

ASSOTSIATIV QOIDALARNI IZLASH USULLARI VA ALGORITMLARI

Quljanova Shukurjon Zaribovna

Urganch RANCH texnologiya universiteti

“Raqamli texnologiyalar”

kafedra o‘qituvchi - stajyor

Email: quljanova83@gmail.com

Tel: +998 88 6005035

Khujayev Otabek Kadambayevich

Abu Rayhon Beruniy nomidagi Urganch davlat universiteti

“Kompyuter injineri” kafedra dotsenti, PhD.

Email: otabek.hujaev@gmail.com

Tel: +998 97 513-27-86

Shonazarov Jayxun Anvarovich

Yangiariq tumanidagi 31-son maktabning

11-sinf o‘qituvchisi

Email: jayxunshonazarov9@gmail.com

Tel: +998 878115035

Annotatsiya: Maqolada katta tranzaksion ma’lumotlar bazalaridan assotsiativ qoidalarni aniqlash masalasi, uning formal modeli va hisoblash bosqichlari ikki fundamental ilmiy manba asosida tahlil qilindi. Birinchi manbada support va confidence cheklovlari, tez-tez uchraydigan elementlar to‘plamini topish, baholash, kesish va xotirani boshqarish usullari yoritilgan [1]. Ikkinchi manbada qidiruv fazosini panjara va ekvivalentlik sinflariga ajratish, vertikal TID-list shakli, Eclat, MaxEclat, Clique va MaxClique algoritmlari hamda ularning Apriori va Partition bilan tajribaviy qiyosi ko‘rib chiqilgan [2]. Tahlil natijasi algoritmlar tanlash faqat bajarilish vaqtiga emas, ma’lumotlar zichligi, minimal support, eng uzun tez-tez uchraydigan to‘plam hajmi, bazani skanerlash soni va tezkor xotira chekloviga bog‘liqligini ko‘rsatadi. Zich ma’lumotlarda vertikal shakl va gibrid qidiruv sezilarli

ustunlik berishi, siyrak bazalarda esa sodda darajama-daraja qidiruv amaliy jihatdan yetarli bo‘lishi mumkin.

Kalit so‘zlar: *Assotsiativ qoida, tez-tez uchraydigan elementlar to‘plami, support, confidence, Apriori, Partition, Eclat, MaxClique, TID-list, panjara, pruning.*

Аннотация: В статье на основе двух фундаментальных научных источников проанализирована задача выявления ассоциативных правил в больших транзакционных базах данных, включая её формальную модель и этапы вычисления. В первом источнике рассмотрены ограничения поддержки (support) и достоверности (confidence), методы поиска и оценки часто встречающихся наборов элементов, отсечения вариантов и управления памятью [1]. Во втором источнике исследованы разбиение пространства поиска на решётку и классы эквивалентности, вертикальное представление в виде TID-списков, алгоритмы Eclat, MaxEclat, Clique и MaxClique, а также результаты их экспериментального сравнения с алгоритмами Apriori и Partition [2]. Результаты анализа показывают, что выбор алгоритма зависит не только от времени выполнения, но и от плотности данных, минимального порога поддержки, размера самого длинного часто встречающегося набора, количества сканирований базы данных и ограничений оперативной памяти. Для плотных данных вертикальное представление и гибридный поиск могут обеспечить значительное преимущество, тогда как для разреженных баз данных простой поуровневый поиск может оказаться практически достаточным.

Ключевые слова: *Ассоциативное правило, часто встречающийся набор элементов, поддержка, достоверность, Apriori, Partition, Eclat, MaxClique, TID-список, решётка, отсечение.*

Abstract: This article analyzes the problem of mining association rules from large transactional databases, including its formal model and computational stages, based on two fundamental scientific sources. The first source discusses support and confidence constraints, frequent itemset discovery, evaluation, pruning, and memory

management methods [1]. The second source examines the partitioning of the search space into a lattice and equivalence classes, the vertical TID-list representation, the Eclat, MaxEclat, Clique, and MaxClique algorithms, as well as their experimental comparison with Apriori and Partition [2]. The analysis demonstrates that algorithm selection depends not only on execution time but also on data density, minimum support, the size of the longest frequent itemset, the number of database scans, and main-memory constraints. Vertical data representation and hybrid search can provide significant advantages for dense datasets, whereas a simple level-wise search may be practically sufficient for sparse databases.

Keywords: Association rule, frequent itemset, support, confidence, Apriori, Partition, Eclat, MaxClique, TID-list, lattice, pruning.

Kirish: Assotsiativ qoidalarni izlash tranzaksiyalar ichida birgalikda uchraydigan obyektlar oʻrtasidagi barqaror bogʻlanishlarni aniqlaydi. Klassik misolda savdo cheki bitta tranzaksiya, mahsulotlar esa elementlar sifatida qaraladi. Natijada “A va B xarid qilinganida C ham xarid qilinadi” koʻrinishidagi qoidalar olinadi. Bunday bilim mahsulotlarni joylashtirish, tavsiya berish, katalog tuzish, mijozlarni segmentlash va telekommunikatsiya hodisalarini tahlil qilishda ishlatilishi mumkin [1], [15], [16].

Muammoning asosiy qiyinligi qoidaning oʻzini yozishda emas, barcha muhim elementlar kombinatsiyalarini topishdadir. m ta element uchun nazariy qidiruv fazosi 2^m ta toʻplamgacha kengayadi. Shu sababli toʻliq sanash katta bazalarda amaliy emas. Samarali algoritmlar nomzodlarni erta rad etishi, bazani kamroq skanerlashi va oraliq tuzilmalarni tezkor xotirada boshqarishi kerak [2], [9], [10], [11], [12].

