Statistinis dažnų posekių paieškos algoritmas

Loreta Savulionienė
Vilniaus universiteto
Matematikos ir informatikos instituto
doktorantė
Vilnius University Institute
of Mathematics and Informatics,
Doctoral student
Akademijos g. 4, LT-08663 Vilnius
Tel.: (+370 5) 210 9323; (+370 5) 219 1611
El. paštas: l.savulioniene@eif.viko.lt

Leonidas Sakalauskas
Vilniaus universiteto
Matematikos ir informatikos instituto
profesorius habil. daktaras
Vilnius University Institute
of Mathematics and Informatics,
Professor, Habil. Doctor
Akademijos g. 4, LT-08663 Vilnius
Tel.: (+370 5) 210 9321
El. paštas: sakal@ktl.mii.lt

Šiuolaikinis gyvenimas susijęs su dideliais informacijos bei duomenų kiekiais. Paieška yra viena iš pagrindinių kompiuterio darbo operacijų. Paieškos tikslas – rasti dideliame duomenų kiekyje tam tikrą elementą ar elementų seką arba patvirtinti, kad jos nėra. Pagrindinis duomenų gavybos tikslas – rasti duomenyse prasmę, t. y. ryšius tarp duomenų, jų pasikartojamumą ir pan. Straipsnyje pasiūlytas naujas statistinis dažnų posekių paieškos algoritmas, eksperimentų rezultatai bei išvados. Statistinio dažnų posekių paieškos algoritmo esmė – greitai nustatyti dažnus posekius. Šis algoritmas netikrina viso rinkmenos turinio keletą kartų. Vykdant algoritmą rinkmena peržiūrima vieną kartą pagal pasirinktą tikimybę p. Šis algoritmas yra netikslus, tačiau jo vykdymo laikas daug trumpesnis nei tiksliųjų algoritmų. Statistinis dažnų posekių paieškos algoritmas gali būti taikomas struktūrų paieškos uždaviniui, kai aktualu nustatyti, koks posekis yra dažniausias, tačiau nėra labai svarbu tikslus dažnų posekių skaičius.

Pagrindiniai žodžiai: *posekis, kandidatinė seka, duomenų rinkinys, dažnas elementas, elementų rinkinių generavimas, hash funkcija, pirmos rūšies klaida, antros rūšies klaida, pasikliautinumo intervalas.*

Įvadas

Kiekviena veikla šiandien susijusi su dideliais informacijos ir duomenų kiekiais. Paieška duomenyse – svarbiausia operacija. Paieškos tikslas – rasti dideliame duomenų kiekyje tam tikrą seką (elementą) arba patvirtinti, kad jo nėra. Duomenų bazės jau tapo terabaitinėmis, todėl duomenų paieška, analizė, greitas sprendimų priėmi-

mas tampa vis sudėtingesnis. Tarp didelių informacijos kiekių slepiasi ir svarbi, ir niekinė informacija. Šioms problemoms spręsti naudojama duomenų gavyba. Duomenų gavyba – tai duomenų apdorojimas naudojant sudėtingas duomenų paieškos galimybes ir statistinius algoritmus pasikartojantiems šablonams ir koreliacijoms rasti didelėse duomenų bazėse. Duomenų gavyba apibūdinama kaip naujų prasmių duomenyse aptikimo, nustatymo, atradimo būdas. Ši technologija taikoma versle, medicinoje ir kitose srityse, kuriose reikia apdoroti didelius informacijos kiekius ir aptikti duomenų tarpusavio ryšius, t. y. iš didelio duomenų kiekio gauti naujas žinias. Šioms problemoms spręsti naudojami žinomi algoritmai, t. y. *Apriori*, GSP, rekursinis ir kt. Šie algoritmai, vykdydami dažnų posekių paiešką daug kartų perrenka duomenų bazę, todėl esant duomenų gausai tampa neefektyvūs, nors kompiuterinė technika pasiekusi didelį skaičiavimų greitį.

Apriori algoritmas buvo pateiktas 1994 m. ir yra vienas iš populiarių posekių paieškos algoritmų. Šis algoritmas yra interaktyvus, skaičiuoja tam tikro dydžio rinkinius pereidamas per duomenų bazę.

Sekų paieškos algoritmai pirmiausia buvo nagrinėjami Rakesho Agrawalo ir Ramakrishnano Srikanto darbuose. Šiuose darbuose buvo išanalizuoti pagrindiniai algoritmai – klasikinis *Apriori* ir jo patobulinta versija GSP. Šių algoritmų pagrindinė idėja – dažnos sekos ieškomos eliminuojant nedažnus posekius iš galimos dažnos sekos. GSP (angl. *Generalized Sequential Pattern mining algorithm*) buvo pateiktas 1995 m. (Srikant, Agrawal, 1995). Šio algoritmo pagrindinis tikslas – nustatyti, kokios sekos yra nedažnos, ir jų toliau netikrinti.

