillingni URL manzili bilan birga saqlash kerak.

9.2.2. DNS ruxsati

Har bir veb-server (haqiqatan ham internetga ulangan har qanday xost) o'ziga xos IP-manzilga ega: to'rt baytdan iborat ketma-ketlik odatda nuqta bilan ajratilgan to'rtta butun son sifatida ifodalanadi, masalan, ***207.142.131.248*** - ***www.wikipedia.org*** xostiga bog'langan raqamli IP ***manzil***. Matn shaklida ***www.wikipedia.org*** kabi **URL** manzili berilgan bo'lsa, uni **IP** manzilga tarjima qilish (*207.142.131.248*) *DNS* ***rezolyutsiyasi*** yoki ***DNS qidiruvi*** deb nomlanuvchi jarayondir. Bu yerda **DNS** asosiy nom xizmatini anglatadi. DNS rezolyutsiyasi vaqtida ushbu tarjimani amalga oshirmoqchi bo'lgan dastur (ularning holatida, veb-

brauzerning komponenti) tarjima qilingan IP manzilini qaytaradigan DNS serveriga murojaat qiladi. (Amalda butun tarjima bitta DNS serverida amalga oshirilmasligi mumkin, aksincha, dastlab bog'langan *DNS server* tarjimani yakunlash uchun boshqa DNS serverlarini rekursiv ravishda chaqirishi mumkin.) *En.wikipedia.org/wiki/* kabi murakkabroq URL uchun *Domain_Name_System, DNS rezolyutsiyasi* uchun mas'ul bo'lgan brauzer komponenti xost nomini chiqaradi - bu holda *en.wikipedia.org* - va *en.wikipedia.org* xostining **IP** manzilini qidiradi.

DNS rezolyutsiyasi internetni skanerlashda taniqli muammo hisoblanadi. Domen nomlari xizmatining taqsimlangan tabiati tufayli, DNS rezolyutsiyasi bir nechta so'rovlar va Internet bo'ylab aylanma sayohatlarni talab qilishi mumkin, bu soniyalar va ba'zan undan ham ko'proq vaqtni talab qiladi. Bu darhol bir soniyada bir necha yuz hujjatlarni olish maqsadimizni xavf ostiga qo'yadi. Standart chora keshlashni joriy qilishdir. Yaqinda DNS qidiruvlarini amalga oshirgan URL manzillar DNS keshida topildi, bu esa internetdagi DNS serverlariga o'tish zaruratidan qochishni taqazo etadi. Biroq, xushmuomalalik cheklovlariga rioya qilish (9.2.3-bo'limga qarang) keshga kirish tezligini cheklaydi. DNS rezolyutsiyasida yana bir muhim qiyinchilik mavjud. Standart kutubxonalardagi qidirish ilovalari (ehtimol, brauzerni ishlab chiquvchi har bir kishi tomonidan ishlatilishi mumkin) odatda sinxrondir. Bu shuni anglatadiki, domen nomi xizmatiga so'rov yuborilgandan so'ng, birinchi so'rov tugaguniga qadar ushbu tugundagi boshqa brauzerning pararllel oqimlari bloklanadi. Buni chetlab o'tish uchun ko'pchilik veb-brauzerlar brauzerning tarkibiy qismi sifatida o'zlarining DNS-rezolyutsiyalarini amalga oshiradilar. Rezolyutor kodini bajarayotgan mavzu DNS serveriga xabar yuboradi va keyin vaqtli kutishni amalga oshiradi: u boshqa oqim tomonidan signal berilganda yoki belgilangan vaqt kvanti tugashi bilan davom etadi. Yagona, alohida DNS oqimi nom xizmatidan kiruvchi javob paketlari uchun standart DNS portida (port 53) tinglanadi. Javob olgandan so'ng, u tegishli skaner oqimiga signal beradi (bu holda, i) va agar i hali davom etmagan bo'lsa, unga javob paketini beradi chunki uning ***vaqt kvantining muddati*** tugagan. ***Kutish vaqti kvantining muddati*** tugaganligi sababli davom etuvchi skaner oqimi belgilangan miqdordagi urinishlar uchun qayta urinadi, DNS serveriga yangi xabar yuboradi va har safar vaqtli kutishni amalga oshiradi. ***Mercator dizaynerlari*** beshta urinish tartibini tavsiya qiladi. ***Kutishning vaqt kvanti*** bu urinishlarning har biri bilan eksponent ravishda ortadi.

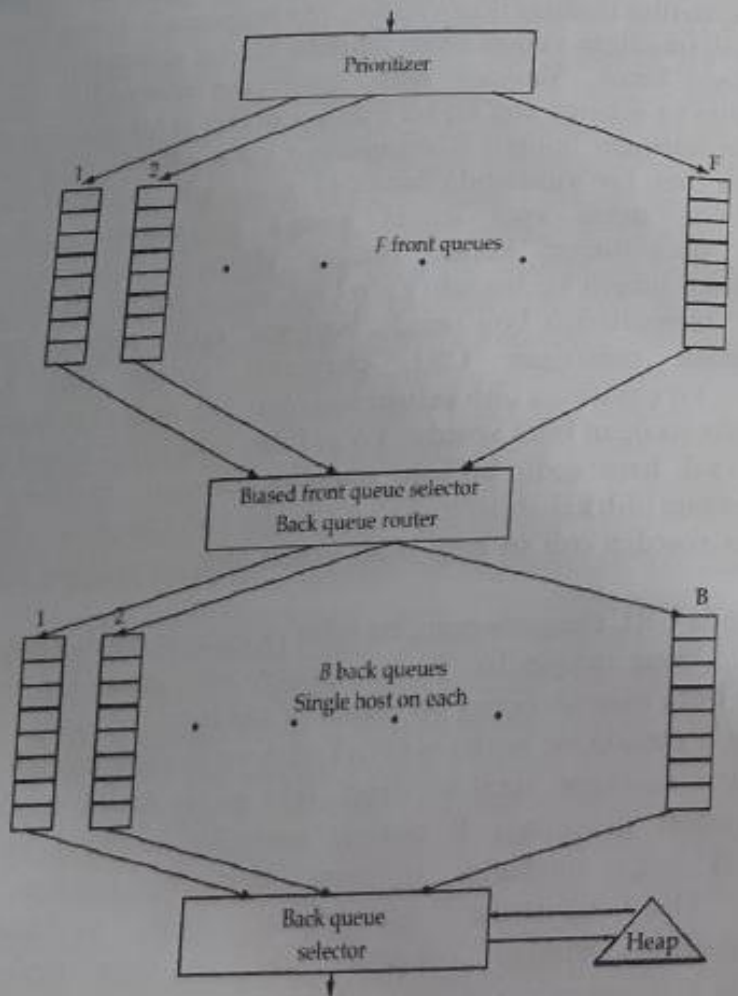
Mercator bir soniya bilan boshlandi va taxminan 90 soniya bilan yakunlandi chunki hal qilish uchun o'nlab soniyalar ketadigan xost nomlari mavjud.

9.2.3. URL chegarasi

Tugundagi URL chegarasi uning skanerlash jarayoni (yoki boshqa skanerlash jarayonining xostni ajratuvchisi tomonidan) orqali beriladi. U URL-manzillarni chegarada saqlaydi va tekshiruvchi oqim URL-manzilni qidirganda ularni qandaydir tartibda qaytaradi. URL-manzillar chegara tomonidan qaytarilish tartibini ikkita muhim fikr boshqaradi. Birinchidan, tez-tez o'zgarib turadigan yuqori sifatli sahifalar tez-tez skanerlash uchun ustuvor bo'lishi kerak. Shunday qilib, sahifaning ustuvorligi uning o'zgarish tezligi va sifatiga bog'liq bo'lishi kerak (Ba'zi bir oqilona sifat bahosidan foydalangan holda). Kombinatsiya zarur chunki ko'p sonli spam-sahifalar har bir yuklashda butunlay o'zgaradi. Ikkinchi fikr - xushmuomalalik: qisqa vaqt ichida xostga bo'ladigan takroriy so'rovlardan qochishimiz kerak. Buning ehtimoli mos yozuvlar joylashuvi shakli tufayli kuchayadi: ko'p URL manzillar bir xil xostdagi boshqa URL manzillariga bog'lanadi. Natijada, oddiy ustuvor navbat sifatida amalga oshirilgan URL chegarasi xostga olib kelish so'rovlarining ko'payishiga olib kelishi mumkin. Bu, agar ular brauzerni istalgan vaqtda istalgan bitta xostdan ko'pi bilan bitta oqim olishi uchun cheklab qo'ysak ham sodir bo'lishi mumkin. Umumiy evristika bu - xostga navbatdagi olib kelish so'rovlari o'rtasidagi bo'shliqni qo'shishdir, bu esa ushbu xostdan eng so'nggi olish uchun olingan vaqtdan kattaroq tartibdir.

9.3-rasmda URL chegarasining bir tekis va ustuvor amalga oshirilishi ko'rsatilgan. Uning maqsadlari (i) har qanday xost uchun bir vaqtning o'zida faqat bitta ulanish ochiq bo'lishini ta'minlash; (ii) xostga ketma-ket so'rovlar o'rtasida bir necha soniya kutish vaqti sodir bo'ladi va (iii) yuqori ustuvor sahifalar afzal ko'riladi. Ikki asosiy kichik modul - bu rasmning yuqori qismidagi **F** oldingi navbatlar to'plami va pastki qismidagi **B** orqa navbatlar to'plami; bularning barchasi *FIFO navbatlari*. Old navbatlar ustuvorlikni, orqa navbatlar esa xushmuomalalikni amalga oshiradi. Old va orqa navbatlardan o'tayotganda chegaraga qo'shilgan URL oqimida ustuvorlik beruvchi

avval URL manziliga 1 va **F** o'rtasidagi butun son ustuvorligini o'zining olib kelish tarixiga asoslanib (tezlikni hisobga olgan holda) belgilaydi. Bu URLdagi veb-sahifa oldingi skanerlashlar orasida o'zgargan). Masalan, tez-tez o'zgarib turadigan hujjatga ustuvorlik beriladi. Boshqa evristikalar dasturga bog'liq va aniq bo'lishi mumkin - masalan, yangiliklar xizmatlarining URL manzillari har doim eng yuqori ustuvorlikka ega bo'lishi mumkin. Endi u **i** ustuvorligi bilan tayinlanganligi sababli, URL endi oldingi navbatlarning **i**-siga qo'shiladi. **B** orqa navbatlarning har biri quyidagi o'zgarmaslikni saqlaydi: (i) skanerlash jarayonida u bo'sh emas va (ii) u faqat bitta xostning URL manzillarini o'z ichiga oladi.



9.3-rasm. **URL chegarasi**

Yordamchi jadval **T** (9.4-rasm) xostlardan orqa navbatlargacha bo'lgan xaritalash saqlash uchun ishlatiladi. Qachonki orqa navbat bo'sh bo'lsa va oldingi navbatdan qayta to'ldirilsa, **T** jadvali mos ravishda yangilanishi kerak. Xostlar soni **B** dan ancha yuqori deb taxmin qilinadi.

Allaqachon tekshirilgan sahifalardan olingan URL-manzillar rasmning yuqori qismida joylashgan. URL so'ragan skanerlash chizig'i uni rasmning pastki qismidan chiqaradi. Marshrutda URL skanerlash uchun uning ustuvorligini boshqaradigan bir nechta oldingi navbatlardan biri orqali oqib o'tadi, so'ngra brauzerning xushmuomalaligini boshqaradigan bir nechta orqa navbatlardan biri orqali oqadi.

Host	Back queue
stanford.edu	23
microsoft.com	47
acm.org	12

9.4-rasm. **Yordamchi xostlar o'rtasidagi navbatlar jadvaliga misol**

Bundan tashqari, ular har bir orqa navbat uchun bitta yozuvga ega bo'lgan to'plamni saqlaymiz, kirish bu navbatga mos keladigan xost bilan qayta bog'lanishi mumkin bo'lgan eng erta vaqt hisoblanadi. Chegaradan URL-manzilni so'ragan skaner oqimi ushbu to'plamning ildizini chiqaradi va (agar kerak bo'lsa) tegishli vaqt kiritilguncha kutadi. Keyin URL **u** ni orqa navbatning boshidagi **j** olingan yig'ma ildizga mos keladi va URL manzilini olishda davom etadi. **U** ni olgandan so'ng, chaqiruvchi oqim **j** ning bo'sh yoki yo'qligini tekshiradi. Agar shunday bo'lsa, u oldingi navbatni tanlaydi va boshidan URL **v** ni chiqaradi. Old navbatni tanlash ustuvorligi yuqori bo'lgan navbatlarga yo'naltirilgan (odatda tasodifiy jarayon orqali), bu yuqori ustuvorlikdagi URL manzillarning orqa navbatlarga tezroq tushishini ta'minlaydi. **v** ni hamda uning xostidan URL manzillarini ushlab turadigan orqa navbat mavjud yoki yo'qligini tekshiramiz. Agar shunday bo'lsa, **v** o'sha navbatga qo'shiladi va bo'sh **j** navbatiga kiritish uchun boshqa nomzod URL manzilini topish uchun oldingi navbatlarga qaytamiz. Bu jarayon **j** yana bo'sh bo'lmaguncha davom etadi. Har qanday holatda, oqim oxirgi olingan **j** dagi URL xossalariga (masalan, uning xostiga oxirgi marta qachon murojaat qilingani, shuningdek, oxirgi olish uchun sarflangan vaqt) asoslangan yangi eng erta vaqt **te** bilan **j** uchun yig'ma yozuvni kiritadi), keyin uni

qayta ishlashni davom ettiradi. Masalan, yangi kirish **te** joriy vaqt va oxirgi olish vaqtining o'n barobari bo'lishi mumkin.

Old navbatlar soni, ustuvorliklarni belgilash va navbatlarni tanlash siyosati bilan birgalikda tizimga kiritmoqchi bo'lgan ustuvor xususiyatlarni belgilaydi. Orqa navbatlar soni xushmuomalalikka rioya qilgan holda barcha skanerlash oqimlarini qanchalik band qilishimiz mumkinligini belgilaydi. ***Mercator*** dizaynerlari skaner oqimlari kabi uch baravar ko'p orqa navbatlarning qo'pol qoidasini tavsiya qiladi. Veb-miqyosda skanerlashda URL chegarasi tugunda mavjud bo'lganidan ko'ra ko'proq xotira talab qiladigan darajada o'sishi mumkin. Yechim URL chegarasining ko'p qismini diskda saqlashga ruxsat berishdir. Har bir navbatning bir qismi xotirada saqlanadi, ko'proq diskdan olib kelinadi chunki u xotiraga tushadi.

9.1-mashq. Nima uchun tarqatilgan skanerlash tizimining tugunlari o'rtasida xostlarni (alohida URL-manzillar o'rniga) bo'lish yaxshiroq?

9.2-mashq. Nima uchun xost ***splitteri Duplicate URL Eliminatordan*** oldin bo'lishi kerak?

9.3-mashq. Oldingi muhokamada tavsiya etilgan ikkita "qattiq konstanta"ga duch keldik - **te** o'sishi oxirgi olish vaqtidan o'n baravar, orqa navbatlar soni esa skanerlash oqimlari sonidan uch baravar ko'p. Bu ikki konstanta qanday bir-biriga bog'liq?