Assotsiativ qoidalarni izlashning umumiy jarayoni



1-rasm. Assotsiativ qoidalarni izlash bosqichlari. Muallif ishlanmasi, [1] va [2] asosida.

$I = \{i_1, i_2, \dots, i_m\}$ elementlar to‘plami, D esa tranzaksiyalar bazasi bo‘lsin. Har bir $t \in D$ tranzaksiya I ning qism to‘plamidir. $X \subseteq I$ to‘plamning support soni X mavjud bo‘lgan tranzaksiyalar soniga, nisbiy support esa shu sonning $|D|$ ga nisbatiga teng. X tez-tez uchraydigan to‘plam deb hisoblanishi uchun uning support qiymati foydalanuvchi belgilagan minsup chegarasidan kam bo‘lmasligi kerak [1], [2].

$X \rightarrow Y$ qoida uchun $X \cap Y = \emptyset$ sharti bajariladi. Qoidaning support qiymati $X \cup Y$ birgalikda uchragan tranzaksiyalar ulushidir. Confidence esa X mavjud bo‘lgan holatlarda Y ham uchrash ehtimolini baholaydi: $\text{conf}(X \rightarrow Y) = \text{supp}(X \cup Y) / \text{supp}(X)$. Support qoidaning qamrovi va statistik ahamiyatini, confidence esa shartli kuchini ifodalaydi [1].

Ko‘rsatkich	Formula	Mazmuni	Qo‘llanishi
Support soni	$\sigma(X) = \{t \in D: X \subseteq t\} $	X qatnashgan tranzaksiyalar soni	Tez-tez uchraydigan to‘plamni ajratish
Nisbiy support	$\text{supp}(X) = \sigma(X) / D $	X ning bazadagi ulushi	minsup bilan solishtirish
Confidence	$\text{conf}(X \rightarrow Y) = \text{supp}(X \cup Y) / \text{supp}(X)$	X bo‘lsa Y ning kuzatilish darajasi	minconf bilan solishtirish
Maksimal to‘plam	$X \not\subseteq Z$ bo‘lgan frequent X	Kengroq frequent ustto‘plami yo‘q	Natijani ixchamlashtirish

1-jadval. Assotsiativ qoidalar uchun asosiy tushunchalar [1], [2].

1 Tadqiqot muammosining qo‘yilishi

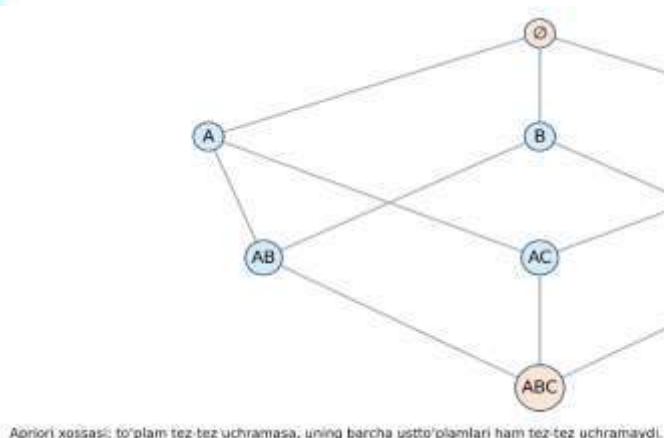
Agrawal, Imielinski va Swami masalani ikki qismga ajratadi. Birinchi bosqichda minsup talabini qanoatlantiruvchi barcha “large itemset”, ya’ni tez-tez uchraydigan to‘plamlar aniqlanadi. Ikkinchi bosqichda topilgan har bir to‘plamdan minconf

talabini bajaradigan qoidalar hosil qilinadi. Ikkinchi bosqich support qiymatlari ma'lum bo'lgach nisbatan sodda; asosiy hisoblash xarajati birinchi bosqichga to'g'ri keladi [1]. Zaki ham tez-tez uchraydigan to'plamlarni sanashni hisoblash va kiritish-chiqarish jihatidan eng og'ir qism sifatida ko'rsatadi [2].

Tadqiqot maqsadi va nazariy asos

Tadqiqotning maqsadi assotsiativ qoidalarni izlashda qo'llanadigan nomzodga asoslangan, bo'limlashga asoslangan va vertikal panjaraviy algoritmlarni yagona mezonlar tizimida qiyoslashdan iborat. Shu maqsadda bazani skanerlash soni, nomzodlar va TID-list kesishmalari miqdori, xotira talabi, minimal supportga sezgirlik hamda uzun frequent to'plamlarni aniqlash qobiliyati asosiy baholash ko'rsatkichlari sifatida olindi [3], [5], [17], [20].

Agar X to'plam tez-tez uchramasa, X ni o'z ichiga olgan har qanday Z ustto'plamning support qiymati ham X nikidan katta bo'la olmaydi. Aksincha, tez-tez uchraydigan to'plamning barcha qism to'plamlari ham tez-tez uchraydi. Mazkur anti-monoton xossa Apriori turidagi algoritmlarda nomzodlar sonini qisqartiradi. Panjara tartibi va grafga asoslangan ajratish qidiruv fazosining murakkabligini izohlashga xizmat qiladi [2], [9], [10],



2-rasm. Uch elementli qidiruv panjarasi va anti-monotonlik xossasi [2].