Attila Gyenesei, Jukka Teuhola savo darbuose pristato dažnų sekų radimo algoritmą, kuriame kandidatinių sekų generavimas remiasi dažnų elementų tikimybiniais įverčiais. Algoritmas PIE (angl. *Probabilistic Iterative Expansion*) pirmiausia randa visus dažnus elementus, o paskui pagal pirmojo žingsnio rezultatus generuoja elementų rinkinius. Dažnų elementų rinkinių tikrinimas šiame algoritme atliekamas taip pat, kaip ir *Apriori* algoritme. Kandidatinės sekos saugomos medžio struktūroje, kurioje kiekvienas kelias nuo viršūnės iki mazgo atitinka vieną elementų rinkinį. Šis algoritmas gali būti apibūdinamas kaip generavimų ir tikrinimų algoritmas (Gyenesei, Teuhola, 2003).

Yra keletas efektyvaus asociatyvių taisyklių naudojimo kliūčių: brangumas ir didelės duomenų bazės. Vienas iš problemos sprendimų būdų – asociatyvumo taisykles taikyti didelėms duomenų bazėms. Kad būtų padidintas paieškos greitis, siūloma asociatyvumo taisyklių paieškos algoritmus taikyti ne visai duomenų bazei, bet atsitiktinei jos imčiai (Cai-Yan, Xie-Ping, 2005).

Iškyla problema, kaip efektyviai apibrėžti ir įvertinti algoritmo darbo rezultatų klaidų tikimybę? Taip pat labai svarbu efektyviai įvertinti, ar pasirinkta imtis yra pakankama?

Daugelis „duomenų apdorojimo – intervalo parinkimo“ strategijų remiasi MRA (angl. *Multi Resolution Analysis*) bei Shannono pasirinkimo teoremomis. Teorinė analizė ir empiriniai tyrimai parodė, kad atliekant paiešką ne visoje didelėje duomenų bazėje, o atsitiktinėje duomenų bazės imtyje sumažinamas algoritmo veikimo greitis, tačiau prarandamas tikslumas.

Cai-Yan Jia ir Xie-Ping Gao savo darbuose aprašo efektyvų ir produktyvų algo-

ritmą, kuris sukurtas naudojant PAC (angl. *Probaly Approximate Correct*) mokymo teoriją.

Apriori algoritmas

Susietumo taisyklių paieškos algoritmų pagrindas – dažnų duomenų rinkinių analizė. Pirmiausia ieškoma dažnų elementų, o paskui iš šių elementų generuojamos kandidatinės sekos. Norint sutrumpinti susietumo taisyklių paiešką naudojama apriorišKumo savybė, t. y. jeigu rinkinys Z yra nedažnas, tai šio rinkinio papildymas bet kokiu nauju elementu A šio rinkinio Z nepadaro dažno.

Jeigu Z nedažnas, tai Z+A taip pat nedažnas.

Šiuolaikinės duomenų bazės saugo didelius duomenų kiekius, todėl susietumo taisyklėms aptikti reikalingi efektyvūs laiko atžvilgiu algoritmai. Vienas iš tokių yra *Apriori* algoritmas. Pirmuoju *Apriori* algoritmo žingsniu randami dažni vieno elemento rinkiniai. Vykdant šį algoritmo žingsnį pereinama visa duomenų rinkmena ir nustatoma, kiek kartų kiekvienas elementas pasitaiko rinkmenoje, ir toliau apdorojami tik tie elementai, kurie tenkina nustatytą minimalų pasirodymų dažnį.

Kiti algoritmo žingsniai susideda iš dviejų dalių: potencialiai dažnų elementų rinkinių generavimo (jie vadinami kandidatais) ir kandidatinių rinkinių dažnumo nustatymo (Ayres, Flannick, Gehrke, Yiu, 2002).

Apriori algoritmas generuoja kito žingsnio elementų kandidatinius rinkinius tik iš rastų dažnų rinkinių prieš tai atliktame žingsnyje. Pagrindinė intuicija yra ta, kad bet kuris dažnas elementų rinkinio poaibis turi būti dažnas rinkinys. Todėl kandidatiniai rinkiniai, sudaryti iš k elementų, generuojami sujungiant dažnus elementų rinkinius, turinčius k–1 elementų, kurie tenkina minimalų pasikartojimų skaičių.