9.3. Indekslarni taqsimlash

Birinchi qismning 4.4-bo'limida taqsimlangan indekslashni tasvirlab berdik. Endi indeksning so'rovlarini qo'llab-quvvatlaydigan 2-sonli katta kompyuter klasteri bo'ylab taqsimlanishi ko'rib chiqiladi. Ikki aniq muqobil indeksni amalga oshirish o'zini ko'rsatadi. Atamalar bo'yicha bo'linish, shuningdek, global indeks tashkiloti sifatida ham tanilgan va hujjatlar bo'yicha bo'linish, shuningdek, mahalliy indeks tashkiloti sifatida ham tanilgan. Birinchisida, indeks atamalari lug'ati kichik to'plamlarga bo'linadi, har bir kichik to'plam tugunda joylashgan. Tugundagi shartlar bilan bir qatorda, ushbu shartlar uchun e'lonlarni saqlaymiz. So'rov uning so'rov shartlariga mos keladigan tugunlarga yo'naltiriladi. Asosan, bu ko'proq parallellikni ta'minlaydi chunki turli so'rov shartlariga ega so'rovlar oqimi turli xil mashinalar to'plamiga tegishli bo'lishi mumkin.

Amalda, indekslarni lug'at terminlari bo'yicha bo'lish ahamiyatsiz bo'lib chiqadi. Ko'p so'zli so'rovlar birlashish uchun tugunlar to'plamlari o'rtasida uzun e'lonlar ro'yxatini yuborishni talab qiladi va buning narxi ko'proq mos kelishidan ko'proq bo'lishi mumkin. Bo'limning vazifalarini muvozanatlash nisbiy atama chastotalarining ***apriori tahlili*** bilan emas, balki ***vaqt o'tishi*** bilan siljishi yoki to'satdan yo'qolishlarni ko'rsatishi mumkin bo'lgan so'rov shartlarini taqsimlash va ularning birgalikdagi hodisalari bilan boshqariladi.

Yaxshi bo'limlarga erishish so'rov shartlarining birgalikdagi funksiyasi bo'lib, miqdorini aniqlash oson bo'lmagan maqsadlarni optimallashtirish uchun atamalarni klasterlashni talab qiladi. Nihoyat, ushbu strategiya dinamik indekslashni amalga oshirishni qiyinlashtiradi. Hujjatlar bo'yicha bo'lish keng tarqalgan dasturdir: har bir tugun barcha hujjatlarning kichik to'plami uchun indeksni o'z ichiga oladi.

Har bir so'rov barcha tugunlarga taqsimlanadi, turli tugunlarning natijalari foydalanuvchiga taqdim etilishidan oldin birlashtiriladi. Ushbu strategiya ko'proq mahalliy disklarni kamroq tugunlararo aloqani qidiradi. Ushbu yondashuvdagi qiyinchiliklardan biri shundaki, ball qo'yishda foydalaniladigan global statistik ma'lumotlar (masalan, idf) butun hujjatlar to'plamida hisoblanishi kerak, garchi har qanday bitta tugundagi indeks faqat hujjatlarning kichik to'plamini o'z ichiga oladi. Vaqti-vaqti bilan tugun indekslarini yangi global statistik ma'lumotlar bilan yangilab turuvchi taqsimlangan "fon" jarayonlari orqali hisoblanadi.

Hujjatlarni tugunlarga bo'lish qanday hal qilinadi? 9.2.1-bo'limda brauzer arxitekturasini ishlab chiqishimizga asoslanib, bitta oddiy yondashuv barcha sahifalarni xostdan bitta tugunga tayinlash bo'ladi. Ushbu bo'linish xostlarning brauzer tugunlariga bo'linishini kuzatishi mumkin.

Bunday bo'linishning xavfli tomoni shundaki, ko'plab so'rovlarda natijalarning ustunligi kam sonli xostlar (va shuning uchun oz sonli indeks tugunlari)dagi hujjatlardan kelib chiqadi. Indeks tugunlari bo'shlig'iga har bir URLning xeshi so'rov vaqtini hisoblashning tugunlar bo'ylab bir xil taqsimlanishiga olib keladi. So'rov vaqtida so'rov tugunlarning har biriga uzatiladi, so'rov uchun eng yuqori **k** hujjatni topish uchun har bir tugunning yuqori **k** natijalari birlashtiriladi. Amalga oshirishning keng tarqalgan evristikasi hujjatlar to'plamini ko'pchilik so'rovlar bo'yicha yuqori ball olish ehtimoli yuqori bo'lgan hujjatlar indekslariga (masalan, 10-bobdagi usullardan foydalangan holda) va qolgan hujjatlar bilan past

ballli indekslarga bo'lishdir. Birinchi qismning 7.2.1-bo'limida tavsiflanganidek, yuqori ball ko'rsatkichlarida juda kam sonli moslik mavjud bo'lgandagina past ball indekslarini qidiramiz.

9.4. Ulanish serverlari

10-bobda aniqroq bo'lishi uchun veb-qidiruv tizimlari veb-grafikda tezkor ulanish so'rovlarini qo'llab-quvvatlaydigan ulanish serverini talab qiladi.

Odatda ulanish so'rovlari qaysi URL manzillar berilgan URL manziliga bog'lanadi? va berilgan URL qaysi URL-larga havola qiladi? Shu maqsadda xaritalarni URL dan tashqi havolalarga va URL dan ichki havolalarga saqlashni xohlaymiz.

Ilovalarga skanerlashni boshqarish, veb-grafik tahlili, murakkab skanerlashni optimallashtirish va havolalar tahlili kiradi (10-bobda muhokama qilinadi). Aytaylik, Internetda to'rt milliard sahifa bor edi, ularning har birida boshqa sahifalarga o'nta havola mavjud. Eng oddiy shaklda, har bir havolaning har bir uchini (manba va maqsadni) ko'rsatish uchun ular 32 bit yoki 4 baytni talab qiladi, jami $4 \times 10^9 \times 10 \times 8 = 10^{11}$ bayt xotira. Veb-grafikning

Ba'zi asosiy xususiyatlaridan ushbu xotira talabining 10% dan kamroq foydalanish uchun foydalanish mumkin. Bir qarashda, ularda ma'lumotlarni siqish muammosi bordek tuyuladi - bu turli xil standart yechimlarga mos keladi. Biroq, ularning maqsadimiz veb-grafani xotiraga sig'dirish uchun shunchaki siqish emas, buni ulanish so'rovlarini samarali qo'llab-quvvatlaydigan tarzda qilishimiz kerak. bu qiyinchilik indekslarni siqishni eslatadi (1-qism, 5-bob).

Har bir veb-sahifa noyob butun son bilan ifodalangan deb taxmin qilinadi.

Bu butun sonlarni tayinlash uchun ishlatiladigan maxsus sxema quyida tasvirlangan.

Teskari indeksga o'xshash qo'shnilik jadvalini tuzamiz: unda har bir veb-sahifa uchun qatorlar mavjud bo'lib, qatorlar mos keladigan butun sonlar bilan tartiblangan.

Har qanday **p** sahifasi uchun qator butun sonlarning tartiblangan ro'yxatini o'z ichiga oladi, ularning har biri **p** ga bog'langan veb-sahifaga mos keladi. Ushbu jadval ularga qaysi sahifalar **p** ga bog'langan shakl

so'rovlariga javob berishga imkon beradi? Shunga o'xshash tarzda jadvalni tuzamiz, uning yozuvlari **p** bilan bog'langan sahifalardir.

```
1: www.stanford.edu/alchemy
2: www.stanford.edu/biology
3: www.stanford.edu/biology/plant
4: www.stanford.edu/biology/plant/copyright
5: www.stanford.edu/biology/plant/people
6: www.stanford.edu/chemistry
```

9.5-rasm. **Leksikografik jihatdan tartiblangan URL manzillar to'plami**

Ushbu jadval ko'rinishi sodda ko'rinishda egallagan bo'sh joyni (bunda har bir havolani ikkita so'nggi nuqta bilan aniq ifodalaymiz, har biri 32 bitli butun son) 50% ga qisqartiradi. Quyidagi tavsifimiz har bir sahifadagi havolalar jadvaliga qaratiladi. Texnikalar har bir sahifaga havolalar jadvaliga ham tegishli ekanligi aniq bo'lishi kerak. Jadval uchun joyni yanada qisqartirish uchun bir nechta g'oyalardan foydalaniladi:

1. Ro'yxatlar o'rtasidagi o'xshashlik: Jadvalning ko'p qatorlarida umumiy ko'plab yozuvlar mavjud. Shunday qilib, agar bir nechta shunga o'xshash qatorlar uchun prototip qatorini aniq ifodalasak, qolgan qismini prototip qatorida qisqacha ifodalash mumkin.

2. Mahalliy: sahifadagi ko'plab havolalar "yaqin" sahifalarga o'tadi - masalan, bir xil xostdagi sahifalar. Bu shuni ko'rsatadiki, havolaning manzilini kodlashda ko'pincha kichik butun sonlardan foydalanishimiz va shu bilan joyni tejashimiz mumkin.

3. Tartiblangan ro'yxatlarda bo'shliq kodlashlaridan foydalaniladi: har bir havolaning manzilini saqlash o'rniga, satrdagi oldingi yozuvdan ofsetni saqlaymiz.

Endi ular ushbu texnikalarning har birini ishlab chiqamiz.

Barcha URL manzillarning leksikografik tartibida har bir URLni alfanumerik qator sifatida ko'rib chiqiladi va bu qatorlarni saralaymiz. 9.5-rasmda ushbu tartiblangan tartibning segmenti ko'rsatilgan. Haqiqiy leksikografik turdagi veb-sahifalar uchun ***www.stanford.edu*** sayti ***edu.stanford.www*** bo'lishi uchun URLning domen nomi qismi teskari bo'lishi kerak, lekin bu yerda bu shart emas, chunki asosan mahalliy havolalar bilan bog'liqmiz. Har bir URL manziliga ushbu tartibdagi o'rnini yagona identifikatsiyalovchi butun son sifatida belgilanadi. 9.6-

rasmda bunday raqamlashning namunasi va natijaviy jadvali ko'rsatilgan. Ushbu misol ketma-ketlikda ***www.stanford.edu/biology*** ga butun son 2 tayinlangan, chunki u ketma-ketlikda ikkinchi o'rinda turadi. Keyinchalik ko'pchilik veb-saytlar o'xshashlik va joylashuvni olish uchun tuzilganidan kelib chiqadigan xususiyatdan foydalaniladi. Aksariyat veb-saytlarda saytning har bir sahifasidan saytdagi qat'iy sahifalar to'plamiga (masalan, mualliflik huquqi to'g'risidagi bildirishnoma, foydalanish shartlari va boshqalar) havolalar to'plamidan iborat shablon mavjud. Bunday holda, veb-saytdagi sahifalarga mos keladigan qatorlar ko'plab umumiy jadval yozuvlariga ega bo'ladi. Bundan tashqari, URL-manzillarning leksikografik tartibida veb-sayt sahifalari jadvalda qo'shni qatorlar sifatida paydo bo'lishi ehtimoldan yiroq emas.

```
1: 1, 2, 4, 8, 16, 32, 64
2: 1, 4, 9, 16, 25, 36, 49, 64
3: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144
4: 1, 4, 8, 16, 25, 36, 49, 64
```

9.6-rasm. **Bog'lanishlar jadvalining to'rt qatorli segmenti**

Quyidagi strategiyani qo'llaymiz: jadval bo'ylab yuramiz, har bir jadval qatorini oldingi ettita qatorga ko'ra kodlaymiz. 9.6-rasmdagi misolda to'rtinchi qatorni "2-ofsetdagi qator bilan bir xil (jadvaldagi ikkita qator oldingi ma'nosi), 9 8 bilan almashtirilgan" deb kodlashimiz mumkin. Bu ofsetning spetsifikatsiyasini talab qiladi, tushirilgan butun sonlar (bu holda 9) va qo'shilgan (bu holda 8) butun sonlar. Oldingi yetti qatordan foydalanish ikkita afzalliklarga ega: (i) ofset faqat 3 bit bilan ifodalanishi mumkin. Bu tanlov empirik ravishda optimallashtirilgan (oldingi sakkiz emas, etti qatorning sababi 9.4-mashq mavzusi) va (ii) maksimal ofsetni ettita kabi kichik qiymatga belgilash ko'plab nomzod prototiplari orasida qimmat qidiruvni amalga oshirishdan qochadi. Joriy qatorni ifodalash uchun. Agar oldingi yetti qatorning hech biri joriy qatorni ifodalash uchun yaxshi prototip bo'lmasa-chi? Bu, masalan, jadval qatorlari bo'ylab yurganimizda, turli veb-saytlar orasidagi har bir chegarada sodir bo'ladi. Bunday holda, oddiygina qatorni bo'sh to'plamdan boshlab va ushbu qatordagi har bir butun sonni "qo'shish" sifatida ifodalaymiz. Har bir satrdagi bo'shliqlarni (haqiqiy butun sonlar o'rniga) saqlash uchun bo'shliqni kodlashdan foydalanish va bu bo'shliqlarni ularning qiymatlarini taqsimlash asosida qattiq kodlash orqali bo'sh joyni yanada qisqartiramiz. 9.5-bo'limda eslatib o'tilgan

tajribalarda, bu yerda ko'rsatilgan texnikalar seriyasi har bir havola uchun o'rtacha 3 bitni ishlatadi, bu sodda ko'rinishda talab qilinadigan 64 tadan keskin pasayish deganidir.

Ushbu g'oyalar ularga xotiraga qulay tarzda mos keladigan katta hajmdagi veb-grafiklarni taqdim etsa-da, hali ham ulanish so'rovlarini qo'llab-quvvatlashimiz kerak. Ushbu tasvirdan sahifadagi havolalar to'plamini olish nimaga olib keladi? Birinchidan, URL-manzildan jadvaldagi satr raqamiga (xesh) indeksni qidirishimiz kerak. Keyinchalik, boshqa qatorlardagi yozuvlar bo'yicha kodlangan bo'lishi mumkin bo'lgan ushbu yozuvlarni qayta qurishimiz kerak. Bu boshqa qatorlarni qayta tiklash uchun ofsetlarni kuzatishni talab qiladi - bu jarayon printsipial jihatdan ko'plab bilvosita darajalariga olib kelishi mumkin. Amalda, bu juda tez-tez sodir bo'lmaydi. Buni boshqarish uchun evristik usulni jadval qurilishiga kiritish mumkin: oldingi yetti qatorni joriy qatorni modellashtirish uchun nomzod sifatida ko'rib chiqayotganda, joriy qator va nomzod prototipi o'rtasidagi o'xshashlik chegarasini talab qiladi. Ushbu chegara ehtiyotkorlik bilan tanlanishi kerak. Agar chegara juda baland bo'lsa, kamdan-kam hollarda prototiplardan foydalaniladi va ko'plab qatorlarni yangidan ifodalaymiz. Agar chegara juda past bo'lsa, ko'pchilik qatorlar prototiplar nuqtai nazaridan ifodalanadi, shuning uchun so'rov vaqtida qatorni qayta tiklash oldingi prototiplar orqali ko'plab bilvosita darajalariga olib keladi.

9.4-mashq. Satrni oldingi yetti qatordan birida ifodalash ularga oldingi qatorlardan qaysi biri prototip sifatida foydalanayotganimizni aniqlash uchun uchta bitdan ko'p bo'lmagan foydalanish imkonini berganligini ta'kidladik. Nega oldingi qatorlar sakkiz emas, yetti? (Maslahat: oldingi etti qatorning hech biri yaxshi prototip bo'lmasa, vaziyatni ko'rib chiqing.)