2 Tadqiqot materiallari va metodlari

Tadqiqot qiyosiy ikkilamchi tahlil dizaynida bajarildi. Material sifatida assotsiativ qoidalar masalasining dastlabki formal modeli va nomzodlarni baholash usullarini bergan Agrawal, Imielinski va Swami maqolasi [1], shuningdek panjara, vertikal TID-list va masshtablanuvchi algoritmlarni tajribada solishtirgan Zaki maqolasi [2] tanlandi. Har bir manbadan muammo ta’rifi, algoritmik bosqichlar, kesish mexanizmlari, ma’lumot ko’rinishi va e’lon qilingan tajriba natijalari ajratildi. Natijalar bir xil mezonlar bo’yicha jadvalga keltirilib, sonli dalillar faqat manbada aniq ko’rsatilgan holatlarda ishlatildi.

Birinchi manbada taklif qilingan umumiy algoritm bazani bir necha marta ko’rib chiqadi. Har bir o’tishda frontier deb ataladigan joriy chegaradagi to’plamlar kengaytiriladi, ularning nomzod ustto’plamlari hosil qilinadi va support qiymati hisoblanadi. Support minsup dan past bo’lsa nomzod rad etiladi; aks holda u keyingi o’tish uchun frontier tarkibiga kiradi. Frontier bo’shaganda jarayon tugaydi [1].

Manbalarni tanlash va qiyosiy tahlil usuli

Manbalarni tanlash mezonlari uchta talabdan iborat bo’ldi: assotsiativ qoidalar vazifasining formal ta’rifi, algoritmning qayta tiklanadigan tavsifi va tajribaviy baholashning mavjudligi. [1] nazariy va algoritmik asosni, [2] esa Apriori, Partition, Eclat, MaxEclat, Clique va MaxClique usullarining kengaytirilgan qiyosini ta’minlaydi. Dinamik nomzodlarni sanash yondashuvi [7] ham nomzodlarni qaysi bosqichda o’lchash masalasini kengaytiradi. Tahlilda manbalardagi natijalar saqlandi; kichik tranzaksion misol metodni tushuntirish uchun muallif tomonidan tuzildi.

Algoritmik mezonlar va pruning mexanizmlari

Remaining tuple optimization joriy skanerlash nuqtasi, kuzatilgan support va hali o’qilmagan tranzaksiyalar sonidan foydalanadi. Nomzod barcha qolgan tranzaksiyalarda uchragan taqdirda ham minsup ga yeta olmasa, u darhol chiqariladi. Aksincha, nomzod kerakli chegaraga allaqachon yetgan bo’lsa, uning yakuniy holati oldindan tasdiqlanishi mumkin. Pruning function optimization esa bir o’tishda hosil

qilingan cheklovlardan keyingi o'tishlarda nomzodlarni erta rad etish uchun foydalanadi [1].

Usul	Qaror uchun ma'lumot	Asosiy foyda	Ehtiyot jihati
Support bo'yicha kesish	minsup va joriy support	Kam uchraydigan shoxlarni yopadi	Chegara pasaysa nomzodlar ko'payadi
Qolgan tranzaksiyalar optimallashtirishi	Joriy sanoq va qolgan satrlar	Nomzodni skan davomida rad etadi	Skan holatini yuritish talab etiladi
Kesish funksiyasi	Oldingi o'tishda olingan cheklov	Keyingi o'tish ishini kamaytiradi	Qo'shimcha qoidalarini saqlaydi
Sintaktik cheklov	Antecedent yoki consequent talabi	Faqat kerakli qoidalarini qidiradi	Foydalanuvchi maqsadi aniq bo'lishi kerak

2-jadval. Nomzodlarni qisqartirish usullari [1].

Xotira sarfini baholash. Nomzodlar tezkor xotiraga sig'masa, algoritmi ayrim to'plamlarni keyingi o'tishga qoldiradi. O'chiriladigan to'plam shunday tanlanadiki, to'liqlik saqlansin, bir to'plam to'liq qayta o'lchanmasin va jarayonning oldinga siljishi kafolatlanmasin. Ushbu yondashuv katta nomzod fazosini cheklangan xotirada qayta ishlashga imkon beradi [1].

Tahlil mezonlari. Yondashuvning kuchli tomoni nomzodlarni baholash, xavfsiz kesish va bufer boshqaruvini yagona sxemaga birlashtirishidir. Uning zaif tomoni ko'p o'tishli tuzilma va ehtimoliy baholashning ma'lumot taqsimotiga bog'liqligidir. Shunga qaramay, mualliflar yirik chakana savdo ma'lumotlarida baholash aniqligi yuqori, pruning esa o'lchanishi kerak bo'lgan to'plamlarning katta qismini chiqarib tashlaganini bildiradi [1].

Apriori va Partition algoritmlarini tahlil qilish usuli. Apriori darajama-daraja ishlaydi. Dastlab barcha 1-elementli to‘plamlar sanaladi. Tez-tez uchraydigan 1-to‘plamlardan 2-to‘plam nomzodlari, keyin 3-to‘plam nomzodlari hosil qilinadi. Har bir k bosqichida baza qayta skanerlanib C_k nomzodlarning support qiymati hisoblanadi. $k-1$ o‘lchamli frequent to‘plamlar o‘zaro birlashtiriladi, ammo barcha $k-1$ qism to‘plamlari frequent bo‘lmagan nomzodlar oldindan kesiladi [3], [5].