Šiame algoritme svarbi kandidatų generavimo funkcija. Vykdant kandidatų generavimą, nesikreipiama į duomenų rinkmeną. Norint gauti k elementų rinkinius, naudojami (k–1) elementų rinkiniai, kurie buvo dažni ankstesniame žingsnyje. Kiekvienas kandidatas C_k konstruojamas papildant dažną (k–1) elementų rinkinį kitu dažno (k–1) elementų rinkinio elementu.

Atlikus elementų rinkinių generavimą tikrinama, kurie nauji kandidatai tenkina nustatytą minimalų pasirodymų dažnį. Akivaizdu, kad dažnų elementų rinkinių gali būti labai daug, todėl reikalingas efektyvus būdas šiems rinkiniams suskaičiuoti. Pats paprasčiausias būdas – sekos elementus lyginti su kiekvienu generuotu kandidatu. Tačiau toks sprendimas užima daug laiko. Greitesnis ir efektyvesnis sprendimas – tai *hash* medžių (maišos medžių) naudojimas.

Hash medis konstruojamas kiekvieną kartą, kai generuojami kandidatai. Iš pradžių medis turi tik šaknį (šaknis apibrėžiama 1 lygmenyje), kuri tampa medžio lapu ir neturi jokių kandidatų rinkinių. Medžio mazgas gali būti rinkinių sąrašas (mazgo lapas) arba lentelė (vidinis mazgas). Kiekvienas vidinis mazgas, kurio lygmuo d, turi šakas į kitus medžio mazgus, kurių lygmuo d+1. Kai generuojamas naujas kandidatas, jis prijungiamas prie mazgo. Kai mazgo lapų skaičius viršija nurodytą slenkstį, mazgas paverčiamas *hash* lentele (vidiniu mazgu) ir šiam mazgui sukuriami lapai. Visi kandidatai pasiskirsto mazguose pagal įeinančių į rinkinį elementų *hash* reikšmę. Kiekvienas naujas kandidatas generuojamas vidiniame mazge, o saugo-

mas mazgo lape. Taip sukuriamas elementų kandidatinių sekų medis. Naudojant šį medį nesunku rasti kiekvieno kandidato pasikartojimų skaičių. Pradedama nuo šaknies ir randami visi kandidatai, kurie sutampa su transakcijos T_i elementais, t. y. $C_k \cap T_i = C_k$. Pirmajame lygmenyje, t. y. medžio šaknyje, *hash* funkcija taikoma kiekvienam transakcijos elementui. Antrajame lygmenyje *hash* funkcija taikoma antrajam elementui ir t. t., k-ajame lygmenyje *hash* funkcija taikoma k elementui. Tai atliekama tol, kol pasiekiamas medžio lapas. Kai kiekviena transakcija „pereina" per *hash* medį, tikrinama, ar gauta reikšmė tenkina nustatytą minimalų pasirodymų dažnį. Kandidatai, kurie tenkina nustatytą minimalų pasirodymų dažnį, priskiriami dažnoms sekoms (Ayres, Flannick, Gehrke, Yiu, 2002).

Pavyzdys. Turima simbolių seka AAB CAABCBABCABABCBACBCBABCA CAB. Sakykime, kad seka yra dažna tada ir tik tada, kai pasirodo ne mažiau kaip keturis kartus.

Pirmojo žingsnio metu skaičiuojamas sekos elementų pasirodymų dažnis, kuris pateikiamas 1 lentelėje.

1 lentelė. ***Pirmojo žingsnio elementų pasirodymų dažnis***

Seka	Pasirodymo dažnis
A	11
B	11
C	8

Visų elementų pasirodymo dažnis yra ne mažiau nei 4, vadinasi, kitame žingsnyje konstruojamos kandidatinės sekos iš visų pirmojo žingsnio elementų, kurie pateikti 2 lentelėje.

2 lentelė. ***Antrojo žingsnio elementų pasirodymų dažnis***

Seka	Pasirodymo dažnis
AB	7
AC	2
BC	6

Ne visų elementų pasirodymo dažnis yra ne mažiau nei 4, vadinasi, kitame žingsnyje konstruojamos kandidatinės sekos iš tų antrojo žingsnio elementų, kurių pasirodymo dažnis tenkino sąlygą (AB, BC). Šie elementai pateikti 3 lentelėje.

3 lentelė. ***Trečiojo žingsnio elementų pasirodymų dažnis***

Seka	Pasirodymo dažnis
ABC	5

Taigi, dažna seka yra seka ABC.