9.5-mashq. 9.4-bo'limdagi sxema uchun URL manzilidagi havolalarning dekodlanishi bilvosita ko'p darajalarga olib kelishi mumkinligini ta'kidladik. Bilvosita darajalar soni URL manzillar soniga qarab chiziqli ravishda o'sadigan misol tuzing.

9- bob bo'yicha foydalanilgan adabiyotlar

Ruthven, Ian, and Mounia Lalmas.
2003.
A survey on the use of relevance feedback for information access

systems.
Knowledge Engineering Review 18 (1).
Sahoo, Nachiketa, Jamie Callan, Ramayya Krishnan, George Duncan, and Rema Padman.
2006.
Incremental hierarchical clustering of text documents.
In *Proc. CIKM*, pp. 357-366.
DOI: *doi.acm.org/10.1145/1183614.1183667*.
Sakai, Tetsuya.
2007.
On the reliability of information retrieval metrics based on graded relevance.
IP&M 43 (2): 531-548.
Salton, Gerard.
1971a.
Cluster search strategies and the optimization of retrieval effectiveness.
In *The SMART Retrieval System - Experiments in Automatic Document Processing* Salton (1971b), pp. 223-242.
Salton, Gerarded.).
1971b.
The SMART Retrieval System - Experiments in Automatic Document Processing.
Prentice Hall.
Salton, Gerard.
1975.
Dynamic information and library processing.
Prentice Hall.
Salton, Gerard.
1989.
Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.
Addison Wesley.
Xu, Jinxi, and W. Bruce Croft.
1996.
Query expansion using local and global document analysis.
In *Proc. SIGIR*, pp. 4-11. ACM Press.
Xu, Jinxi, and W. Bruce Croft.

1999.
Cluster-based language models for distributed retrieval.
In *Proc. SIGIR*, pp. 254-261. ACM Press.
DOI: *doi.acm.org/10.1145/312624.312687*.
Yang, Hui, and Jamie Callan.
2006.
Near-duplicate detection by instance-level constrained clustering.
In *Proc. SIGIR*, pp. 421-428. ACM Press.
DOI: *doi.acm.org/10.1145/1148170.1148243*.
Yang, Yiming.
1994.
Expert network: Effective and efficient learning from human decisions in text categorization and retrieval.
In *Proc. SIGIR*, pp. 13-22. ACM Press.
Yang, Yiming.
1999.
An evaluation of statistical approaches to text categorization.
IR 1: 69-90.
Yang, Yiming.
2001.
A study of thresholding strategies for text categorization.
In *Proc. SIGIR*, pp. 137-145. ACM Press.
DOI: *doi.acm.org/10.1145/383952.383975*.

9- bob bo‘yicha nazariy va amaliy test savollari

1. 1,50,000 talaba yozuvlaridan iborat ma’lumotlar fayli blok hajmi 4096 bayt bo‘lgan qattiq diskda saqlanadi. Ma’lumotlar fayli asosiy kalit RollNo bo‘yicha tartiblangan. Ushbu disk uchun yozuv ko‘rsatgichining hajmi 7 baytni tashkil qiladi. Har bir talaba yozuvida 12 bayt o‘lchamdagi ANum deb nomlangan nomzod kalit atributi mavjud. Aytaylik, ikkita maydondan, ANum qiymatidan va mos yozuvlar ko‘rsatgichidan iborat bo‘lgan indeks fayli bir xil diskda qurilgan va saqlanadi. Ma’lumotlar fayli va indeks faylining yozuvlari disk bloklariga bo‘linmagan deb faraz qiling. Indeks faylidagi bloklar soni _____ .

A) 698
B) 898
C) 899

D) 4096

2. Ba'zi asosiy qiymatlar uchun yozuvga ega bo'lgan indeks qanday tasniflanadi?
   A) Zich bo'lmagan indeks
   B) Zich indeks
   C) Chiziqli indeks
   D) Klaster indeks

3. Birlamchi indekslar, ikkilamchi indekslar va klaster indekslar bu barchasi _____.
   A) Tartiblangan indekslar
   B) Tartiblanmagan indeksalar
   C) Chiziqli indekslar
   D) Nisbiy qidiruv indekslar

4. Ko'p darajali indekslarda birinchi daraja uchun yaratilgan asosiy indeks quyidagicha tavsiflanadi.
   A) Ko'p darajali indeksning ikkinchi darajasi
   B) Ko'p darajali indeksning nol darajasi
   C) Ko'p darajali indeksning birinchi darajasi
   D) Ko'p darajali indeksning uchinchi darajasi

5. Jismonoiy ko'rsatkich bilan diskdagi yozuvlar manzilini ko'rsatadigan indekslar quyidagicha tasniflanadi.
   A) Jismoniy indeks
   B) Tarkibiy indeks
   C) Xeshlash indeksi
   D) Mantiqiy indeks

6. Zich bo'lmagan indeksga misol keltiring.
   A) Klasterlash indeksi
   B) Uchlik indeks
   C) Ikkilamchi indeks
   D) Asosiy indeks

7. Ma'lumotlar faylida qanday blokning birinchi yozuvi chaqiriladi?
   A) Langar rekordi
   B) Zich rekord
   C) Zich bo'lmagan rekord
   D) To'g'ri javob yo'q

8. Qanday fayl har bir maydon uchun ikkilamchi indeksga ega bo'lgan fayli sifatida tasniflanadi?
   A) To'liq teskari fayl

   B) To'liq indekslangan fayl
   C) Ikkilamchi indekslangan fayl
   D) Asosiy indekslangan fayl

9. Qanday kalit buyurtma maydonidagi bir xil ma'lumotlar turiga ega bo'lgan asosiy indeksdagi birinchi maydon deb hisoblanadi?
   A) Asosiy kalit
   B) Indekslangan kalit
   C) Uchlik kalit
   D) Ikkinchi darajali kalit

10. Ko'p darajali indekslarda uning ikkinchi darajasi uchun yaratilgan asosiy indeks quyidagicha tavsiflanadi.
   A) Ko'p darajali indeksning uchinchi darajasi
   B) Ko'p darajali indeksning ikkinchi darajasi
   C) Ko'p darajali indeksning birinchi darajasi
   D) Ko'p darajali indeksning nol darajasi

Giperhavolalar va ***Internetning grafik tuzilishi***ni tahlil qilish veb-qidiruvni rivojlantirishda muhim rol o'ynadi. Ushbu bobda veb-qidiruv natijalarini tartiblash uchun giperhavolalardan foydalanishga e'tibor qaratamiz. Bunday havola tahlili veb-qidiruv tizimlari tomonidan har qanday so'rov bo'yicha veb-sahifa uchun kompozit ballni hisoblashda ko'rib chiqiladigan ko'plab omillardan biridir. 10.1-bo'limda Internetning Ba'zi asoslarini grafik sifatida ko'rib chiqishni boshlaymiz, so'ngra reyting uchun havolalarni tahlil qilish elementlarini texnik ishlab chiqishga o'tamiz.

Veb-qidiruv uchun havolalar tahlili iqtibos tahlili sohasida intellektual ajdodlarga ega bo'lib, uning jihatlari ***bibliometriya*** deb nomlanuvchi soha bilan bir-biriga mos keladi. Ushbu fanlar ilmiy maqolalarning ta'sirini orasidagi iqtiboslarni tahlil qilish orqali aniqlashga intiladi. Iqtiboslar ilmiy maqoladan boshqalarga vakolat berilishini ifodalaganidek, Internetdagi havolalarni tahlil qilish veb-sahifadan boshqasiga giperhavolalarni vakolat berish sifatida ko'rib chiqadi. Shubhasiz, har bir iqtibos yoki giperhavola bunday vakolatni berishni nazarda tutmaydi.

Shu sababli, veb-sahifa sifatini havolalar soni bo'yicha o'lchash (boshqa sahifalardan iqtiboslar) yetarli darajada mustahkam emas. Masalan, havolalar sonini sun'iy ravishda oshirish maqsadida maqsadli veb-sahifaga ishora qiluvchi bir nechta veb-sahifalarni o'rnatish mumkin. Ushbu hodisa ***havola spam*** deb ataladi.

Shunga qaramay, iqtibos hodisasi keng tarqalgan va yetarlicha ishonchli bo'lib, veb-qidiruv mexanizmlari yanada murakkab havola tahlilidan reyting uchun foydali signallarni olishlari mumkin. Havolani tahlil qilish, shuningdek, vebni skanerlashda keyingi qaysi sahifa(lar)ni skanerlash kerakligini ko'rsatadigan foydali ko'rsatkichdir.

Bu 9-bobning oldingi navbatlarida ustuvorlikni belgilash uchun havola tahlilidan foydalanish orqali amalga oshiriladi. 10.1-bo'lim havolalarni tahlil qilishda veb-grafikadan foydalanishning asosiy g'oyalarini ishlab chiqadi. Keyin 10.2 va 10.3 bo'limlari havolalarni tahlil qilish uchun ikkita alohida usulni ishlab chiqadi, ***PageRank*** va ***HITS***.

8.2.1-bo'lim va ayniqsa 8.2-rasmdagi veb-grafik tushunchasini eslang. Aloqa tahlilini o'rganishimiz ikkita tushunchaga asoslanadi:

1. **B** sahifaga ishora qiluvchi langar matni **B** sahifaning yaxshi tavsifi.
2. **A** dan **B** gacha bo'lgan giperhavola **A** sahifani yaratuvchisi tomonidan **B** sahifani tasdiqlaydi. Bu har doim ham shunday emas. Masalan, bitta veb-saytdagi sahifalar orasidagi ko'plab havolalar umumiy shablon foydalanuvchisidan kelib chiqadi. Misol uchun, ko'pgina korporativ veb-saytlarda mualliflik huquqi to'g'risidagi bildirishnomani o'z ichiga olgan har bir sahifada ko'rsatgich mavjud - bu aniq. Shunga ko'ra, havolalarni tahlil qilish algoritmlarini amalga oshirish odatda bunday "ichki" havolalarni kamaytiradi.

10.1.1. Anchor matni va veb-grafik

Veb-sahifadagi HTML kodining quyidagi qismi ACM jurnalining bosh sahifasiga ishora qiluvchi giperhavolani ko'rsatadi:

```
<a href="http://www.acm.org/jacm/">Journal of the ACM.</a>
```

Bunday holda, havola ***http://www.acm.org/jacm/*** sahifasiga ishora qiladi va langar matni ACM jurnalidir. Shubhasiz, bu misolda langar maqsadli sahifani tavsiflaydi. Ammo keyin maqsadli sahifaning (*B* = ***http://www.acm.org/jacm/***) o'zi bir xil tavsifni va jurnalda qo'shimcha ma'lumotlarni o'z ichiga oladi. Xo'sh, langar matni nima uchun ishlatiladi? Vebning **B** sahifasi o'zini to'g'ri tavsiflamaydigan holatlarga to'la. Ko'p hollarda bu **B** sahifasi noshirlari o'zlarini qanday ko'rsatishni tanlashlari masalasidir; Bu ayniqsa, korporativ veb-sahifalarda keng tarqalgan bo'lib, bu yerda veb-sayt mavjudligi marketing bayonotidir. Misol uchun, ushbu kitobni yozish vaqtida IBM korporatsiyasining bosh sahifasida (***http://www.ibm.com***) dunyodagi eng yirik kompyuter ishlab chiqaruvchisi sifatida IBM keng tarqalgan bo'lsa-da, uning HTML kodida kompyuter atamasi mavjud emas edi. Xuddi shunday, ***Yahoo!*** bosh sahifasining HTML kodi! (***http://www.yahoo.com***) hozirda portal so'zini o'z ichiga olmaydi.

Shunday qilib, veb-sahifadagi atamalar va veb-foydalanuvchilar ushbu veb-sahifani qanday tasvirlashlari o'rtasida ko'pincha bo'shliq mavjud. Shunday qilib, veb-qidiruvchilar uni so'rash uchun sahifadagi

atamalardan foydalanishlari shart emas. Bundan tashqari, ko'pgina veb-sahifalar grafik va tasvirlarga boy va o'z matnlarini ushbu tasvirlarga joylashtiradi. Bunday hollarda, skanerlash paytida bajariladigan HTML tahlili ushbu sahifalarni indekslash uchun foydali bo'lgan matnni chiqarmaydi. Bunga "standart AQ" yondashuvi 1- qism, 9-bob va 1.4-bo'limda ko'rsatilgan usullardan foydalanish bo'ladi. ***Anchor matn*** ortidagi tushuncha shundan iboratki, bunday usullarni langar matni bilan almashtirish mumkin va shu bilan veb-sahifa mualliflari hamjamiyatining asabiga tegadi.

http://www.ibm.com ga ishora qiluvchi ko'plab giperhavolalarning langarlari kompyuter so'zini o'z ichiga olganligi veb-qidiruv tizimlari tomonidan ishlatilishi mumkin. Masalan, langar matn atamalari maqsadli veb-sahifani indekslash uchun shartlar sifatida kiritilishi mumkin. Shunday qilib, kompyuter atamasi uchun e'lonlar ***http://www.ibm.com*** hujjatini va portal atamasi uchun ***http://www.yahoo.com*** hujjatini o'z ichiga oladi, bu atamalar ushbu shartlarni ko'rsatish uchun maxsus ko'rsatkichdan foydalanadi. Langar (sahifa ichidagi emas) matn sifatida yuzaga keladi. Sahifa ichidagi atamalarda bo'lgani kabi, langar matn atamalari odatda chastotaga qarab o'lchanadi, bunda juda tez-tez uchraydigan shartlar uchun jazo qo'llaniladi (Internetdagi langar matnidagi eng keng tarqalgan atamalar **idf** ga juda o'xshash usullardan foydalanilgan). Terminlarning haqiqiy vazni 4.4.1-bo'limda bo'lgani kabi, mashinada o'rganilgan ball bilan aniqlanadi. Hozirgi veb-qidiruv mexanizmlari langar matn atamalariga katta vazn qo'ygandek ko'rinadi. Anchor matnidan foydalanish ba'zi qiziqarli yon ta'sirlarga ega. Ko'pgina veb-qidiruv tizimlarida katta ko'k rangni qidirish IBM korporatsiyasining bosh sahifasini eng yaxshi natija sifatida qaytaradi. Bu ko'pchilik IBMga murojaat qilish uchun foydalanadigan mashhur taxallusga mos keladi. Boshqa tomondan, yovuz imperiya kabi haqoratli langar matni veb-qidiruv tizimlarida ushbu atamalarni so'rashda biroz kutilmagan natijalarga olib keladigan holatlar ko'p bo'lgan (va shunday bo'ladi). Ushbu hodisadan ma'lum saytlarga qarshi tashkillashtirilgan kampaniyalarda foydalanilgan. Bunday tashkil etilgan langar matni spam yuborishning bir shakli bo'lishi mumkin chunki veb-sayt tanlangan so'rov shartlari bo'yicha reytingini oshirish uchun o'ziga ishora qiluvchi chalg'ituvchi langar matnini yaratishi mumkin. Anchor matnining bunday muntazam ravishda suiiste'mol qilinishini aniqlash va unga qarshi

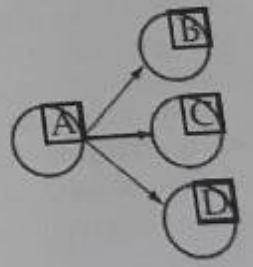
kurashish veb-qidiruv tizimlari amalga oshiradigan spamni aniqlashning yana bir shaklidir.