Apriori tushunarli va to‘liq algoritm bo‘lsa-da, uzun frequent to‘plam mavjud bo‘lganida bazani ko‘p marta skanerlaydi. Zaki tajribasidagi T30 — har bir tranzaksiyada o‘rtacha 30 ta element mavjud, I16 — oldindan shakllantirilgan potensial tez-tez uchraydigan elementlar to‘plamining o‘rtacha uzunligi 16 ta, D400K — bazada 400 000 ta tranzaksiya mavjud (T30.I16.D400K) bazasida eng uzun frequent to‘plam 22 elementga yetgan va Apriori 22 marta o‘tish qilgan [2]. Minimal support pasaygani sari nomzodlar hamda tez-tez uchraydigan kombinatsiyalar soni keskin ortadi.

Partition algoritmi. Partition gorizontaal bazani kesishmaydigan qismlarga ajratadi. Har bir qism tezkor xotiraga yuklanadi, lokal frequent to‘plamlar topiladi va ularning birlashmasi global nomzodlar ro‘yxatini beradi. Global frequent to‘plam butun bazada tez-tez uchrasa, u kamida bitta bo‘limda lokal frequent bo‘lishi shart. Shuning uchun Partition odatda bazani ikki marta skanerlaydi [2], [17].

Bo‘limlar sonining ortishi doimo foyda bermaydi. Lokal darajada frequent, global darajada esa kam uchraydigan to‘plamlar ko‘payadi. Zaki tajribalarida o‘nta bo‘limli Partition10 ko‘pincha uch bo‘limli Partition3 dan 1,5–2 marta sekin, T30.I16 ma’lumotida esa 1 foizlik supportda 13 marta sekin bo‘lgan. Partition10 bajaradigan TID-list kesishmalari Partition3 ga nisbatan 4–5 marta ko‘p bo‘lgan [2].

Mezon	Apriori	Partition
Baza shakli	Gorizontaal	Gorizontaal, bo‘lim ichida vertikal

Skanlar soni	Eng uzun frequent to‘plam darajasigacha	Odatda 2
Nomzodlar	Darajama-daraja hosil qilinadi	Lokal frequent to‘plamlar birlashmasi
Asosiy ustunlik	Sodda va anti-monoton kesish aniq	Kiritish-chiqarish soni kam
Asosiy cheklov	Past minsup da skan va nomzod ko‘p	Soxta lokal nomzodlar ko‘payishi
Mos sharoit	Siyrak baza, qisqa naqshlar	Bo‘limlar xotiraga sig‘adigan baza

3-jadval. Apriori va Partitionning qiyosiy xususiyatlari [2].

Soddalashtirilgan psevdokod

F_1 = frequent 1-to‘plamlar; $k = 2$. $F_{(k-1)}$ bo‘sh bo‘lmaguncha C_k nomzodlar birlashtirish va kesish orqali olinadi, baza skanerlanib har bir nomzod sanaladi, so‘ng $F_k = \{c \in C_k \mid \text{supp}(c) \geq \text{minsup}\}$ shakllantiriladi. Yakunda barcha F_k to‘plamlar birlashtiriladi. Bu ifoda Apriori mantiqini beradi; amaliy unumdorlik nomzodlarni saqlash va support hisoblash tuzilmasiga bog‘liq [2].

Vertikal ma’lumot shakli va Eclat. Vertikal ko‘rinishda har bir element bilan u uchraydigan tranzaksiya identifikatorlari ro‘yxati bog‘lanadi. Masalan, A elementi {1,2,4}, B elementi {1,3,4} tranzaksiyalarda uchrasa, AB to‘plamining TID-listi ularning kesishmasi {1,4} bo‘ladi. Demak, support qiymati oddiy ro‘yxat kesishmasining uzunligi orqali olinadi [2].

Gorizontaal ko'rinish

TID	Elementlar
1	A, B, C
2	A, C
3	B, C
4	A, B

Vertikal TID-list ko'rinishi

Element	TID ro'yxati
A	1, 2, 4
B	1, 3, 4
C	1, 2, 3

3-rasm. Gorizontaal bazadan vertikal TID-list shakliga o'tish. Muallif misoli, [2] asosida.

Ekvivalentlik sinflari. Zaki qidiruv panjarasini umumiy prefiksga ega ekvivalentlik sinflariga ajratadi. Masalan, AB, AC va AD to'plamlari A prefiksli bitta sinfga kiradi. Har bir sinf alohida kichik panjara sifatida xotirada qayta ishlanishi mumkin. Bu ajratish muammoni qismlarga bo'ladi, sinflar orasidagi keraksiz kombinatsiyalarni kamaytiradi va ma'lumotlarni lokal qayta ishlashni yaxshilaydi [2].

Eclat algoritmi. Eclat prefiksga asoslangan ekvivalentlik sinflari va pastdan yuqoriga qidiruvdan foydalanadi. Sinfdagi ikki elementning TID-listlari kesishib yangi to'plam hosil qiladi. Natija support chegarasidan o'tsa, yangi to'plam shu prefiks ostida davom ettiriladi. Uzunroq to'plamlar hosil bo'lgani sari TID-listlar qisqaradi; shu sababli keyingi kesishmalar ko'pincha tezroq bajariladi [2].

Eclat murakkab hash-tree tuzilmasini talab qilmaydi va odatda bazani bir necha marta skanerlash bilan cheklanadi. Chiziqli tartiblangan ikki TID-list kesishmasi ketma-ket o'qish orqali bajariladi, bu esa xotira lokalizatsiyasini yaxshilaydi. Zaki tajribalarida Eclat barcha ko'rilgan holatlarda Apriori va Partitiondan tez ishlagan; u Aprioridan odatda bir tartibdan ko'proq, Partition3 dan taxminan 2 marta va Partition10 dan 4 marta ustun bo'lgan [2], [20].