GSP (Generate Sequence Pattern) algoritmas

Šio algoritmo pagrindinis uždavinys – nustatyti, kokios sekos yra tikrai nedažnos, ir jų toliau netikrinti.

Tarkime, yra aibė $L=\{i_1, i_2, \ldots, i_n\}$, sudaryta iš n elementų. Nagrinėjama duomenų bazė yra Q, ji sudaryta iš įvairių aibės L elementų derinių. Reikia rasti dažnas sekas. Pirmiausia tikrinamos pirmojo lygmens sekos. Tokių sekų yra n: $(i_1, i_2, \ldots, i_n)$. Nustačius jų dažnius, pereinama tikrinti antrojo lygmens sekas. Jų jau bus n^2: $(i_1i_1, i_1i_2, \ldots i_1i_n, i_2i_1, \ldots i_2i_n, \ldots, i_ni_1, \ldots i_ni_n)$. Tačiau dabar tikrinamos ne visos sekos. Ką tikrinti, sprendžiama pagal prieš tai buvusį lygmenį. Jeigu į antrojo lygmens seką įeina nedažna pirmojo lygmens seka, tai antrojo lygmens seka irgi yra nedažna ir ją galima atmesti toliau netikrinant.

Taip pereinama prie kito lygmens, kuris buvo sukurtas iš prieš tai buvusio antrojo lygmens. Trečiajame lygmenyje bus n^3 derinių, bet vėl tikrinamos ne visos sekos, o tik tos, kuriose nėra nedažnų antrojo lygmens posekių. Vadinasi, bus tikrinami ne visi deriniai, bet tik tie, kurie yra prieš tai buvusio lygmens dažnų sekų posekiai (Toivonen, 1996).

Pavyzdys. Turima simbolių seka AAB CAABCBABCABABCBACBCBABCA CAB. Sakykime, kad seka yra dažna tada ir tik tada, kai pasirodo ne mažiau kaip 4 kartus.

Pirmajame žingsnyje skaičiuojamas sekos elementų pasirodymų dažnis, kuris pateikiamas 4 lentelėje.

4 lentelė. ***Pirmojo žingsnio elementų pasirodymų dažnis***

Seka	Pasirodymo dažnis
A	11
B	11
C	8

Antrajame žingsnyje generuojamos kandidatinės sekos ir tikrinamas jų pasirodymo dažnis. Jis pateikiamas 5 lentelėje.

5 lentelė. ***Antrojo žingsnio elementų pasirodymų dažnis***

Seka	Pasirodymo dažnis
AA	2
AB	7
AC	2
BA	4
BB	0
BC	6
CA	4
CB	4
CC	0

Trečiajame žingsnyje generuojamos kandidatinės sekos tik iš dažnų antrojo žingsnio sekų. Šio žingsnio dažnos sekos pateikiamos 6 lentelėje.

6 lentelė. ***Antrojo žingsnio dažnos sekos***

Seka	Pasirodymo dažnis
AB	7
BA	4
BC	6
CA	4
CB	4

Trečiojo žingsnio kandidatinės sekos ir jų pasirodymo dažnis pateikiamas 7 lentelėje.

7 lentelė. ***Trečiojo žingsnio elementų pasirodymų dažnis***

Seka	Pasirodymo dažnis
ABA	1
ABB	0
ABC	5
BAA	0
BAB	3
BAC	1
BCA	3
BCB	3
BCC	0
CAA	1
CAB	2
CAC	1
CBA	3
CBB	0
CBC	1

Sekos, kurios tenkina sąlygą, t. y. pasirodymo dažnis ne mažiau nei 4, pateikiamos 8 lentelėje.

8 lentelė. ***Trečiojo žingsnio dažnos sekos***

Seka	Pasirodymo dažnis
ABC	5

Taigi, dažna seka yra seka ABC.

Rekursinis algoritmas

Rekursija – viena iš pirmųjų matematikos ir informatikos sąvokų.

Rekursinis algoritmas generuoja naujas kandidatines sekas ne „platyn", bet „gilyn". Pirmajame žingsnyje imama tuščia seka ir iš jos generuojamos kandidatinės sekos $\{i_1, i_2, ..., i_n\}$. Iš šių sekų kiekvienos sekos generuojamos antrojo lygmens sekos $\{i_1i_1, i_1i_2, ..., i_1i_n, i_2i_1, i_2i_2, ..., i_2i_n, ..., i_ni_1, i_ni_2, ..., i_ni_n\}$. Sekos generuojamos „į gylį". Nustatomas kiekvienos generuotos sekos dažnumas. Jei seka yra nedažna lygyje n, tai iš jos daugiau negeneruojama n+1 lygio kandidatinių sekų (Juozapavičius, 2007).