Anchor matni atrofidagi matn oynasi (ba'zan kengaytirilgan langar matni deb ataladi) ko'pincha langar matnining o'zi kabi foydalanish mumkin; Masalan, veb-matnning bir bo'lagini ko'rib chiqing, u yerda vedik oyatlari haqida yaxshi muhokama qilingan <a>bu yerda. Bu bir qator sozlamalarda ko'rib chiqildi va ushbu oynaning foydali kengligi o'rganildi. Malumot uchun 10.4-bo'limga qarang.

10.1-mashq. Har qanday tugundan (veb-sahifadan) boshqasiga har doim veb-grafikda yo'naltirilgan giperhavolalarni kuzatib borish mumkinmi? Nima uchun yoki nima sababdan?

10.2-mashq. Internetda chalg'ituvchi langar-matn misolini toping.

10.3-mashq. X veb-sahifasi uchun langar-matnli iboralar to'plamini hisobga olgan holda, ushbu to'plamdan **x** ni eng tavsiflovchi atama yoki iborani tanlash uchun evristikadan foydalaning.



10.1-rasm. **A tugunidagi tasodifiy syorfer B, C va D ning har biriga 1/3 ehtimollik bilan boradi**

10.4-mashq. Oldingi mashqdagi evristikangiz **D** dagi bir nechta sahifalardan **x** uchun langar matnini takrorlaydigan bitta **D** domenini hisobga oladimi?

10.2. PageRank

Endi faqat havola tuzilishidan olingan reyting va reyting o'lchovlariga e'tibor qaratamiz. Havolani tahlil qilish uchun ularning birinchi texnikamiz veb-grafikdagi har bir tugunga ***PageRank*** deb nomlanuvchi 0 dan 1 gacha bo'lgan raqamli ballni belgilaydi. Tugunning

PageRank darajasi veb-grafaning havola tuzilishiga bog'liq bo'ladi. So'rovni hisobga olgan holda, veb-qidiruv tizimi har bir veb-sahifa uchun kosinus o'xshashligi (1-qism, 6.3-bo'lim) va atama yaqinligi kabi yuzlab xususiyatlarni birlashtirgan umumiy ballni hisoblab chiqadi. 4.4.1-bo'lim usullaridan foydalangan holda ishlab chiqilgan ushbu kompozit ball so'rov natijalarining tartiblangan ro'yxatini taqdim etish uchun ishlatiladi.

Veb-sahifadan (veb-grafikning tugunidan) boshlanadigan va Internetda tasodifiy yurishni quyidagi tarzda amalga oshiradigan tasodifiy syorfingistni ko'rib chiqing. Har bir bosqichda syorfingist o'zining joriy **A** sahifasidan **A** giperhavola qiladigan tasodifiy tanlangan veb-sahifaga o'tadi. 10.1-rasmda **B**, **C** va **D** tugunlariga uchta giperhavola mavjud bo'lgan **A** tugunidagi syorfingist ko'rsatilgan. Syorfingist keyingi bosqichda ushbu uchta tugundan biriga o'tadi, ehtimollik 1/3 ga teng. Surfer tugundan tugunga bu tasodifiy yurishda davom etar ekan, u Ba'zi tugunlarga boshqalarga qaraganda tez-tez tashrif buyuradi. Intuitiv ravishda, bu boshqa tez-tez tashrif buyuriladigan tugunlardan keladigan ko'plab havolalarga ega tugunlardir. Orqa fikr shundan iboratki, bu yurishda tez-tez tashrif buyurilgan sahifalar muhimroqdir. Agar syorfingistning joriy joylashuvi, **A** tugunining tashqi havolalari bo'lmasa-chi? Buni hal qilish uchun tasodifiy surferimiz uchun qo'shimcha operatsiyani kiritamiz: ***teleport operatsiyasi***. Teleport operatsiyasida syorfingist veb-grafikdagi tugundan boshqa tugunga o'tadi. Bu uni brauzerining URL satriga manzil kiritganligi sababli sodir bo'lishi mumkin.

*Teleport operatsiyasi*ning maqsadi barcha veb-sahifalardan tasodifiy tanlanishi uchun modellashtirishdir. Boshqacha qilib aytganda, agar **N** veb-grafikdagi tugunlarning umumiy soni 1 bo'lsa, teleport operatsiyasi surferni 1/N ehtimollik bilan har bir tugunga olib boradi. Surfchi o'zining hozirgi holatiga 1/N ehtimollik bilan teleportatsiya qiladi. Veb-grafaning har bir tuguniga ***PageRank*** ballini belgilashda teleport operatsiyasidan ikki usulda foydalaniladi: (1) chiqish havolalari bo'lmagan tugunda bo'lsa, surfer teleport operatsiyasini chaqiradi. (2) Chiquvchi havolalarga ega bo'lgan har qanday tugunda surfchi $0 < a < 1$ ehtimollik bilan teleport operatsiyasini va standart tasodifiy yurishni (10.1-rasmda bo'lgani kabi tasodifiy ravishda tanlab olingan havolaga rioya qiling) 1 - a ehtimollik bilan chaqiradi, bu yerda **a** - oldindan tanlangan o'zgarmas parametr. Odatda, **a** 0,1 bo'lishi mumkin. 10.2.1-bo'limda ***Markov zanjirlari nazariyasi***dan foydalanib, surfer ushbu qo'shma jarayonni (tasodifiy

yurish va teleport) bajarganida, u veb-grafaning har bir **v** tuguniga $\pi(v)$ vaqtning belgilangan qismida tashrif buyuradi. (1) veb-grafaning tuzilishiga va (2) **a** qiymatiga bog'liq. Bu qiymatni $\pi(v)$ ning **PageRank** deb ataymiz va bu qiymatni qanday hisoblashni 10.2.2-bo'limda ko'rsatamiz.

10.2.1. Markov zanjirlari

Markov zanjiri - bu diskret vaqtli stoxastik jarayon: har birida tasodifiy tanlov amalga oshiriladigan bir qator vaqt bosqichlarida sodir bo'ladigan jarayondir. Markov zanjiri N ta holatdan iborat. Har bir veb-sahifa ***Markov zanjiridagi*** holatga mos keladi.

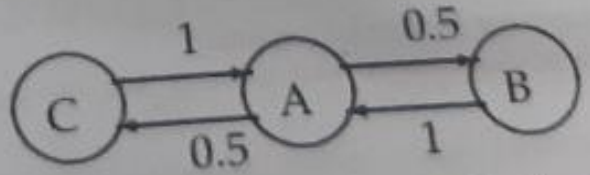
Markov zanjirining har bir yozuvi [0, 1] oraliqda joylashgan N×N o'tish ehtimoli **P** matritsasi bilan tavsiflanadi. **P** ning har bir qatoridagi yozuvlar 1 ga teng. Markov zanjiri har qanday vaqt oralig'ida **N** holatlardan birida bo'lishi mumkin; u holda, P_{ij} yozuvi hozirgi holat **i** bo'lishi sharti bilan keyingi vaqt bosqichidagi holat **j** bo'lish ehtimolini bildiradi. Har bir kirish P_{ij} o'tish ehtimoli sifatida tanilgan va faqat joriy holatga bog'liq. Bu Markov mulki sifatida tanilgan. Shunday qilib, Markov mulkiga ko'ra,

$$\forall i, j, P_{i,j} \in [0,1]$$

$$\forall i, \sum_{j=1}^{N} P_{i,j} = 1$$

(10.1) tenglamani qanoatlantiradigan manfiy bo'lmagan yozuvlari bo'lgan matritsa ***stoxastik matritsa*** deb ataladi. Stoxastik matritsaning asosiy xususiyati shundaki, uning eng katta xos qiymatiga mos keladigan asosiy chap xos vektori 1 ga teng.

1. Bu kollektsiyadagi hujjatlar soni uchun **N** dan foydalanishimizga mos keladi:



10.2-rasm. **Uchta holatga ega oddiy Markov zanjiri**

Havolalardagi raqamlar o'tish ehtimolini ko'rsatadi. Markov zanjirida Markov zanjiri uchun keyingi holatlarning ehtimollik taqsimoti faqat hozirgi holatga bog'liq, Markov zanjiri hozirgi holatga qanday kelganiga emas. 10.2-rasmda uchta holatga ega oddiy Markov zanjiri ko'rsatilgan. O'rta **A** holatidan 0,5 ga **B** yoki **C** ga (teng) ehtimolliklar bilan davom etamiz. **B** yoki **C** dan 1 ehtimoldan **A** ga o'tamiz. Bu Markov zanjirining o'tish ehtimoli matritsasi quyidagicha:

$$\begin{pmatrix} 0 & 0.5 & 0.5 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Markov zanjirini uning holatlari bo'yicha ehtimollik taqsimotini ehtimollik vektori sifatida ko'rish mumkin: barcha yozuvlari [0, 1] oraliqda bo'lgan vektor, va yozuvlar yig'indisi 1 ga teng. Har bir komponenti N o'lchovli ehtimollik vektori. Markov zanjirining N holatlaridan biriga to'g'ri keladi, uning holatlari bo'yicha ehtimollik taqsimoti sifatida qaralishi mumkin. 10.2-rasmdagi oddiy Markov zanjirimiz uchun ehtimollik vektori yig'indisi 1 ga teng 3 ta komponentga ega bo'ladi.

Veb-grafikda tasodifiy surferni Markov zanjiri sifatida ko'rishimiz mumkin, har bir veb-sahifa uchun bitta holat va har bir o'tish ehtimoli bir veb-sahifadan ikkinchisiga o'tish ehtimolini ifodalaydi. Teleport operatsiyasi bu o'tish ehtimoliga hissa qo'shadi. Veb-grafaning **A** qo'shnilik matritsasi quyidagicha aniqlanadi: agar **i** sahifadan **j** sahifaga giperhavola mavjud bo'lsa, u holda, $A_{ij} = 1$ aks holda $A_{ij} = 0$. Markov zanjirimiz uchun **P** o'tish ehtimoli matritsasini osongina olishimiz mumkin. $N \times N$ matritsasi **A** dan:

1. Agar A qatorida 1 bo'lmasa, har bir elementni $1/N$ ga almashtiring. Boshqa barcha qatorlar uchun quyidagicha davom eting.
2. A dagi har bir 1 ni qatoridagi 1 lar soniga bo'ling. Shunday qilib, agar uchta 1 bo'lgan qator bo'lsa, ularning har biri 1/3 ga almashtiriladi.
3. Olingan matritsani 1 - a ga ko'paytiring.
4. P ni olish uchun olingan matritsaning har bir yozuviga a/N qo'shing.

Istalgan vaqtda surfer pozitsiyasining ehtimollik taqsimotini $\vec{x}$ *is* 1 ehtimollik vektori bilan tasvirlashimiz mumkin. $\vec{x}$. *at* $t = 0$ dagi mos

yozuvi 1, qolganlari esa nolga teng bo'lgan holatda boshlashi mumkin. Ta'rifga ko'ra, surferning **t=1** da taqsimlanishi ehtimollik vektori $\vec{x}P$ bilan berilgan; $(\vec{x}P) = \vec{x}P^2$, $\vec{x}P$; *at* $t = 2$ va hokazo. Ushbu jarayonni 10.2.2-bo'limda batafsil bayon qilinadi. Shunday qilib, istalgan vaqtda surferning shtatlar bo'yicha taqsimlanishini faqat dastlabki taqsimot va **P** o'tish ehtimoli matritsasi hisobga olingan holda hisoblashimiz mumkin. Agar ***Markov zanjiri*** ko'p vaqt bosqichlarida ishlashga ruxsat bersa, har bir holatga (turli) chastotada tashrif buyuriladi. Markov zanjirining tuzilishiga bog'liq. Ularning ishlayotgan analogiyamizda, surfer boshqa sahifalarga qaraganda ma'lum veb-sahifalarga (masalan, mashhur yangiliklar bosh sahifalariga) tez-tez tashrif buyuradi. Endi ushbu tushunchani aniq qilib, bunday tashrif chastotasi o'zgarmasdir, barqaror holatdagi miqdorga yaqinlashadigan shartlarni belgilanadi. Shundan so'ng, har bir **v** tugunining ***PageRank*** qiymatini ushbu doimiy tashrif chastotasiga o'tamiz va uni qanday hisoblash mumkinligini ko'rsatamiz.

Ta'rif: Markov zanjiri ergodik deb ataladi, agar T 0 musbat butun son mavjud bo'lsa, Markov zanjiridagi barcha i, j holatlar juftligi uchun, agar u i holatda 0 vaqtda boshlangan bo'lsa, u holda barcha $t > T$ 0 uchun ,t vaqtida j holatda bo'lish ehtimoli 0 dan katta.

Markov zanjiri ergodik bo'lishi uchun uning holatlari va nolga teng bo'lmagan o'tish ehtimoli uchun ikkita texnik shart talab qilinadi: bu shartlar ***qaytarilmaslik*** va ***aperiodiklik*** deb ataladi. Norasmiy ravishda birinchisi har qanday holatdan boshqasiga nolga teng bo'lmagan ehtimollikdagi o'tishlar ketma-ketligi mavjudligini ta'minlaydi, ikkinchisi esa holatlar to'plamlarga bo'linmasligini ta'minlaydi, shunda barcha holat o'tishlari bir to'plamdan ikkinchisiga siklik ravishda sodir bo'ladi.

10.1- teorema. Har qanday ergodik Markov zanjiri uchun yagona barqaror holat ehtimollik vektori $\vec{\pi}$ mavjud bo'lib, u **P** ning asosiy chap xos vektori bo'lib, agar $\vec{\pi}$**(i, t)** **i** holatiga **t** bosqichda tashriflar soni bo'lsa, u holda

$$\lim_{t \to \infty} \frac{\eta(i,t)}{t} = \pi(i)$$

Bu yerda $\pi(i) > 0$ holat uchun barqaror holat ehtimolidir.

10.1- teoremadan kelib chiqadiki, teleportatsiya bilan tasodifiy yurish induktsiyalangan Markov zanjiri holatlari bo'yicha barqaror holat ehtimolining yagona taqsimlanishiga olib keladi. Davlat uchun bu barqaror holat ehtimoli mos keladigan veb-sahifaning ***PageRank*** darajasidir.

10.2.2. PageRankni hisoblash

PageRank qiymatlarini qanday hisoblaymiz? 7.2- tenglamadan chap xos vektor ta'rifini eslang. **P** o'tish ehtimoli matritsasining chap xos vektorlari N-vektorlar $\vec{\pi}$ bo'lib, quyidagicha hisoblanadi:

$$\vec{\pi}P = \lambda\vec{\pi}$$

Asosiy xos vektor $\vec{\pi}$ dagi N ta yozuvlar teleportatsiya bilan tasodifiy yurishning barqaror holat ehtimoli va shuning uchun tegishli veb-sahifalar uchun ***PageRank*** qiymatlaridir. (10.2) tenglamani quyidagicha talqin qilishimiz mumkin: agar $\vec{\pi}$ surferning veb-sahifalar bo'ylab ehtimollik taqsimoti bo'lsa, u **~p** barqaror holat taqsimotida qoladi. $\vec{\pi}$ barqaror holat taqsimoti ekanligini hisobga olsak, ularda $\pi P = 1\pi$ bor, shuning uchun P ning xos qiymatidir. Shunday qilib, agar P matritsasining asosiy chap xos vektorini - xos qiymati 1 bo'lganini hisoblasak, shunday bo'lar edi. ***PageRank*** qiymatlarini hisoblab chiqish kerak.