Bosqich	Amal	Natija

1	Frequent 1-elementlarning TID-listlarini tuzish	Boshlang'ich vertikal baza
2	Umumiy prefiks bo'yicha sinflarga ajratish	Mustaqil kichik qidiruv fazolari
3	TID-listlarni kesish	Yangi to'plam va uning supporti
4	minsup bo'yicha kesish	Faqat frequent kengaytmalar
5	Sinfni rekursiv davom ettirish	Barcha frequent to'plamlar

4-jadval. Eclat algoritmining ish ketma-ketligi [2].

Eclatning cheklovi TID-listlarni saqlash xarajati bilan bog'liq. Juda ko'p tranzaksiya va yuqori supportli elementlar mavjud bo'lsa, boshlang'ich ro'yxatlar uzun bo'ladi. Biroq sinflar alohida qayta ishlangani sababli barcha oraliq ro'yxatlarni bir vaqtda saqlash talab qilinmaydi. Zaki keltirgan bir tajribada o'rtacha xotira sarfi baza hajmining taxminan 2 foiziga teng bo'lgan [2].

Pastdan yuqoriga qidiruv 1-elementli to'plamlardan boshlanib uzunroq kombinatsiyalarga o'tadi. Afzalligi shundaki, anti-monoton xossa yordamida har bir bosqich xavfsiz kesiladi va barcha frequent to'plamlar to'g'ridan-to'g'ri sanaladi. Kamchiligi uzun naqsh mavjud bo'lsa, uning juda ko'p qism to'plamlarini ko'rib chiqish zarur bo'ladi [2].

Yuqoridan pastga qidiruv katta to'plamlardan boshlanib minimal infrequent to'plamlarni aniqlashga intiladi. Uzun frequent naqshlar mavjud bazada bu yo'l foydali bo'lishi mumkin [6]. Ammo katta nomzod supportini aniqlash ko'p tomonlama TID-list kesishmasini talab qiladi; zich grafda maksimal klikalar yiriklashsa hisoblash og'irlashadi. Tasodifiy qidiruv usullari ham katta kombinatsiyalar fazosini qisqartirish yo'llaridan biri sifatida o'rganilgan [13].

Gibrid qidiruv avval uzun frequent to'plamlarni imkon qadar erta topadi, so'ng ular orqali ko'plab qism to'plamlarni bevosita sanamasdan natijani to'ldiradi.

MaxEclat prefiksli sinflar, MaxClique esa maksimal klikalarga asoslangan sinflar bilan gibril qidiruvni birlashtiradi. Zaki natijalarida uzunligi 19 yoki 22 bo‘lgan frequent to‘plamlarda aynan gibril usullar xotira cheklovidan o‘ta olgan [2].

Clique yondashuvi frequent 2-to‘plamlardan tuzilgan assotsiatsiya grafidagi maksimal klikalardan qidiruv ostfazorlari sifatida foydalanadi. Maksimal klika elementlari o‘zaro juft frequent bo‘lgani uchun ular ehtimoliy uzun naqshni chegaralaydi. MaxClique ushbu ajratishni gibril qidiruv bilan birlashtiradi. Kichikroq ostpanjaralar nomzodlar sonini kamaytiradi, ammo graf juda zich bo‘lsa klikalar soni, hajmi va o‘zaro qoplanishi ortadi [2].

Strategiya	Boshlanish nuqtasi	Kuchli jihati	Cheklovi
Bottom-up	Qisqa to‘plamlar	To‘liq va sodda kesish	Uzun naqshda ko‘p qism to‘plam
Top-down	Yirik to‘plamlar	Uzun naqshni erta topishi mumkin	Ko‘p tomonlama kesish qimmat
Hybrid	Yirik va kichik chegaralarni uyg‘unlashtiradi	Uzun frequent to‘plamda samarali	Boshqaruv mantiqi murakkabroq
Maximal clique	Frequent juftliklar grafi	Kichik ostpanjaralar yaratadi	Zich grafda klikalar qoplanadi

5-jadval. Panjara bo‘ylab qidiruv strategiyalari [2].

Amaliy tanlovda qidiruv yo‘nalishi ma’lumotlar tuzilishiga mos kelishi kerak. Qisqa va siyrak naqshlarda bottom-up usulining soddaligi afzal. Past minsup, uzun naqsh va zich tranzaksiyalarda vertikal shakl hamda gibril qidiruv ko‘proq foyda beradi. Katta hajmda parallel va taqsimlangan bajarish variantlari hisoblash yukini

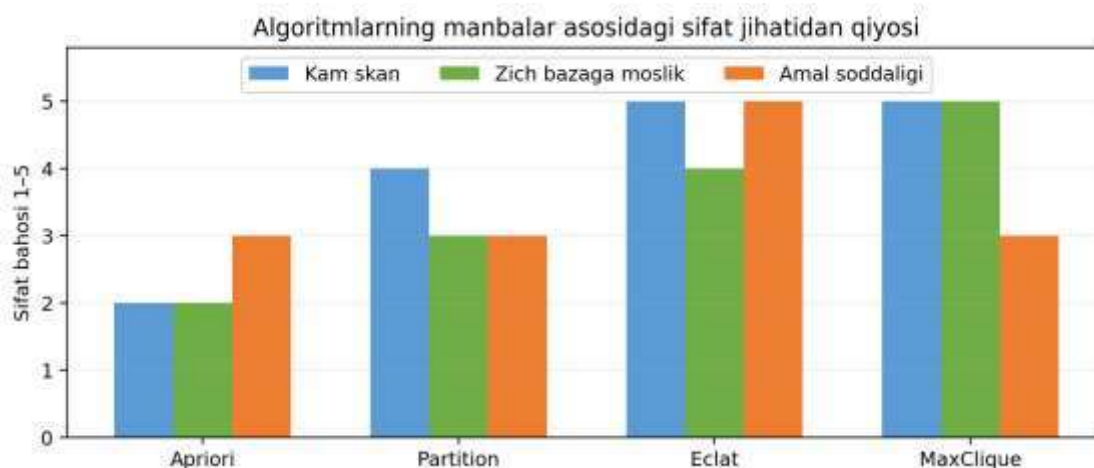
tugunlar o'rtasida taqsimlaydi [4], [8], [14]. Clique turidagi algoritmni qo'llashdan oldin frequent juftliklar grafi zichligini baholash lozim [2].