Pavyzdys. Turima simbolių seka AAB CAABCBABCABABCBACBCBABCA CAB. Sakykime, kad seka yra dažna tada ir tik tada, kai pasirodo ne mažiau kaip keturis kartus. Sugeneruotos sekos ir jų dažniai pateikiami 9 lentelėje.

9 lentelė. ***Sekos ir jų dažniai***

Eil. Nr.	Seka	Pasirodymo dažnis	Pastaba
1.	A	11	Dažna seka
2.	AA	2	*Nedažna seka*
3.	AB	7	Dažna seka
4.	AC	2	*Nedažna seka*
5.	ABA	1	*Nedažna seka*
6.	ABB	0	*Nedažna seka*
7.	ABC	5	Dažna seka
8.	ABCA	3	*Nedažna seka*
9.	ABCB	2	*Nedažna seka*
10.	ABCC	0	*Nedažna seka*
11.	B	11	Dažna seka
12.	BA	4	Dažna seka
13.	BB	0	*Nedažna seka*
14.	BC	6	Dažna seka
15.	BAA	0	*Nedažna seka*
16.	BAB	2	*Nedažna seka*
17.	BAC	0	*Nedažna seka*
18.	BCA	3	*Nedažna seka*
19.	BCB	3	*Nedažna seka*
20.	BCC	0	*Nedažna seka*
21.	C	8	Dažna seka
22.	CA	4	Dažna seka
23.	CB	4	Dažna seka
24.	CC	0	*Nedažna seka*
25.	CAA	1	*Nedažna seka*
26.	CAB	2	*Nedažna seka*
27.	CAC	1	*Nedažna seka*
28.	CBA	3	*Nedažna seka*
29.	CBB	0	*Nedažna seka*
30.	CBC	1	*Nedažna seka*

Taigi, dažnos sekos yra A, AB, ABC, B, BA, BC, C, CA, CB.

Algoritmų palyginimas

Eksperimentui buvo sugeneruota 100 failų.

Failai buvo generuojami pagal charakteristikas, kurios pateikiamos 10 lentelėje.

10 lentelė. ***Failų generavimo charakteristikos***

Paslėptas fragmentas	SIENA
Simbolių skaičius faile	100000
Skirtingų simbolių skaičius faile	5 (S, I, E, N, A)
Paslėpto fragmento tikimybė faile	0,2

Šie failai buvo apdoroti šiais algoritmais:

1. *Apriori* algoritmu, kai slenkstis lygus 50, 100, 200;
2. GSP algoritmu, kai slenkstis lygus 50, 100, 200;
3. Rekursiniu algoritmu, kai slenkstis lygus 50, 100, 200.

Rabino Karpo algoritmas pritaikytas tiksliam paslėpto fragmento skaičiui faile nustatyti.

Apriori, GSP ir rekursinis algoritmas skirti duomenų sekoms apdoroti, norint rasti dažnas sekas, kurios tenkina nustatytą minimalų pasirodymų slenkstį. *Apriori* ir GSP – tai algoritmai, kurie vykdo paiešką „platyn“, o rekursinis algoritmas vykdo paiešką „gilyn“. Šiuose algoritmuose skiriasi kandidatinių sekų generavimas ir jų pasikartojimo skaičiaus nustatymas, taip pat algoritmų vykdymo laikas bei rastų dažnų sekų skaičius.

Algoritmų vykdymo vidutinė trukmė pateikiama 11 lentelėje.

11 lentelė. ***Algoritmų vykdymo vidutinė trukmė***

Eil. Nr.	Algoritmo pavadinimas	Vidutinė vykdymo trukmė
1.	*Apriori*	00:00:12:541
2.	GSP	00:00:12.086
3.	Rekursinis	00:15:31.687

Pagal apdorojimo trukmę efektyviausi yra statistinis, *Apriori* ir GSP algoritmai, o ilgiausiai sekų paieška užtrunka naudojant rekursinį algoritmą.

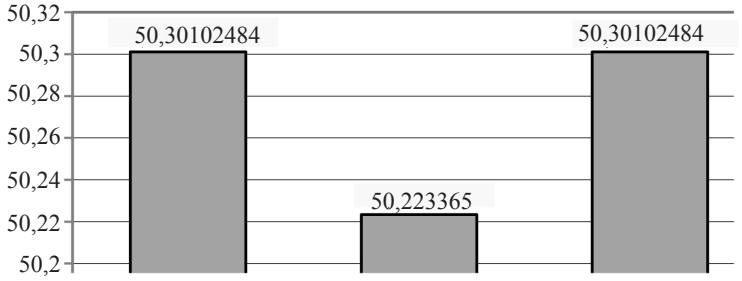
Apdorojus failus, įvertintas vidutinis standartinis nuokrypis nuo tiksliosios reikšmės, kuri buvo nustatyta Rabino Karpo algoritmu. *Apriori*, GSP ir rekursinio algoritmo standartiniai nuokrypiai apytiksliai vienodi (žr. 1 pav.).