Chap xos vektorlarni hisoblash uchun ko'plab algoritmlar mavjud; 7-bobning oxiridagi havolalar va ushbu bob bular uchun qo'llanmadir. Bu yerda ba'zan quvvat iteratsiyasi deb nomlanuvchi oddiy usulni beramiz. Agar $\vec{x}$ holatlar bo'yicha dastlabki taqsimot bo'lsa, u holda **t** vaqtdagi taqsimot $\vec{x}P^t$ bo'ladi. **T** kattalashganda, $\vec{x}P^{t2}$ taqsimoti $\vec{x}P^{t+1}$ taqsimotiga juda o'xshash bo'lishini kutamiz chunki katta **t** uchun ***Markov zanjiri*** barqaror holatga kelishini kutamiz. 10.1- teoremaga ko'ra, bu $\vec{x}$ boshlang'ich taqsimotiga bog'liq emas.

Quvvatli iteratsiya usuli surferning yurishini taqlid qiladi: bir holatda boshlang va har bir shtat uchun tashrif chastotalarini kuzatib, **t** uchun ko'p sonli qadamlarni yuring. Ko'p sonli qadamlardan so'ng, bu chastotalar hisoblangan chastotalarning o'zgarishi oldindan belgilangan chegaradan past bo'lishi uchun "joylashadi". Ushbu jadvalli chastotalarni ***PageRank qiymatlari*** deb e'lon qilinadi.

10.6-mashqdagi veb-grafani a=0,5 bilan ko'rib chiqiladi. Serferning teleportatsiya bilan yurishining o'tish ehtimoli matritsasi quyidagicha hisoblanadi:

$$P = \begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 5/12 & 1/6 & 5/12 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}$$

Tasavvur qiling-a, surfer 1-holatda boshlanadi, bu $\vec{x}_0 = (1\ 0\ 0)$. ehtimollikning dastlabki taqsimot vektoriga mos keladi. Keyin, bir qadamdan so'ng, tarqatish hisoblanadi:

$$\vec{x}_0 P = (1/6 \quad 2/3 \quad 1/6) = \vec{x}_1.$$

2. E'tibor bering, P^t P^T bilan belgilangan **P** ning transpozitsiyasini emas, balki **t** darajaga ko'tarilgan **P** ni ifodalaydi.

$\vec{x}_0$	1	0	0
$\vec{x}_1$	1/6	2/3	1/6
$\vec{x}_2$	1/3	1/3	1/3
$\vec{x}_3$	1/4	1/2	1/4
$\vec{x}_4$	7/24	5/12	7/24
...	...	...	...
$\vec{x}$	5/18	4/9	5/18

10.3-rasm. **Ehtimollik vektorlari ketma-ketligi**

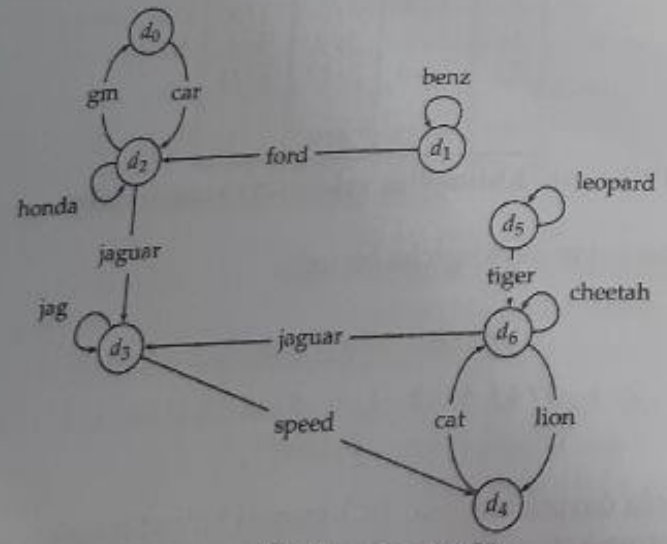
Ikki qadamdan keyin quyidagicha bo'ladi:

$$\vec{x}_1 P = (1/6 \quad 2/3 \quad 1/6)\begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 5/12 & 1/6 & 5/12 \\ 1/6 & 2/3 & 1/6 \end{pmatrix} = (1/3 \quad 1/3 \quad 1/3) = \vec{x}_2.$$

Shu tarzda davom ettirilsa, 10.3-rasmda ko'rsatilganidek, ehtimollik vektorlari ketma-ketligi hosil bo'ladi.

Bir necha bosqichlarni davom ettirsak, taqsimot $\vec{x} = (5/18 \quad 4/9 \quad 5/18)$. barqaror holatiga yaqinlashayotganini ko'ramiz. Ushbu oddiy misolda ***Markov zanjirining simmetriyasini*** kuzatish orqali bu barqaror holatdagi ehtimollik taqsimotini to'g'ridan-to'g'ri hisoblashimiz mumkin: 1 va 3 holatlar simmetrikdir, bu tenglamadagi

o'tish ehtimoli matritsasining birinchi va uchinchi qatorlari ekanligidan ko'rinib turibdi. 10.3) bir xil. Demak, ularning ikkalasining ham bir xil barqaror holat ehtimoli borligini va bu ehtimolni **p** bilan belgilagan holda, barqaror holat taqsimoti $\bar{\pi} = (p \quad 1-2p \quad p)$. ko'rinishda ekanligini bilamiz. Endi $\bar{\pi} = \bar{\pi}P$ identifikatsiyasidan foydalanib, ***p=5/18*** va natijada $\bar{\pi} = (5/18 \quad 4/9 \quad 5/18)$. ni olish uchun oddiy chiziqli tenglamani yechamiz. Sahifalarning **PageRank** qiymatlari (ular orasidagi yashirin tartib) foydalanuvchi qo'yishi mumkin bo'lgan har qanday so'rovga bog'liq emas. Shunday qilib, **PageRank** har bir veb-sahifaning statik sifatining so'rovdan mustaqil o'lchovidir. Boshqa tomondan, sahifalarning nisbiy tartibi intuitiv ravishda xizmat ko'rsatilayotgan so'rovga bog'liq bo'lishi kerak. Shu sababli, qidiruv tizimlari so'rovda veb-sahifani baholashda ko'plab omillardan biri sifatida **PageRank** kabi statik sifat ko'rsatkichlaridan foydalanadi. Haqiqatan ham, **PageRankning** umumiy ballga nisbiy hissasi yana 4.4.1-bo'limdagi kabi mashinali o'qitishdan olingan ball bilan aniqlanilishi mumkin.



10.4-rasm. **Kichik veb-grafik**

Yoylar tegishli havolaning langar matnida uchraydigan so'z bilan izohlanadi.

10.1-misol. Uning (stoxastik) o'tish ehtimoli matritsasi: 10.4-rasmdagi grafikni ko'rib chiqing. 0,14 teleportatsiya tezligi uchun:

0.02	0.02	0.88	0.02	0.02	0.02	0.02
0.02	0.45	0.45	0.02	0.02	0.02	0.02
0.31	0.02	0.31	0.31	0.02	0.02	0.02
0.02	0.02	0.02	0.45	0.45	0.02	0.02
0.02	0.02	0.02	0.02	0.02	0.02	0.88
0.02	0.02	0.02	0.02	0.02	0.45	0.45
0.02	0.02	0.02	0.31	0.31	0.02	0.31

Ushbu matritsaning **PageRank vektori**:

$\bar{x} = (0.05 \quad 0.04 \quad 0.11 \quad 0.25 \quad 0.21 \quad 0.04 \quad 0.31)$.

E'tibor bering, 10.4-rasmda q_2, q_3, q_4 va q_6 kamida ikkita ichki bog'lanishga ega bo'lgan tugunlardir. Ulardan q_2 eng past **PageRankga** ega chunki tasodifiy yurish grafikning yuqori qismidan chiqib ketishga intiladi - yuruvchi u yerga faqat teleportatsiya orqali qaytishi mumkin.

10.2.3. PageRankni hisoblash teleport operatsiyasi

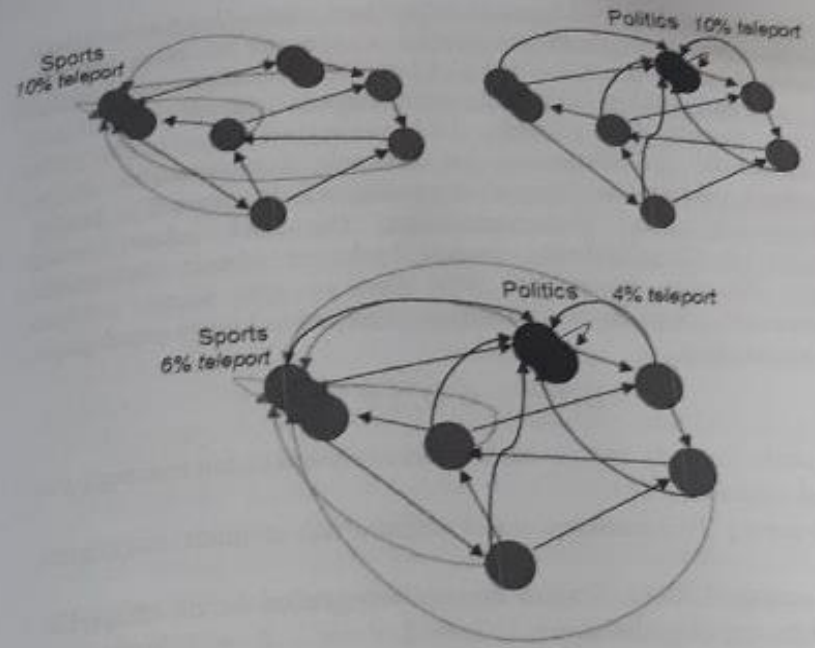
Hozirgacha ***PageRank hisobini*** teleport operatsiyasi bilan muhokama qildik, bunda surfer tasodifiy bir xil tarzda tanlangan tasodifiy veb-sahifaga o'tadi. Endi bir xilda tanlanmagan tasodifiy veb-sahifaga teleportatsiya qilish ko'rib chiqiladi. Shunday qilib, muayyan manfaatlarga moslashtirilgan ***PageRank qiymatlarini*** olishimiz mumkin. Misol uchun, sport ishqibozi sportga oid sahifalar sport bo'lmagan sahifalarga qaraganda yuqoriroq reytingga ega bo'lishini xohlashi mumkin. Aytaylik, sportga oid veb-sahifalar veb-grafikda bir-biriga "yaqin" joylashgan. Keyin, tasodifiy surfer tez-tez tasodifiy sport sahifalarida o'zini topadigan kishi (tasodifiy yurish paytida) ko'p vaqtini sport sahifalarida o'tkazishi mumkin, shuning uchun sport sahifalarining barqaror taqsimlanishi kuchaytiriladi. Aytaylik, tasodifiy surfer avvalgidek teleport operatsiyasiga ega bo'lib, bir xil tanlangan tasodifiy veb-sahifaga teleportatsiya qilish o'rniga sport mavzusidagi tasodifiy

veb-sahifaga teleportatsiya qiladi. Sport mavzusidagi barcha veb-sahifalarni qanday yig'ishimizga e'tibor qaratmaymiz. Aslida, faqat sport bilan bog'liq veb-sahifalarning nolga teng bo'lmagan S kichik to'plamiga muhtojmiz, shuning uchun teleport operatsiyasini amalga oshirish mumkin. Buni, masalan, ochiq katalog loyihasi (*http://www.dmoz.org/*) yoki *Yahoo* kabi sport sahifalarining qo'lda tuzilgan katalogidan olish mumkin.

Sport bilan bog'liq sahifalarning **S** to'plami bo'sh bo'lmagan taqdirda, $\gamma \supseteq S$ bo'sh bo'lmagan veb-sahifalar to'plami mavjud bo'lib, ular ustida tasodifiy yurish barqaror holat taqsimotiga ega. Ushbu sport **PageRank** taqsimotini $\vec{\pi}_s$ bilan belgilanadi. **Y** harfida bo'lmagan veb-sahifalar uchun **PageRank** qiymatlarini nolga o'rnatamiz. $\vec{\pi}_s$ ni sport mavzuga oid PageRank deb ataymiz.

Teleportatsiya tasodifiy surferni bir xil tanlangan sport sahifasiga olib borishini talab qilmaymiz. Teleportatsiya maqsadlari **S** bo'yicha taqsimlash aslida avtomatik bo'lishi mumkin. Xuddi shunday, fan, din, siyosat va boshqalar kabi bir nechta mavzularning har biri uchun mavzuga oid **PageRank** taqsimotini tasavvur qilishimiz mumkin. Ushbu tarqatishlarning har biri har bir veb-sahifaga [0, 1) oraliqda PageRank qiymatini belgilaydi. Ushbu mavzular orasidan faqat bitta mavzuga qiziqqan foydalanuvchi uchun qidiruv natijalarini baholash va tartiblashda tegishli **PageRank** taqsimotidan foydalanishimiz mumkin. Bu qidiruv tizimi foydalanuvchini qaysi mavzuga qiziqtirayotganini bilishi mumkin bo'lgan sozlamalarni ko'rib chiqish imkoniyatini beradi. Bu foydalanuvchilar o'z qiziqishlarini aniq qayd etishlari yoki tizim vaqt o'tishi bilan har bir foydalanuvchining xatti-harakatlarini kuzatish orqali o'rganadi.

Ammo foydalanuvchi bir nechta mavzulardan qiziqishlar aralashmasiga ega ekanligi ma'lum bo'lsa-chi? Misol uchun, foydalanuvchi 60% sport va 40% siyosat bo'lgan qiziqish aralashmasi bo'lishi mumkin; Ushbu foydalanuvchi uchun shaxsiylashtirilgan PageRankni hisoblay olamizmi? Bir qarashda, bu juda qo'rqinchli ko'rinadi: qanday qilib har bir foydalanuvchi profili uchun turli xil ***PageRank taqsimotini*** hisoblashimiz mumkin (potentsial, cheksiz ko'p profillar bilan)? Agar shaxsning qiziqishlari mavzu sahifalarini taqsimlashning oz sonli chiziqli birikmasi sifatida yaqinlashishi mumkin deb taxmin qilsak, buni hal qilishimiz mumkin.



10.5-rasm. **Mavzuga oid PageRank**

Ushbu misolda qiziqishlari 60% sport va 40% siyosat bo'lgan foydalanuvchi ko'rib chiqiladi. Agar teleportatsiya ehtimoli 10% bo'lsa, bu foydalanuvchi 6% sport sahifalariga va 4% siyosat sahifalariga teleportatsiya sifatida modellashtirilgan.