3 Tadqiqot natijalari

Algoritm unumdorligi bitta mezon bilan aniqlanmaydi. Baza skanlari disk kiritish-chiqarish xarajatini, nomzodlar soni protsessor yukini, TID-listlar esa xotira sarfini belgilaydi. Minimal support pasayganda frequent to'plamlar soni va uzunligi ortadi. Shunday sharoitda ko'p o'tishli Apriori sekinlashadi, vertikal algoritmlar esa qisqaruvchi TID-listlardan foyda ko'radi [2], [18], [19].

Algoritm	Ma'lumot shakli	Ajratish	Qidiruv	Skanlar	Eng mos holat
Apriori	Gorizontal	Darajalar	Bottom-up	Ko'p	Siyrak, qisqa naqsh
Partition	Gorizontal/vertikal	Bazani bo'limlash	Bottom-up	2	Bo'limlar xotiraga sig'sa
Eclat	Vertikal	Prefiks sinflari	Bottom-up	Kam	Zich baza, past minsup
MaxEclat	Vertikal	Prefiks sinflari	Hybrid	Kam	Uzun frequent to'plam
Clique	Vertikal	Maksimal klika	Bottom-up	Kam	O'rtacha zich graf
MaxClique	Vertikal	Maksimal klika	Hybrid	Kam	Uzun naqsh, boshqariladigan klika

6-jadval. Asosiy algoritmlarning umumlashtirilgan qiyosi [2].



4-rasm. Algoritmlarning sifat ko'rsatkichlari bo'yicha qiyosi. 1–5 ballar muallifning [1] va [2] dagi tavsiflar asosidagi analitik bahosidir; ular bajarilish vaqti emas.

3 Algoritmlarning tajribaviy natijalari

Zaki sintetik bazalarda $N = 1000$ element va $L = 2000$ maksimal namunadan foydalangan. T30.I16.D400K bazasida 0,5 foiz supportda 13 480 771 ta frequent to'plam, eng uzun to'plam esa 22 elementdan iborat bo'lgan. MaxClique barcha 2^{22} qism to'plamlarni sanash o'rniga 197 601 ta kesishma bajargan. 0,75 foiz supportda MaxClique 69 soniya, Apriori 22 963 soniya sarflagan, ya'ni farq 332 martaga yetgan; Partition10 esa virtual xotira yetishmovchiligiga uchragan [2].

Biroq MaxClique har doim mutlaq g'olib emas. T40.I8 ma'lumotida 0,25 foiz supportda klika asosidagi usullar ishlay olmagan, prefiksga asoslangan Eclat va MaxEclat esa ishlagan. Sabab graf zichlashganda maksimal klikalar soni va ularning o'zaro qoplanishi ortishidir. Shuning uchun “eng tez algoritmi” degan umumiy hukm o'rniga baza xususiyatiga mos algoritmi tanlanishi kerak [2].

Natijalarni amaliy tekshirish metodikasi

Har bir yozuv noyob TID va elementlar ro'yxatiga aylantiriladi. Takroriy elementlar, bo'sh tranzaksiyalar va noto'g'ri kodlar tozalanadi. Elementlarning juda noyob qiymatlari zarur bo'lmasa oldindan chiqarilishi mumkin, chunki ular

belgilangan minsup ga yeta olmaydi. Shaxsga oid ma'lumotlarda identifikatorlar psevdonimlashtirilishi kerak.

Chegaralarni belgilashda minsup juda yuqori bo'lsa muhim, ammo kamroq uchraydigan naqshlar yo'qoladi; juda past bo'lsa natijalar soni va hisoblash xarajati keskin ortadi. minconf yuqori qo'yilganda kuchli qoidalar qoladi, lekin antecedenti juda tez-tez uchraydigan qoidalar noto'g'ri talqin qilinishi mumkin. Katta bazada dastlabki namuna asosida chegaralarni baholash mumkin, biroq namunadagi naqshlarning butun bazada tasdiqlanishi tekshirilishi lozim [18], [19]. Shu sababli bir nechta support oralig'ida tajriba o'tkazish maqsadga muvofiq [1], [2].

Algoritm tanlash natijalari

Kuzatilgan sharoit	Tavsiya etiladigan usul	Asos
Siyrak tranzaksiya, qisqa naqsh, o'rtacha minsup	Apriori	Sodda amalga oshirish va samarali anti-monoton kesish
Baza katta, bo'lim tezkor xotiraga sig'adi	Partition	Odatda ikki skan bilan I/O ni kamaytiradi
Zich baza yoki past minsup	Eclat	TID-list kesishmasi va kam skan
Uzun frequent to'plam kutiladi	MaxEclat yoki MaxClique	Gibrid qidiruv uzun naqshni erta aniqlaydi
Frequent juftliklar grafi juda zich	Eclat yoki MaxEclat	Klikalarning ortiqcha qoplanishidan qochadi

7-jadval. Ma'lumot xususiyatiga ko'ra algoritm tanlash [1] va [2] asosida.