Apytiksliai dažnų sekų paieškos algoritmai

Vienas iš apytikslių tikimybinių dažnų posekių paieškos algoritmų yra ProMFS (Tumasonis, Dzemyda, 2004). Šis tikimybinis dažnų sekų nustatymo algoritmas remiasi statistinėmis pagrindinės sekos charakteristikomis, t. y. elemento pasirodymo sekoje tikimybe; tikimybe, kad vienas elementas eis po kito bei atstumo tarp dviejų elementų pagrindinėje sekoje vidurkiu.

ProMFS algoritmas, remdamasis tikimybinėmis charakteristikomis, kurios apibūdina elementų pozicijas pagrindinėje sekoje, generuoja naują daug trumpesnę modelinę seka, kuri analizuojama GSP algoritmu bei nustato dažnus posekius naujojoje sekoje ir daro išvadas apie dažnus posekius pradinėje duomenų sekoje.

Tikimybinis algoritmas (Pragarauskaitė, Dzemyda, 2009) analizuoja ne visą pradinę duomenų seką, o daug trumpesnę jos atsitiktinę imtį. Generuojama atsitiktinio dydžio vienodų tikimybių seka. Ieškomi pirmojo lygio dažni posekiai, antrojo lygio atsitiktinė imtis elementų poroms $a_i a_j$ ir t. t. Tokia imtis yra sudaryta grąžintiniu ėmimu, nes kai kurie posekiai gali pasikartoti. Negrąžintiniu ėmimu sudaryta atsitiktinė imtis formuojama iš pasikartojančių



1 pav. **Apriori,** ***GSP ir rekursinio algoritmo standartiniai nuokrypiai***

posekių pašalinant visus pasikartojančius elementus bei papildomai generuojant naujus posekius, kol bus gautas nesikartojančių posekių rinkinys. GSP algoritmu nustatomi dažni posekiai, kurie skirstomi į tris grupes: dažni posekiai, reti posekiai, tarpiniai posekiai.

Statistinis dažnų posekių paieškos algoritmas

Statistinio algoritmo esmė – greitai nustatyti dažnus posekius didelėse duomenų bazėse. Dažniausiai aktualu išsiaiškinti, koks posekis yra dažnas, o ne tikslų dažnų posekių skaičių. Dažnų posekių paieška aktuali daugelyje veiklos sričių, t. y. tiek versle, tiek pramonėje, medicinoje ir t. t. Statistinis algoritmas tinkamas pirkinių krepšelio analizės (angl. *market basket analysis*), aptarnavimo kokybės, genetikos uždaviniams spręsti ir pan.

Vykdant statistinį algoritmą nereikia tikrinti rinkmenos turinio keletą kartų. Duomenų bazė skenuojama vieną kartą, peržiūrint atsitiktinai pasirinktus atsitiktinio ilgio fragmentus. Šis algoritmas leidžia suderinti du svarbius kriterijus, t. y. laiką ir tikslumą, atitinkamai parenkant parametrų p ir q reikšmes. Beveik neįmanoma sukurti metodo, kuris veiktų visada geriausiai pasitaikius pačiam nepalankiausiam, pačiam palankiausiam arba vidutiniškam atvejams, todėl buvo siekiama sukurti algoritmą, kuris būtų vidutiniškai geriausias.

Tariama, kad atstumas tarp dviejų gretimų peržiūrimų fragmentų yra pasiskirstęs pagal geometrinį dėsnį su parametru p, o peržiūrimo fragmento ilgis yra pasiskirstęs taip pat pagal geometrinį dėsnį su parametru q.

Geometrinio skirstinio tikimybės nusakomos formule: $P(X = k) = p(1-p)^{k-1}$, $k = 1, 2, \ldots$. Posekio palaikymas yra lygus jo dažniui tarp visų peržiūrėtų poaibių. Santykinis posekio palaikymas yra lygus jo dažniui tarp visų peržiūrėtų to paties ilgio poaibių.

Poaibis yra dažnas, jeigu jo palaikymas viršija tam tikrą kritinį ilgį α.