Qiziqishlar aralashmasiga ega foydalanuvchi quyidagi tarzda teleportatsiya qilishi mumkin: avval ma'lum sport sahifalarining **S** to'plamiga yoki ma'lum siyosat sahifalari to'plamiga teleportatsiya qilishni aniqlang. Bu tanlov tasodifiy amalga oshiriladi, 60% vaqt sport sahifalarini va 40% siyosat sahifalarini tanlaydi. Ular ma'lum bir teleport bosqichi tasodifiy sport sahifasi (aytaylik) ekanligini tanlaganimizdan so'ng, ular teleport qilish uchun **S**-dagi veb-sahifani tasodifiy ravishda tanlaymiz. Bu, o'z navbatida, ushbu foydalanuvchining mavzular bo'yicha afzalliklariga moslashtirilgan barqaror holat taqsimotiga ega bo'lgan ergodik Markov zanjiriga olib keladi (10.16-mashqga qarang). Bu g'oya intuitiv jozibador bo'lsa-da, uni amalga oshirish mashaqqatli

ko'rinadi: har bir foydalanuvchi uchun buni talab qiladigan ko'rinadi. O'tish ehtimoli qobiliyati matritsasini hisoblaymiz va uning barqaror holatdagi taqsimotini hisoblaymiz. Ularni Markov zanjirining holatlari bo'yicha ehtimollik taqsimotining evolyutsiyasini chiziqli tizim sifatida ko'rish mumkinligi qutqardi. 10.16-mashqda mavzular bo'yicha foydalanuvchi qiziqishlarining har bir aniq kombinatsiyasi uchun **PageRank** vektorini hisoblash shart emasligini ko'rsatamiz. Har qanday foydalanuvchi uchun shaxsiylashtirilgan ***PageRank vektori*** asosiy mavzuga xos PageRankning chiziqli birikmasi sifatida ifodalanishi mumkin. Masalan, qiziqishlari 60% sport va 40% siyosat bo'lgan foydalanuvchi uchun shaxsiylashtirilgan PageRank vektorini quyidagicha hisoblash mumkin:

$$0.6\vec{\pi}_S = 0.4\vec{\pi}_P$$

bu yerda $\vec{\pi}_s$ va $\vec{\pi}_p$ mos ravishda sport va siyosat uchun mavzuga xos PageRank vektorlari.

10.5-mashq. 10.2-rasmdagi misol uchun o'tish ehtimoli matritsasini yozing.

10.6-mashq. 1, 2 va 3 uchta tugunli veb-grafani ko'rib chiqaylik. Bog'lanishlar quyidagicha: $1 \to 2$, $3 \to 2$, $2 \to 1$, $2 \to 3$. Surferning teleportatsiya bilan yurishi uchun o'tish ehtimoli matritsalarini yozing. Teleport ehtimolining quyidagi uchta qiymati: (a) $a = 0$; (b) $a = 0{,}5$ va (c) $a = 1$.

10.7-mashq. Brauzer foydalanuvchisi o'zi ko'rayotgan **x** sahifasidagi giperhavolani bosishdan tashqari, **x** ga kirgan sahifaga qaytish uchun orqaga tugmasidan foydalanishi mumkin. Orqa tugmachalarning bunday foydalanuvchisini ***Markov zanjiri sifatida modellashtirish*** mumkinmi? Orqaga tugmachasini takroriy chaqiruvlarini qanday modellashimiz mumkin?

10.8-mashq. Uchta A, B va C holatga ega bo'lgan Markov zanjirini va quyidagi o'tish ehtimolini ko'rib chiqing. A holatidan keyingi holat 1 ehtimolli B. B dan keyingi holat yoki $\mathbf{p_A}$ ehtimolli **A** yoki $1 - \mathbf{p_A}$ ehtimolli **C** holatidir. **C** dan keyingi holat **A** ehtimolligi 1. $p_A \in [0,1]$ ning qaysi qiymatlari uchun bu Markov zanjiri ergodik hisoblanadi?

10.9-mashq. Har qanday yo'naltirilgan grafik uchun teleport operatsiyasi bilan tasodifiy yurish natijasida paydo bo'lgan Markov zanjiri ergodik ekanligini ko'rsating.

10.10-mashq. Har bir sahifaning PageRank kamida **a/N** ekanligini ko'rsating. Bu **a** 1 ga yaqinlashganda PageRank qiymatlaridagi farq (turli sahifalar bo'yicha) haqida nimani anglatadi?

10.11-mashq. 10.1-misoldagi ma'lumotlar uchun kichik tartibni yozing yoki (10.6) tenglamada ko'rsatilgan PageRank qiymatlarini hisoblash uchun ilmiy kalkulyatordan foydalaning.

10.12-mashq. Aytaylik, veb-grafik diskda qo'shni ro'yxat sifatida saqlangan, shunday qilib, siz faqat sahifalarning qo'shnilarini saqlash tartibi bo'yicha so'rashingiz mumkin. Grafikni asosiy xotiraga yuklay olmaysiz, lekin to'liq grafik ustida bir necha marta o'qishingiz mumkin. Ushbu sozlamada PageRank ni hisoblash algoritmini yozing.

10.13-mashq. 10.2.3-bo'lim boshiga yaqin kiritilgan **S** va **Y** to'plamlarini eslang. **Y** to'plami **S** bilan qanday bog'liq?

10.14-mashq. Y to'plami har doim barcha veb-sahifalar to'plamimi? Nima uchun yoki nima sababdan?

10.15-mashq. **S**-dagi har qanday sahifaning sport PageRank hech bo'lmaganda PageRank kabi kattami?

10.16-mashq. Har bir veb-sahifa uchun ikkita mavzuga oid PageRank qiymatiga ega bo'lgan sozlamani ko'rib chiqaylik: sport PageRank $\vec{\pi}_s$ va siyosat PageRank $\vec{\pi}_p$. Mavzuga oid PageRanksning ikkala to'plamini hisoblashda ishlatiladigan (umumiy) teleportatsiya ehtimoli a bo'lsin. $q \in [0,1]$ uchun qiziqish profile sportdagi q kasr va siyosatdagi $1 - q$ kasr o'rtasida bo'lingan foydalanuvchini ko'rib chiqing. Foydalanuvchining shaxsiylashtirilgan PageRank tasodifiy yurishining barqaror taqsimoti ekanligini ko'rsating, bunda teleport bosqichida yurish q ehtimoli bo'lgan sport sahifasiga va $1 - q$ ehtimollik bilan siyosat sahifasiga teleportatsiya qilinadi.

10.17-mashq.10.16-mashqdagi yurishga mos keladigan Markov zanjiri ergodic ekanligini va shuning uchun foydalanuvchining shaxsiylashtirilgan PageRank-ni ushbu Markov zanjirining barqaror taqsimotini hisoblash orqali olish mumkinligini ko'rsating.

10.18-mashq. 10.17-mashqning barqaror holat taqsimotida istalgan veb-sahifa **i** uchun barqaror holat ehtimoli $q\pi_S(i) + (1-q)\pi_P(i)$ ga teng ekanligini ko'rsating.

Endi sxemani ishlab chiqamiz, unda so'rov berilganda har bir veb-sahifa imzolangan ikkita balldan iborat. Ulardan biri ***markaz balli***, ikkinchisi esa ***avtoritet reytingi*** deb ataladi. Har qanday so'rov uchun ular bir emas, balki ikkita tartiblangan natijalar ro'yxati hisoblanadi. Bir ro'yxatning reytingi markaz ballari bilan, ikkinchisi esa avtoritet ballari bilan belgilanadi.

Ushbu yondashuv veb-sahifalarni yaratish bo'yicha ma'lum bir tushunchadan kelib chiqadi chunki keng mavzuli qidiruvlar uchun natijalar sifatida foydali bo'lgan ikkita asosiy turdagi veb-sahifalar mavjud. Keng mavzuni qidirish deganda "Men leykemiya haqida ma'lumot olishni xohlayman" kabi ma'lumot so'rovini tushunamiz. Mavzu bo'yicha nufuzli ma'lumot manbalari mavjud. Bu holda, Milliy saraton institutining leykemiya bo'yicha sahifasi shunday sahifa bo'ladi. Bunday sahifalar hokimiyat organlariga chaqiriladi. Ta'riflamoqchi bo'lgan hisoblashda yuqori avtoritet ballari bilan paydo bo'ladigan sahifalardir.

Boshqa tomondan, Internetda ma'lum bir mavzu bo'yicha nufuzli veb-sahifalarga havolalar ro'yxati qo'lda tuzilgan ko'plab sahifalar mavjud. Ushbu markaz sahifalari o'z-o'zidan mavzuga oid ma'lumotlarning nufuzli manbalari emas, balki mavzuga qiziqqan odam yig'ish uchun vaqt sarflagan kompilyatsiyalardir. Shunday qilib, qabul qiladigan yondashuv bu markaz sahifalarini avtoritet sahifalarini ochish uchun ishlatishdir. Hozir ishlab chiqayotgan hisob-kitoblarda bu ko'prik sahifalar yuqori ballari bilan chiqadigan sahifalardir. Yaxshi markaz sahifasi ko'plab yaxshi hokimiyatlarga ishora qiladi. Yaxshi avtoritet sahifasi ko'plab yaxshi markaz sahifalari tomonidan ko'rsatilgan sahifadir. Shunday qilib, markazlar va hokimiyatlarning doiraviy ta'rifiga egamiz. Buni iterativ hisoblashga aylantiramiz. Faraz qilaylik, yaxshi markaz va avtoritet sahifalari hamda ular orasidagi giperhavolalarni o'z ichiga olgan veb-to'plam mavjud. Ushbu kichik to'plamdagi har bir veb-sahifa uchun ball va avtoritet ballini takroriy hisoblab chiqamiz va bu kichik to'plamni qanday tanlashimiz haqidagi muhokamani 10.3.1-bo'limgacha qoldiramiz.

Ularning veb-to'plamimizdagi **v** veb-sahifasi uchun uning ballini va **a(v)** avtoritet ballini belgilash uchun **h(v)** dan foydalaniladi. Dastlab,

barcha **v** tugunlari uchun **h(v)=a(v)=1** ni o'rnatamiz. Shuningdek, **v** dan **y** gacha bo'lgan giperhavola mavjudligini **v** 7 → **y** bilan ham belgilanadi. Iterativ algoritmning asosi 10.8 tenglamada berilgan barcha sahifalarning markaz va vakolat ballarini yangilash juftligi bo'lib, yaxshi markazlar yaxshi vakolatlarga ishora qiladi va yaxshi vakolatlar yaxshi markazlarga ishora qiladi degan intuitiv tushunchalarni qamrab oladi.

$$h(v) \leftarrow \sum_{v \mapsto y} a(y)$$

$$a(v) \leftarrow \sum_{y \mapsto v} h(y)$$

Shunday qilib, tenglamaning birinchi qatori **v** sahifaning markaz ballini u bog'langan sahifalarning vakolat ballari yig'indisiga o'rnatadi. Boshqacha qilib aytadigan bo'lsak, agar **v** yuqori vakolatli ballarga ega bo'lgan sahifalarga havola qilsa, uning markaz balli ortadi. Ikkinchi qator teskari rol o'ynaydi. Agar **v** sahifasi yaxshi markazlar bilan bog'langan bo'lsa, uning vakolat bahosi ortadi. Ushbu yangilanishlarni iterativ ravishda amalga oshirganimizda, markaz ballarini qayta hisoblaganimizda, so'ngra qayta hisoblangan markaz ballari asosida yangi vakolat ballarini va hokazolarni amalga oshirsak nima bo'ladi? (10.8) tenglamani matritsa-vektor ko'rinishga keltiramiz. $\vec{h}$ va $\vec{a}$ veb-grafikning quyi to'plamidagi sahifalar uchun mos ravishda barcha markaz va barcha vakolat ballarining vektorlarini bildirsin. Ular ko'rib chiqayotgan veb-grafik to'plamining qo'shnilik matritsasi **A** bilan belgilansin: **A** - bu kichik to'plamdagi har bir sahifa uchun bitta satr va bitta ustunli kvadrat matritsa. $\mathbf{A}_{ij}$ yozuvi agar **i** sahifadan **j** sahifaga giperhavola bo'lsa 1 ga, aks holda 0 ga teng. Keyin (10.8) tenglamani yozishimiz mumkin.

$$\vec{h} \leftarrow A\vec{a}$$

$$\vec{a} \leftarrow A^T\vec{h},$$

Bu yerda A^T matritsaning ko'chirilishini bildiradi. Endi (10.9) tenglamaning har bir satrining o'ng tomoni boshqa (10.9) tenglamaning chap tomoni bo'lgan vektordir. Bularni bir-biriga almashtirib, (10.9) tenglamani shunday yozishimiz mumkin:

$$\vec{h} \leftarrow AA^T\vec{h}$$

$$\vec{a} \leftarrow A^TA\vec{a}$$

Endi (10.10) tenglama bir juft xos vektor tenglamalariga g'alati o'xshaydi (7.1-bo'lim); Haqiqatan ham, agar ular ← belgilarini = belgilari bilan almashtirsak va (noma'lum) xos qiymatni kiritsak, (10.10) tenglamaning birinchi qatori AA^T ning xos vektorlari uchun tenglamaga aylanadi, ikkinchisi esa A^TA xos vektorlari uchun tenglamaga aylanadi:

$$\bar{h} \leftarrow (1/\lambda_h) AA^T\bar{h}$$

$$\bar{a} \leftarrow (1/\lambda_a) A^TA\bar{a}$$

Bu yerda $\mathbf{AA^T}$ ning xos qiymatini belgilash uchun $\mathbf{1/L_h}$ va $\mathbf{A^TA}$ ning xos qiymatini belgilash uchun 1 dan foydalandik. Bu ba'zi muhim oqibatlarga olib keladi:

1. Tenglama (10.8) (yoki ekvivalenti (10.9)) dagi takroriy yangilanishlar, agar tegishli xos qiymatlar bilan masshtablangan bo'lsa, AA^T va A^TA xos vektorlarini hisoblash uchun quvvatni takrorlash usuliga ekvivalentdir. AA^T ning asosiy xos qiymati yagona bo'lgan taqdirda, $\bar{h}$ va $\bar{a}$ ning takroriy hisoblangan yozuvlari **A** yozuvlari va demak, grafikning bog'lanish tuzilishi bilan aniqlangan yagona barqaror holat qiymatlariga joylashadi.

2. Ushbu xos vektorli yozuvlarni hisoblashda quvvatni takrorlash usulidan foydalanish bilan cheklanmaymiz. Haqiqatan ham, stoxastik matritsaning asosiy xos vektorini hisoblash uchun har qanday tezkor usuldan foydalanishimiz mumkin. Natijada hisob-kitob quyidagi shaklni oladi:

1. Veb-sahifalarning maqsadli to'plamini yig'ing, ularning giperhavolalari bilan bog'liq grafikni tuzing va AA^T va A^TA ni hisoblang.

2. AA^T va A^TA ning asosiy xos vektorlarini hisoblab, markaz ballari $\bar{h}$ va avtoritet ballari $\bar{a}$ vektorini hosil qiling.

3. Eng yuqori ball to'plagan markazlarni va yuqori ball to'plagan organlarni chiqaring.

HITS. Ushbu havolani tahlil qilish usuli **HITS** deb nomlanadi, bu *Hyperlink-Induced* Mavzular Qidiruvining qisqartmasi hisoblanadi.

10.2-misol: langarlar so'rov so'zini, matritsani o'z ichiga oladi. Rasm uchun jaguar A so'rovini va 10.4 quyidagi havolalarni ikki marta og'irlashtirgan holda:

$$\begin{matrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 2 & 1 & 0 & 1 \end{matrix}$$

Markaz va vakolat vektorlari quyidagilardir:

$$\bar{h} = (0.03 \quad 0.04 \quad 0.33 \quad 0.18 \quad 0.04 \quad 0.04 \quad 0.35)$$

$$\bar{a} = (0.10 \quad 0.01 \quad 0.12 \quad 0.47 \quad 0.16 \quad 0.01 \quad 0.13)$$

Bu yerda $\mathbf{q_3}$ asosiy vakolatdir - ikkita markaz ($\mathbf{q_2}$ va $\mathbf{q_6}$) yuqori vaznli yaguar havolalari orqali unga ishora qiladi. Takroriy yangilanishlar yaxshi markazlar va yaxshi vakolatli organlarning sezgisini qamrab olganligi sababli, chiqargan yuqori ballli sahifalar bizga veb-sahifalarning maqsadli to'plamidan yaxshi markazlar va vakolatlarni beradi. 10.3.1-bo'limda qolgan tafsilotlarni tasvirlaymiz: leykemiya kabi mavzu bo'yicha veb-sahifalarning maqsadli to'plamini qanday yig'ishimiz mumkin?