Natijalarni tekshirish

Topilgan qoida support va confidence chegarasidan o'tgani bilan avtomatik ravishda foydali hisoblanmaydi. U biznes yoki ilmiy mazmunga ega bo'lishi,

tasodifiy bog‘lanish bo‘lmasligi va mustaqil vaqt oralig‘ida qayta kuzatilishi kerak. Qoidalar soni katta bo‘lsa maksimal frequent to‘plamlar, yopiq to‘plamlar yoki maqsad elementiga qo‘yilgan sintaktik cheklovlar natijani ixchamlashtiradi. Antecedent va consequent tarkibiga cheklov qo‘yish qidiruvni maqsadga yo‘naltiradi [1], [7], [13].

Tajriba jurnalida tranzaksiyalar soni, o‘rtacha tranzaksiya uzunligi, noyob elementlar soni, minsup, minconf, frequent to‘plamlar soni, eng uzun to‘plam, bajarilish vaqti, baza skanlari, kesishmalar hamda maksimal xotira sarfi qayd etiladi. Ushbu ko‘rsatkichlar algoritmlarni adolatli taqqoslash va chegaralarni qayta tanlash uchun yetarli amaliy asos beradi.

Takrorlanuvchanlik uchun elementlarni saralash tartibi, bo‘limlar soni, apparat konfiguratsiyasi va dastur versiyasi ham ko‘rsatiladi. Ayniqsa Partitionda bo‘limlar soni, Clique oilasida graf zichligi, Eclatda esa TID-listlarni kodlash usuli natijaga ta’sir qiladi [2].

Muhokama

Quyidagi kichik misol algoritmlarning umumiy ishlashini tushuntiradi. $D = \{T1:\{A,B,C\}, T2:\{A,C\}, T3:\{B,C\}, T4:\{A,B\}, T5:\{A,B,C\}\}$ bo‘lsin. minsup = 60 foiz yoki kamida 3 tranzaksiya, minconf = 70 foiz deb olinadi. A 4 marta, B 4 marta, C 4 marta uchraydi; AB, AC va BC juftliklarning har biri 3 marta uchraydi. ABC esa 2 marta kuzatilgani uchun frequent emas.

To‘plam	Support soni	Support foizi	Holati
A, B, C	har biri 4	80%	Frequent
AB, AC, BC	har biri 3	60%	Frequent
ABC	2	40%	Infrequent

8-jadval. Namunaviy bazadagi frequent to‘plamlar. Muallif hisob-kitobi.

AB to'plamidan $A \rightarrow B$ va $B \rightarrow A$ qoidalari olinadi. $\text{conf}(A \rightarrow B) = \text{supp}(AB)/\text{supp}(A) = 3/4 = 75$ foiz. Xuddi shuningdek $\text{conf}(B \rightarrow A) = 75$ foiz. Ikkala qoida ham 60 foizlik support va 70 foizlik confidence talabini bajaradi. ABC frequent bo'lmagani uchun undan qoidalar hosil qilinmaydi. Anti-monotonlikka ko'ra ABC ning barcha katta ustto'plamlari mavjud bo'lgan taqdirda ham tekshirilmas edi [1], [2].

Apriori natijasining talqini. Birinchi skanda A, B va C topiladi. Ikkinchi skanda AB, AC va BC supporti hisoblanadi. Uchinchi bosqichda ABC nomzodi hosil qilinadi, ammo support 40 foiz bo'lgani uchun rad etiladi. Natijada algoritm uch darajani ko'rib chiqadi.

Eclat natijasining talqini. Vertikal ro'yxatlar $A:\{1,2,4,5\}$, $B:\{1,3,4,5\}$, $C:\{1,2,3,5\}$ ko'rinishida tuziladi. A va B ro'yxatlari kesishib $AB:\{1,4,5\}$ ni, A va C kesishib $AC:\{1,2,5\}$ ni, B va C kesishib $BC:\{1,3,5\}$ ni beradi. AB va C kesishmasi $ABC:\{1,5\}$ bo'lib, minsup dan past. Natija gorizontol bazani har bir darajada qayta skanerlashsiz olinadi [2].

Murakkablik va natijalarni talqin qilish. Nazariy qidiruv fazosi elementlar soniga nisbatan eksponentsialdir. Amaliy algoritmlar bu fazoni to'liq ko'rmaydi: supportning anti-monotonligi, sintaktik cheklov, bo'limlash, ekvivalentlik sinflari, maksimal klikalar va gibril qidiruv orqali ko'p kombinatsiyalar chiqariladi [1], [2]. Shuning uchun real murakkablik faqat m bilan emas, support taqsimoti va frequent to'plamlar tuzilishi bilan belgilanadi.

Namunadagi 75 foizlik confidence A ning B ga sabab bo'lishini isbotlamaydi. Qoida birgalikda kuzatilish modelidir. Natijani qaror qabul qilishda ishlatishdan oldin vaqt, segment va tashqi omillar bo'yicha tekshirish kerak. Bu farq assotsiativ tahlilning amaliy ishonchliligi uchun muhim.

Umumiy muhokama. Ikki manba assotsiativ qoidalarni izlash rivojining ikki muhim bosqichini ko'rsatadi. Dastlabki model support va confidence orqali masalani aniq formal shaklga keltirdi, tez-tez uchraydigan to'plamlarni topishni asosiy

hisoblash muammosi sifatida ajratdi hamda baholash, pruning va xotira boshqaruvini taklif qildi [1]. Keyingi ish qidiruv fazosining panjaraviy tuzilishidan, vertikal TID-listlardan va mustaqil ostpanjaralardan foydalanib ushbu bosqichni sezilarli tezlashtirdi [2].