Priimant arba atmetant hipotezę H_0, galimos dviejų rūšių klaidos. Jos vadinamos pirmos ir antros rūšies klaidomis. Pirmos rūšies klaida: hipotezė H_0 atmetama, kai ji teisinga. Antros rūšies klaida: hipotezė H_0 priimama, kai ji klaidinga. Statistinis dažnų posekių paieškos algoritmas yra apytikslis, todėl galimos pirmos ir antros rūšies klaidos. Fiksuojamas posekis c_{i1}, c_{i2}, ..., c_{ik}.

Pirmos rūšies klaida, kai posekis yra dažnas, tačiau statistinio algoritmo neaptiktas kaip dažnas.

Antros rūšies klaida, kai posekis yra nedažnas, o statistinio algoritmo priskirtas dažnų posekių aibei.

Tarkime, p_1, p_2 dvi tokios statistikos, kad $P(p_1 < p < p_2) = \alpha$. Intervalas $[p_1; p_2]$ vadinamas parametro p pasikliautinuoju intervalu. Skaičius γ vadinamas pasikliovimo lygmeniu. Statistinio algoritmo tikslumo kriterijus – tai fragmento radimo tikimybės pasikliovimo rėžis.

Pasikliovimo tikimybės γ rėžiai įvertinami pagal šias formules:

$$p_1 = 1 - \mathrm{BetaInv}\left(\frac{1-\gamma}{2}, n-k, k+1\right);$$

$$p_2 = 1 - \mathrm{BetaInv}\left(1-\left(\frac{1-\gamma}{2}\right), n-k+1, k\right).$$

čia: p_1 ir p_2 – pasikliovimo tikimybės rėžiai,

n – visų fragmentų skaičius,

k – fragmento pasirodymų skaičius,

BetaInv – beta skirstinio kvantilis,

γ – pasikliovimo tikimybė.

Poaibis yra dažnas, jeigu jo pasirodymo tikimybės apatinis rėžis viršija kritinį ilgį α.

Pirmojo eksperimento statistinio dažnų posekių paieškos algoritmo rezultatai

Eksperimento metu 100 rinkmenų apdorota statistiniu dažnų posekių paieškos algoritmu po 100 kartų, kai tikimybė p = 0; p = 0,1; p = 0,2; p = 0,3; p = 0,4; p = 0,5; p = 0,6; p = 0,7; p = 0,8; p = 0,9; p = 1. Po eksperimento įvertintas vidutinis vienos rinkmenos apdorojimo laikas, kai tikimybės p = 0; p = 0,1; p = 0,2; p = 0,3; p = 0,4; p = 0,5; p = 0,6; p = 0,7; p = 0,8; p = 0,9; p = 1. Visų bandymų metu dažno posekio ilgis 5 simboliai. Rastas dažnas posekis visų bandymų metu yra SIENA.

Statistinio dažnų posekių paieškos algoritmo trukmė priklauso nuo p reikšmės: kuo p reikšmė didesnė, tuo ilgesnis algoritmo vykdymo laikas (žr. 2 pav.).

Iš grafiko matoma, kad rinkmenų apdorojimo laikas pradeda didėti, kai p > 0,6. Kai p = 1, tai rinkmenos apdorojimo laikas išauga, nes dažnai neatliekamas sekos simbolių paėmimas ir praleidimas.

Eksperimentiškai buvo apskaičiuotos šios charakteristikos:

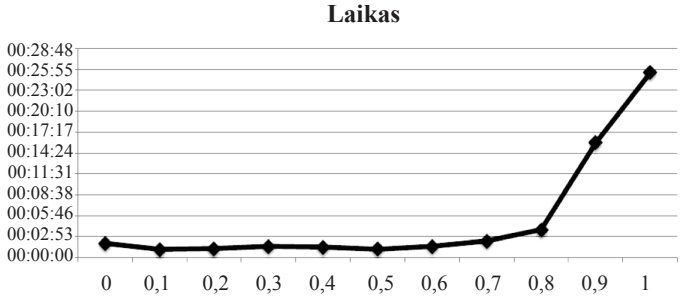
- fragmentų skaičiaus ir imčių skaičiaus santykis. Šis dydis bus žymimas S;
- imčių skaičiaus ir simbolių skaičiaus rinkmenoje santykis. Šis dydis bus žymimas A;
- rastų fragmentų skaičiaus ir paslėptų fragmentų skaičiaus santykis. Šis dydis bus žymimas B;
- paslėptų fragmentų skaičiaus ir simbolių skaičiaus rinkmenoje santykis. Šis dydis bus žymimas C.

Po eksperimento apskaičiuotos kiekvieno badymo charakteristikos S, A, B, C, paskui – visų charakteristikų vidurkiai (12 lentelė).