10.3.1. Internetning kichik to'plamini tanlash

Leykemiya kabi mavzu bo'yicha veb-sahifalarning kichik to'plamini yig'ishda yaxshi vakolatli sahifalarda leykemiya so'rovining o'ziga xos atamasi bo'lmasligi mumkinligi bilan kurashishimiz kerak. Bu, ayniqsa, 10.1.1-bo'limda ta'kidlaganimizdek, avtoritet sahifasi ma'lum bir marketing imidjini loyihalash uchun o'z veb-saytidan foydalanganda to'g'ri keladi. Misol uchun, **IBM** veb-saytidagi ko'plab sahifalar kompyuter texnikasi bo'yicha ishonchli ma'lumot manbalari hisoblanadi, garchi bu sahifalarda kompyuter yoki apparat atamasi bo'lmasa ham. Biroq, kompyuter uskunalari resurslarini kompilyatsiya qiluvchi markaz bu atamalardan foydalanishi va IBM veb-saytidagi tegishli sahifalarga havola qilishi mumkin. Ushbu kuzatishlarga asoslanib, markaz va avtoritet ballarini hisoblash uchun Internetning quyi to'plamini tuzish uchun quyidagi tartib taklif qilindi.

1. So'rov berilganda (aytaylik, leykemiya), leykemiyani o'z ichiga olgan barcha sahifalarni olish uchun matn indeksidan foydalaning. Buni ***sahifalarning ildiz to'plami*** deb nomlang.

2. Ildiz to'plamini, shuningdek, ildiz to'plamidagi sahifaga bog'laydigan yoki ildiz to'plamidagi sahifa bilan bog'langan har qanday

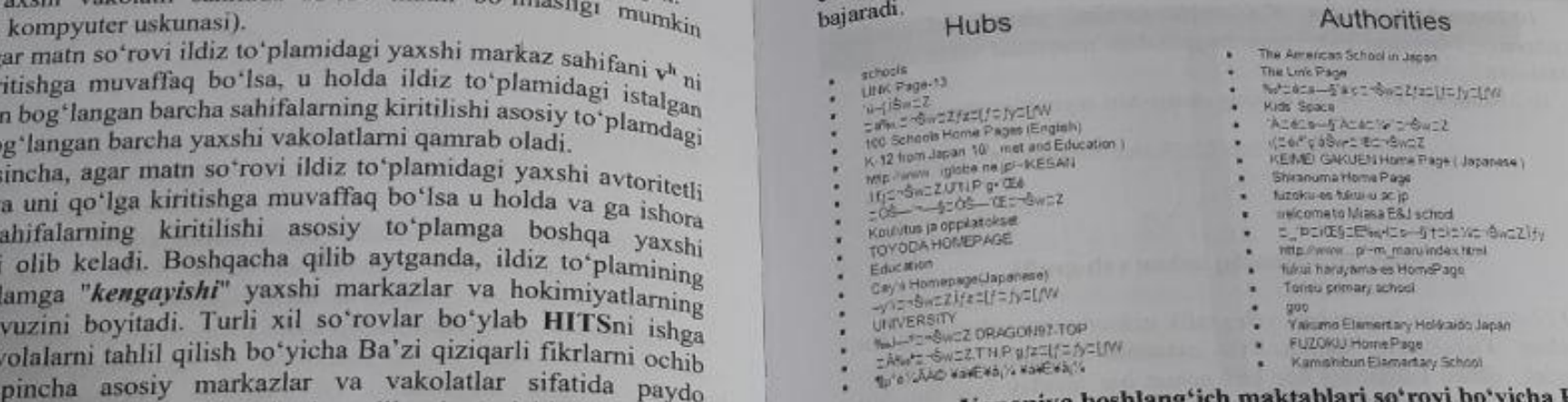
sahifani o'z ichiga olish uchun asosiy sahifalar to'plamini yarating. Keyin hisoblash markazi va avtoritet ballari uchun baza to'plamidan foydalaniladi. Asosiy to'plam uchta sababga ko'ra shu tarzda qurilgan:

1. Yaxshi vakolatli sahifada so'rov matni bo'lmasligi mumkin (masalan, kompyuter uskunasi).

2. Agar matn so'rovi ildiz to'plamidagi yaxshi markaz sahifani v^h ni qo'lga kiritishga muvaffaq bo'lsa, u holda ildiz to'plamidagi istalgan sahifa bilan bog'langan barcha sahifalarning kiritilishi asosiy to'plamdagi v^h bilan bog'langan barcha yaxshi vakolatlarni qamrab oladi.

3. Aksincha, agar matn so'rovi ildiz to'plamidagi yaxshi avtoritetli sahifa va va uni qo'lga kiritishga muvaffaq bo'lsa u holda va ga ishora qiluvchi sahifalarning kiritilishi asosiy to'plamga boshqa yaxshi markazlarni olib keladi. Boshqacha qilib aytganda, ildiz to'plamining asosiy to'plamga "***kengayishi***" yaxshi markazlar va hokimiyatlarning umumiy hovuzini boyitadi. Turli xil so'rovlar bo'ylab **HITS**ni ishga tushirish havolalarni tahlil qilish bo'yicha Ba'zi qiziqarli fikrlarni ochib beradi. Ko'pincha asosiy markazlar va vakolatlar sifatida paydo bo'ladigan hujjatlar so'rov tilidan boshqa tillarni o'z ichiga oladi. Bu sahifalar, ehtimol, ildiz to'plamining yig'ilishidan so'ng, asosiy to'plamga o'tadi. Shunday qilib, bu yerda tillar aro qidiruvning Ba'zi elementlari (bir tildagi so'rov boshqa tildagi hujjatlarni oladi) aniq ko'rinadi. Qizig'i shundaki, bu tillararo ta'sir faqat havolalarni tahlil qilish natijasida yuzaga kelgan, lingvistik tarjima amalga oshirilmagan. Ushbu bo'limni ushbu algoritmni amalga oshirish bo'yicha ba'zi eslatmalar bilan yakunlaymiz. Ildiz to'plami matn so'roviga mos keladigan barcha sahifalardan iborat. Aslida, amalga oshirishlar (10.4-bo'limdagi havolalarni ko'ring) matn so'roviga mos keladigan barcha sahifalardan ko'ra ildiz to'plami uchun 200 yoki undan ortiq veb-sahifalardan foydalanish kifoya qiladi. Markaz/avtorite ball vektorini hisoblash uchun xos vektorlarni hisoblash uchun har qanday algoritmdan foydalanish mumkin. Aslida, bu ballarning aniq qiymatlarini hisoblashimiz shart emas. Yuqori markazlar va hokimiyatlarni aniqlashimiz uchun ballarning nisbiy qiymatlarini bilish kifoya. Shu maqsadda, quvvat iteratsiyasi usulining oz sonli iteratsiyasi yuqori markazlar va hokimiyatlarning nisbiy tartibini keltirib chiqarishi mumkin. Tajribalar shuni ko'rsatdiki, amalda (10.8) tenglamaning beshga yaqin takrorlanishi juda yaxshi natijalar beradi. Bundan tashqari, veb-grafaning havola tuzilishi juda siyrak bo'lganligi sababli (o'rtacha veb-sahifa

taxminan o'ntaga bog'langan), ularni matritsa-vektor mahsuloti sifatida emas, balki (10.8) tenglamadagi kabi qo'shimcha yangilanishlar sifatida bajaradi.



10.6-rasm. **Yaponiya boshlang'ich maktablari so'rovi bo'yicha HITS namunasi**

10.6-rasmda yapon boshlang'ich maktablari so'rovi bo'yicha **HITS**ni ishga tushirish natijalari ko'rsatilgan. Rasmda yuqori markazlar va hokimiyatlar ko'rsatilgan; har bir qatorda tegishli HTML sahifasidan sarlavha tegi keltirilgan. Natijada paydo bo'lgan satr lotin harflarida bo'lishi shart emasligi sababli, natijada chop etish (ko'p hollarda) gibberish qatoridir. Ularning har biri lotin harflaridan foydalanmaydigan veb-sahifaga mos keladi, bu holda yapon tilidagi sahifalar bo'lishi mumkin. Boshqa ingliz bo'lmagan tillardagi sahifalar ham bor ko'rinadi, bu so'rovlar qatori ingliz tilida bo'lganligi sababli ajablanarli ko'rinadi. Aslida, bu natija **HITS** faoliyatining ramzi hisoblanadi - ildiz to'plami yig'ilgandan so'ng (inglizcha) so'rovlar qatori e'tiborga olinmaydi. Asosiy to'plamda boshqa tillardagi sahifalar bo'lishi mumkin, masalan, ingliz tilidagi markaz sahifasi yapon boshlang'ich maktablarining yapon tilidagi bosh sahifalariga havola qilsa. Yuqori markazlar va vakolatlarning keyingi hisob-kitoblari butunlay havolaga asoslanganligi sababli, ingliz tilidan tashqari sahifalarning ba'zilari eng yaxshi markazlar va vakolatlar orasida paydo bo'ladi.

10.19-mashq. Agar barcha markaz va vakolat ballari 1 ga ishga tushirilsa, bir iteratsiyadan keyin tugunning hub/avtorite balli qanday bo'ladi?

10.20-mashq. AA^T va A^TA matritsalarining yozuvlarini qanday izohlaysiz? 7-bobdagi CC^T ning birgalikdagi matritsasi bilan qanday aloqasi bor?

10.21-mashq. AA^T va A^TA ning asosiy xos qiymatlari qanday?

10.7-rasm. **10.22-mashq uchun veb-grafik**

10.22-mashq. 10.7-rasmdagi veb-grafik uchun uchta sahifaning har biri uchun **PageRank**, markaz va avtoritet ballarini hisoblang. Shuningdek, ushbu ballarning har biri uchun har qanday bog'lanishni ko'rsatadigan 3 ta tugunning nisbiy tartibini bering. **PageRank**: Faraz qilaylik, PageRank tasodifiy yurishining har bir bosqichida 0,1 ehtimollik bilan tasodifiy sahifaga teleportatsiya qiladi va qaysi sahifaga teleportatsiya qiladi. Hublar/vakolatlar: Hub (hokimiyat) ballarini maksimal hub (avtoritet) balli 1 bo'lishi uchun normallashtiring. 1-maslahat: Simmetriyalarni soddalashtirish va chiziqli tenglamalar yordamida yechish iterativ usullardan foydalanishdan osonroq bo'lishi mumkin. 2-maslahat: uchta baholash o'lchovining har biri uchun uchta tugunning nisbiy tartibini (har qanday bog'lanishni ko'rsatib) taqdim eting.

10- bob bo'yicha foydalanilgan adabiyotlar

Trotman, Andrew, Shlomo Geva, and Jaap Kampseds.).
2007.
SIGIR Workshop on Focused Retrieval. University of Otago.

Trotman, Andrew, Nils Pharo, and Miro Lehtonen.
2006.
XML-IR users and use cases.
In *Proc. INEX*, pp. 400-412.

Tseng, Huihsin, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning.
2005.
A conditional random field word segmenter.
In *SIGHAN Workshop on Chinese Language Processing*.

Tsochantaridis, Ioannis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun.
2005.
Large margin methods for structured and interdependent output variables.
JMLR 6: 1453-1484.

Turpin, Andrew, and William R. Hersh.
2001.
Why batch and user evaluations do not give the same results.
In *Proc. SIGIR*, pp. 225-231.

Turpin, Andrew, and William R. Hersh.
2002.
User interface effects in past batch versus user experiments.
In *Proc. SIGIR*, pp. 431-432.

Turpin, Andrew, Yohannes Tsegay, David Hawking, and Hugh E. Williams.
2007.
Fast generation of result snippets in web search.
In *Proc. SIGIR*, pp. 127-134. ACM Press.

Kazai, Gabriella va Mounia Lalmas.
2006.
Kontentga yo'naltirilgan XML qidirishni baholash uchun kengaytirilgan yig'ilgan daromad o'lchovlari.
TOIS 24 (4): 503-542.
DOI: *doi.acm.org/10.1145/1185883* .

Kekälaynen, Yaana.
2005.
IR baholashda ikkilik va darajali dolzarblik - IR tizimlarining reytingiga ta'sirini taqqoslash.
IP&M 41: 1019-1033.

Kekäläinen, Yaana va Kalervo Järvelin.

2002.
IRni baholashda darajali muvofiqlik baholashlaridan foydalanish.
JASIST 53 (13): 1120-1129.
Kemeny, Jon G. va J. Laurie Snell.
1976.
Cheklangan Markov zanjirlari .
Springer.
Kent, Allen, Madeline M. Berry, Fred U. Luehrs, Jr. va J. W. Perry.
1955.
Mashina adabiyotlarini qidirish VIII. Axborot-qidiruv tizimlarini loyihalashning operatsion mezonlari.
Amerika hujjatlari 6 (2): 93-101.

10- bob bo'yicha nazariy va amaliy test savollari

1. Qanday javob kalitga yoki takroriy qiymatlarga ega bo'lgan kalit bo'lmagan maydonga asoslangan indeks turi sifatida tasniflanadi?
   A) Ikkilamchi indeks
   B) Uchlik indeks
   C) Asosiy indeks
   D) Klasterlash indeksi
2. Har bir kalit qiymati uchun yozuvga ega indeks qanday nomlanadi?
   A) Zich indeks
   B) Chiziqli indeks
   C) Siyrak indeks
   D) Klaster indeks
3. Daraxt tuzilmasidagi ko'p tugun va bitta ota-ona tuguniga ega bo'lgan maxsus tugun nima deb nomlanadi?
   A) Ildiz tugun
   B) Nasl tugunlari
   C) Barg tugunlari
   D) Qidiruv tugunlari
4. Ma'lumotlar ko'rsatkichlari diagrammaning barg tugunlarida saqlanadigan daraxt strukturasi diagrammasi sifatida qanday tasniflanadi?
   A) B+ daraxt
   B) B daraxt
   C) B^2 daraxt
   D) B^* daraxt

5. O'z bloklariga yangi yozuvlar kiritish uchun bo'sh joy qoldiradigan ko'p darajali indeks turi nima deyiladi?
   A) Dinamik ko'p darajali indeks
   B) Statik ko'p darajali indeks
   C) Zich bo'lmagan ko'p darajali indeks
   D) Zich ko'p darajali indeks
6. Daraxt tuzilishida tugunlardan holi tugun nima deyiladi?
   A) Barg tugun
   B) Nasl tugun
   C) Ildiz tugun
   D) Qidiruv tugun
7. Yozuvlar faqat ikkita maydon bilan belgilangan uzunlikka ega bo'lgan indeks turi sifatida qanday tasniflanadi?
   A) Asosiy indeksi
   B) Langar indeksi
   C) Klaster indeksi
   D) Ikkilamchi indeksi
8. Daraxt tuzilishi diagrammalarida barg barg bo'lmagan tugun nima deyiladi?
   A) Tashqi tugun
   B) Qidiruv tugun
   C) Nasl tugun
   D) Ichki tugun
9. Diskdan ma'lumotlarni uzluksiz oqim bloklari ko'rinishida olish uchun foydalanadigan va qidirish vaqtini yo'q qiladigan usul qanday tasniflanadi?
   A) Ikki marta buferlash
   B) Bir vaqtda buferlash
   C) Parallel buferlash
   D) Yagona buferlash
10. Asosiy hotirada bitta blokni saqlash uchun ajratilgan maydon qanday nomlanadi?
   A) Buffer manzili
   B) Disk manzili
   C) Apparat manzili
   D) Dasturiy ta'minot manzili