Asosiy ilmiy xulosa shuki, samaradorlik nomzod yaratish va support sanashni birgalikda optimallashtirish orqali oshadi. Apriori nomzodlarni anti-monotonlik bilan kamaytiradi, ammo ko'p skan bajaradi. Partition skanlarni kamaytiradi, lekin lokal soxta musbat to'plamlar xarajatini oshirishi mumkin. Eclat vertikal kesishma orqali supportni tez hisoblaydi. MaxEclat va MaxClique uzun frequent to'plamlarni erta topishga qaratilgan gibrid qidiruv bilan eksponentsial qism fazosining katta qismini bevosita sanashdan qochadi [2].

Xulosa

Assotsiativ qoidalarni izlash ikki ketma-ket vazifadan iborat: frequent elementlar to'plamlarini aniqlash va ulardan ishonchli qoidalarni hosil qilish. Support qamrovni, confidence shartli bog'lanish kuchini ifodalaydi. Hisoblashning eng qimmat qismi frequent to'plamlarni topishdir [1].

Tahlil natijasida quyidagi amaliy xulosalar olindi. Siyrak va qisqa naqshli bazada Apriori soddaligi bilan foydali. Disk skanlari asosiy cheklov bo'lib, bo'limlar xotiraga sig'sa Partition ma'qul. Zich ma'lumot va past minsup sharoitida Eclatning vertikal TID-list kesishmalari ustun. Uzun frequent naqshlarda MaxEclat yoki MaxClique gibrid qidiruvi katta tezlik yutug'ini berishi mumkin. Biroq assotsiatsiya grafi haddan tashqari zich bo'lsa, klika asosidagi ajratish samarasi kamayadi [2].

Kelgusi tadqiqotlarda algoritmnı avtomatik tanlash uchun baza zichligi, TID-list uzunligi, frequent juftliklar grafi zichligi va kutilayotgan xotira sarfini birgalikda baholaydigan adaptiv mezon ishlab chiqish maqsadga muvofiq. Shuningdek, natijalarni yopiq va maksimal frequent to'plamlar orqali ixchamlashtirish hamda qoidalarni sohaviy cheklovlar bilan tekshirish katta amaliy qiymatga ega.

Foydalanilgan adabiyotlar

- [1] Agrawal R., Imielinski T., Swami A. Mining Association Rules Between Sets of Items in Large Databases. ACM SIGMOD, 1993, pp. 207–216.
- [2] Zaki M. J. Scalable Algorithms for Association Mining. IEEE TKDE, 12(3), 2000, pp. 372–390.
- [3] Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo A. I. Fast Discovery of Association Rules. Advances in Knowledge Discovery and Data Mining, 1996, pp. 307–328.
- [4] Agrawal R., Shafer J. C. Parallel Mining of Association Rules. IEEE TKDE, 8(6), 1996, pp. 962–969.
- [5] Agrawal R., Srikant R. Fast Algorithms for Mining Association Rules. VLDB, 1994, pp. 487–499.
- [6] Bayardo R. J. Efficiently Mining Long Patterns from Databases. ACM SIGMOD, 1998, pp. 85–93.
- [7] Brin S., Motwani R., Ullman J., Tsur S. Dynamic Itemset Counting and Implication Rules for Market Basket Data. ACM SIGMOD, 1997, pp. 255–264.
- [8] Cheung D., Han J., Ng V., Fu A., Fu Y. A Fast Distributed Algorithm for Mining Association Rules. PDIS, 1996, pp. 31–42.
- [9] Davey B. A., Priestley H. A. Introduction to Lattices and Order. Cambridge University Press, 1990.
- [10] Eppstein D. Arboricity and Bipartite Subgraph Listing Algorithms. Information Processing Letters, 51, 1994, pp. 207–211.
- [11] Garey M. R., Johnson D. S. Computers and Intractability. W. H. Freeman, 1979.
- [12] Gunopulos D., Khardon R., Mannila H., Toivonen H. Data Mining, Hypergraph Transversals, and Machine Learning. ACM PODS, 1997, pp. 209–216.
- [13] Gunopulos D., Mannila H., Saluja S. Discovering All the Most Specific Sentences by Randomized Algorithms. ICDT, 1997, pp. 215–229.
- [14] Han E. H., Karypis G., Kumar V. Scalable Parallel Data Mining for Association Rules. ACM SIGMOD, 1997, pp. 277–288.
- [15] Holsheimer M., Kersten M., Mannila H., Toivonen H. A Perspective on Databases and Data Mining. KDD, 1995, pp. 150–155.
- [16] Houtsma M., Swami A. Set-Oriented Mining of Association Rules in Relational Databases. ICDE, 1995, pp. 25–33.
- [17] Savasere A., Omiecinski E., Navathe S. An Efficient Algorithm for Mining Association Rules in Large Databases. VLDB, 1995, pp. 432–444.

[18] Toivonen H. Sampling Large Databases for Association Rules. VLDB, 1996, pp. 134–145.

[19] Zaki M. J., Parthasarathy S., Li W., Ogihara M. Evaluation of Sampling for Data Mining of Association Rules. RIDE, 1997, pp. 42–50.

[20] Zaki M. J., Parthasarathy S., Ogihara M., Li W. New Algorithms for Fast Discovery of Association Rules. KDD, 1997, pp. 283–286.