FOYDALANILGAN ADABIYOTLAR RO'YXATI

1. Aberer, Karl. 2001. P-Grid: A self-organizing access structure for P2P information systems. In Proc. International Conference on Cooperative Information Systems, pp. 179–194. Springer. xxxiv, 521

2. Allan, James. 2005. HARD track overview in TREC 2005: High accuracy retrieval from documents. In Proc. TREC. 174, 521

3. Allwein, Erin L., Robert E. Schapire, and Yoram Singer. 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. JMLR 1:113–141. URL: www.jmlr.org/papers/volume1/allwein00a/allwein00a.pdf. 315, 521, 532, 533

4. Alonso, Omar, Sandeepan Banerjee, and Mark Drake. 2006. GIO: A semantic web application using the information grid framework. In Proc. WWW, pp. 857–858. ACM Press. DOI: doi.acm.org/10.1145/1135777. 1135913.373, 521, 524

5. Gerhard Weikum. 2005. Report on the DB/IR panel at SIGMOD 2005. SIGMOD Record 34(4):71–74. DOI: doi.acm.org/10.1145/1107499. 1107514.217, 521, 523, 532, 534

6. Amer-Yahia, Sihem, and Mounia Lalmas. 2006. XML search: Languages, INEX and scoring. SIGMOD Record 35(4):16–23. DOI: doi.acm.org/10.1145/1228268.1228271. 217, 521, 528

7. Anagnostopoulos, Aris, Andrei Z. Broder, and Kunal Punera. 2006. Effective and efficient classification on a search-engine model. In Proc. CIKM, pp. 208–217. ACM Press. DOI: doi.acm.org/10.1145/1183614. 1183648. 315, 521, 522, 531

8. Andoni, Alexandr, Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab Mirrokni. 2006. Locality-sensitive hashing using stable distributions. In Nearest Neighbor Methods in Learning and Vision: Theory and Practice. MIT Press. 314, 521, 524, 526, 529

9. Anh, Vo Ngoc, and Alistair Moffat. 2005. Inverted index compression using word-aligned binary codes. IR 8(1):151–166. DOI: dx.doi.org/10.1023/B: INRT.0000048490.99518.5c. 106, 521, 529

10. Anh, Vo Ngoc, and Alistair Moffat. 2006c. Structured index organizations for highthroughput text querying. In Proc. SPIRE, pp. 304–315. Springer. 149, 521, 529

11. Arthur, David, and Sergei Vassilvitskii. 2006. How slow is the k-means method? In Proc. ACM Symposium on Computational Geometry, pp. 144–153. 373, 521, 534

12. Aslam, Javed A., and Emine Yilmaz. 2005. A geometric interpretation and analysis of R-precision. In Proc. CIKM, pp. 664–671. ACM Press. 174, 521, 535

13. Baeza-Yates, Ricardo, Paolo Boldi, and Carlos Castillo. 2005. The choice of â damping function for propagating importance in link-based ranking. Technical report, Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano. 481, 521, 522, 523

14. Bar-Ilan, Judit, and Tatyana Gutman. 2005. How do search engines respond to some non-English queries? Journal of Information Science 31(1):13–28. 46, 521, 525

15. Bar-Yossef, Ziv, and Maxim Gurevich. 2006. Random sampling from a search engine's index. In Proc. WWW, pp. 367–376. ACM Press. DOI: doi.acm.org/10.1145/1135777.1135833. 442, 521, 525

16. Bast, Holger, and Debapriyo Majumdar. 2005. Why spectral retrieval works. In Proc. SIGIR, pp. 11–18. ACM Press. DOI: doi.acm.org/10.1145/ 1076034.1076040. 417, 522, 529

17. Berkhin, Pavel. 2005. A survey on pagerank computing. Internet Mathematics 2(1): 73–120. 481, 522

18. Berkhin, Pavel. 2006a. Bookmark-coloring algorithm for personalized pagerank computing. Internet Mathematics 3(1):41–62. 481, 522

19. Berkhin, Pavel. 2006b. A survey of clustering data mining techniques. In Jacob Kogan,

20. Betsi, Stamatina, Mounia Lalmas, Anastasios Tombros, and Theodora Tsikrika. 2006. User expectations from XML element retrieval. In Proc. SIGIR, pp. 611–612. ACM Press. 217, 522, 528, 533, 534

21. Bishop, Christopher M. 2006. Pattern Recognition and Machine Learning. Springer. 315, 522

22. Blanco, Roi, and Alvaro Barreiro. 2006. TSP and cluster-based solutions to the reassignment of document identifiers. IR 9(4):499–517. 106, 521, 522

23. Blanco, Roi, and Alvaro Barreiro. 2007. Boosting static pruning of inverted files. In Proc. SIGIR. ACM Press. 105, 521, 522

24. Boldi, Paolo, Massimo Santini, and Sebastiano Vigna. 2005. PageRank as a function of the damping factor. In Proc. WWW. URL: citeseer.ist.psu.edu/ boldi05pagerank.html. 481, 522, 532, 534

25. Boldi, Paolo, and Sebastiano Vigna. 2005. Compressed perfect embedded skip lists for quick inverted-index lookups. In Proc. SPIRE. Springer. 46, 522, 534

26. Burges, Chris, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In Proc. ICML. 348, 522, 524, 525, 526, 528, 531, 532

27. Cao, Guihong, Jian-Yun Nie, and Jing Bai. 2005. Integrating word relationships into language models. In Proc. SIGIR, pp. 298–305. ACM Press. 252, 521, 523, 530

28. Cao, Yunbo, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. 2006. Adapting Ranking SVM to document retrieval. In Proc. SIGIR. ACM Press. 348, 523, 526, 528, 535

29. Caruana, Rich, and Alexandru Niculescu-Mizil. 2006. An empirical comparison of supervised learning algorithms. In Proc. ICML. 347, 523, 530

30. Chierichetti, Flavio, Alessandro Panconesi, Prabhakar Raghavan, Mauro Sozio, Alessandro Tiberi, and Eli Upfal. 2007. Finding near neighbors through cluster pruning. In Proc. PODS. 149, 523, 530, 531, 533, 534

31. Cho, Junghoo, and Hector Garcia-Molina. 2012. Parallel crawlers. In Proc. WWW, pp. 124–135. ACM Press. DOI:doi.acm.org/10.1145/511446. 511464.458, 523, 525

32. Ferragina, Paolo, and Rossano Venturini. 2007. Compressed permuterm indexes. In Proc. SIGIR. ACM Press. 65, 524, 534

33. Forman, George. 2006. Tackling concept drift by temporal inductive transfer. In Proc. SIGIR, pp. 252–259. ACM Press. DOI: doi.acm.org/10. 1145/1148170.1148216. 286, 525

34. Fuhr, Norbert, and Mounia Lalmas. 2007. Advances in XML retrieval: The INEX initiative. In International Workshop on Research Issues in Digital Libraries. 216, 525, 528

35. Fuhr, Norbert, Mounia Lalmas, Saadia Malik, and Gabriella Kazai (eds.). 2006. Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005. Springer. 216, 525, 527, 528, 529

36. Gao, Jianfeng, Mu Li, Chang-Ning Huang, and Andi Wu. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. Computational Linguistics 31(4):531–574. 46, 525, 526, 528, 535

37. Gerrand, Peter. 2007. Estimating linguistic diversity on the internet: A taxonomy to avoid pitfalls and paradoxes. Journal of Computer-Mediated Communication 12(4). URL: jcmc.indiana.edu/vol12/issue4/gerrand.html. article 8. 30, 525

38. Hearst, Marti A. 2006. Clustering versus faceted categories for information exploration. CACM 49(4):59–61. DOI: doi.acm.org/10.1145/1121949. 1121983.372, 526

39. Hiemstra, Djoerd, and Wessel Kraaij. 2005. A language-modeling approach to TREC. In Voorhees and Harman (2005), pp. 373–395. 252, 526, 528

40. Huang, Yifen, and Tom M. Mitchell. 2006. Text clustering with extended user feedback. In Proc. SIGIR, pp. 413–420. ACM Press. DOI:doi.acm.org/ 10.1145/1148170.1148242. 374, 526, 529

41. Hughes, Baden, Timothy Baldwin, Steven Bird, Jeremy Nicholson, and Andrew MacKinlay. 2006. Reconsidering language identification for written language resources. In Proc. International Conference on Language Resources and Evaluation, pp. 485–488. 46, 521, 522, 526, 529, 530

42. Kazai, Gabriella, and Mounia Lalmas. 2006. eXtended cumulated gain measures for the evaluation of content-oriented XML retrieval. TOIS 24(4):503–542. DOI: doi.acm.org/10.1145/1185883. 217, 527, 528

43. Langville, Amy, and Carl Meyer. 2006. Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press. 481, 528, 529

44. Larson, Ray R. 2005. A fusion approach to XML structured document retrieval. IR 8 (4):601–629. DOI: dx.doi.org/10.1007/s10791-005-0749-0. 216, 528

45. Liddy, Elizabeth D. 2005. Automatic document retrieval. In Encyclopedia of Language and Linguistics, 2nd edition. Elsevier. 17, 528

46. List, Johan, Vojkan Mihajlovic, Georgina Ramírez, Arjen P. Vries, Djoerd Hiemstra, and Henk Ernst Blok. 2005. TIJAH: Embracing IR methods in XML databases. IR 8(4):547–570. DOI: dx.doi.org/10.1007/s10791-005-0747-2. 216, 522, 526, 528, 529, 531, 534

47. Lu, Wei, Stephen E. Robertson, and Andrew MacFarlane. 2007. CISR at INEX 2006. In Fuhr et al. (2007), pp. 57–63. 216, 529, 531

48. Nigam, Kamal, Andrew McCallum, and Tom Mitchell. 2006. Semi-supervised text classification using EM. In Chapelle et al. (2006), pp. 33–56. 347, 529, 530

49. Ntoulas, Alexandros, and Junghoo Cho. 2007. Pruning policies for two-tiered inverted index with correctness guarantee. In Proc. SIGIR, pp. 191–198. ACM Press. 105, 523, 530

50. Ogilvie, Paul, and Jamie Callan. 2005. Parameter estimation for a simple hierarchical generative model for XML retrieval. In Proc. INEX, pp. 211–224. DOI: dx.doi.org/10.1007/11766278_16. 216, 523, 530

51. Pirolli, Peter L. T. 2007. Information Foraging Theory: Adaptive Interaction With Information. Oxford University Press. 373, 531

52. Richardson, M., A. Prakash, and E. Brill. 2006. Beyond PageRank: machine learning for static ranking. In Proc. WWW, pp. 707–715. 348, 522, 531

53. Riezler, Stefan, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In Proc. ACL, pp. 464–471. Association for Computational Linguistics. URL: www.aclweb.org/anthology/P/P07/P07-1059. 194, 529, 531, 534

54. Sakai, Tetsuya. 2007. On the reliability of information retrieval metrics based on graded relevance. IP&M 43(2):531–548. 174, 532

55. Tao, Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. 2006. Language model information retrieval with document expansion. In Proc. Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics, pp. 407–414. 252, 529, 533, 534, 535

56. Weikum. 2008. TopX: Efficient and versatile top-k query processing for semistructured data. VLDB Journal 17(1):81–115. 216, 522, 529, 532, 533, 534

57. Theobald, Martin, Ralf Schenkel, and Gerhard Weikum. 2005. An efficient and versatile query engine for TopX search. In Proc. VLDB, pp. 625–636. VLDB Endowment. 216, 532, 533, 534

58. Treeratpituk, Pucktada, and Jamie Callan. 2006. An experimental study on automatically labeling hierarchical clusters using statistical features. In Proc. SIGIR, pp. 707–708. ACM Press. DOI: doi.acm.org/10.1145/1148170. 1148328. 400, 523, 534

59. Trotman, Andrew, Nils Pharo, and Miro Lehtonen. 2006. XML-IR users and use cases. In Proc. INEX, pp. 400–412. 216, 528, 531, 534

60. Tseng, Huihsin, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter. In SIGHAN Workshop on Chinese Language Processing. 46, 521, 523, 527, 529, 534

61. Turpin, Andrew, Yohannes Tsegay, David Hawking, and Hugh E. Williams. 2007. Fast generation of result snippets in web search. In Proc. SIGIR, pp. 127–134. ACM Press. 174, 526, 534, 535

62. Witten, Ian H., and Eibe Frank. 2005. Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition. Morgan Kaufmann. 374, 525, 535

63. Christopher D. Manning Prabhakar Raghavan Hinrich Schutze. An Introduction to Information Retrieval, Cambridge University Press — Cambridge, 2009. Режим доступа: http://nlp.stanford.edu/IR-book/information-retrieval-book

64. Кристофер Д.Маннинг(Стенфордский университет), Прабхакар Рагхаван (Yahoo! Research), Хайнрих Шютце (Университет Штуттарта) Введение в информационный поиск, Москва • Санкт-Петербург • Киев 2011

65. В.С.Гусев Google эффективный поиск краткое руководство, диалектика Москва, Санкт-Петербург, Киев, 2006

66. Т.В.Батура. Математическая лингвистика и автоматическая обработка текстов на естественном языке. учеб. пособие / Т. В. Батура ; Новосиб. гос. ун-т. – Ново- сибирск : РИЦ НГУ, 2016. – 166 с.

67. Т.В.Батура, М. В.Чаринцева. Основы обработки текстовой информации. Учебное пособие. Новосибирск 2016 г.

68. Лукашевич Н.В. Автоматический анализ тональности текстов по отношению к заданному объекту и его характеристикам // Электронные библиотеки. 2015. Т. 18. № 3-4. С. 88-119

MUNDARIJA

Mallayev Oybek Usmankulovich

AXBOROTLARNI IZLASH VA AJRATIB OLISH

(2-QISM)

Muhammad Al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti tomonidan darslik sifatida tavsiya etilgan

Toshkent - "METODIST NASHRIYOTI" - 2024

Muharrir: Bakirov Nurmuhammad

Texnik muharrir: Tashatov Farrux
Musahhih: Hazratqulova Ruxshona
Dizayner: Ochilova Zarnigor

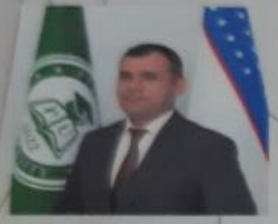
Bosishga 18.05.2024.da ruxsat etildi.
Bichimi 60x90. "Times New Roman" garniturasi.
Ofset bosma usulida bosildi.
Shartli bosma tabog'i 20. Nashr bosma tabog'i 19,12.
Adadi 300 nusxa.

***"METODIST NASHRIYOTI"** MCHJ matbaa bo'limida chop etildi.*
Manzil: Toshkent shahri, Shota Rustaveli 2-vagon tor ko'chasi, 1-uy.

+99893 552-11-21

Nashriyot roziligisiz chop etish ta'qiqlanadi.

AXBOROTLARNI IZLASH VA AJRATIB OLISH



Mallayev Oybek Usmankulovich

Texnika fanlari falsafa doktori (PhD), dotsent
Perfect University Raqamli texnologiyalari kafedrasi mudiri

ISBN 978-9910-03-104-5

9 789910 031045