Iš 12 lentelėje pateiktų duomenų matoma, kad charakteristika C yra pastovus dydis, nes ji nepriklauso nuo rinkmenos apdorojimo rezultatų.

Charakteristikų S, A, B, C tarpusavio sąryšiai:

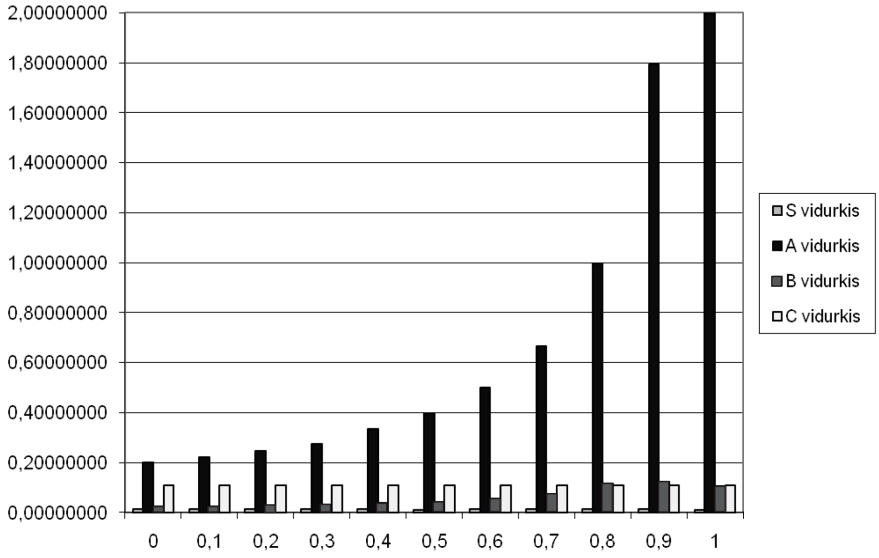
1. Kai $p \in (0; 0,5)$, tai fragmentų kiekio ir imčių skaičius santykis S yra apytiksliai lygus rastų fragmentų skaičiaus ir paslėptų fragmentų skaičiaus santykiui B, t. y. $S \approx B$.



2 pav. ***Statistinio dažnų posekių algoritmo vykdymo trukmės priklausomybė nuo parametro p reikšmės***

12 lentelė. ***Charakteristikų S, A, B, C vidurkiai***

Tikimybė	S vidurkis	A vidurkis	B vidurkis	C vidurkis
0	0,01257019	0,1997446	0,02316206	0,10796
0,1	0,01210599	0,2225090	0,02491133	0,10796
0,2	0,01216679	0,2467121	0,02741206	0,10796
0,3	0,01195124	0,2757958	0,03082554	0,10796
0,4	0,01213383	0,3331287	0,03750797	0,10796
0,5	0,01176993	0,3992666	0,04339021	0,10796
0,6	0,01210091	0,4995012	0,05588621	0,10796
0,7	0,01213194	0,6672925	0,07484155	0,10796
0,8	0,01236089	0,9986325	0,11417181	0,10796
0,9	0,01210648	1,7951489	0,12412773	0,10796
1	0,01146041	1,9998992	0,10611643	0,10796



3 pav. ***Charakteristikų S, A, B, C kitimas***

2. Kai $p \in (0{,}5; 1)$, tai rastų fragmentų skaičiaus ir paslėptų fragmentų skaičiaus santykis B yra apytiksliai lygus paslėptų fragmentų skaičiaus ir simbolių skaičiaus rinkmenoje santykiui C, t. y. $B \approx C$.
3. Imčių skaičiaus ir simbolių skaičiaus rinkmenoje santykis A didėja didėjant tikimybei p.

Antrojo eksperimento statistinio dažnų posekių paieškos algoritmo rezultatai

Eksperimento metu buvo generuotos rinkmenų grupės. Rinkmenų grupės ir jų generavimo charakteristikos pateikiamos 1 priede. Kiekvieną rinkmenų grupę sudaro 100 rinkmenų. Visos 1900 rinkmenų apdorotos statistiniu dažnų posekių paieškos algoritmu.

Rinkmenose dažniausi posekiai ir jų pasikartojimų skaičius nustatytas naudojantis GSP ir Rabino Karpo algoritmu.

13 lentelė. ***Dažniausi posekiai***

Eil. Nr.	Dažniausias posekis	Rinkmenų, kuriose posekis dažniausias, skaičius	Procentai
1.	SIENA	949	49,95 %
2.	SSSSS	951	50,05 %

Eksperimentiškai buvo tiriama:

1. Kiek yra rinkmenų, kuriose posekis SIENA yra dažniausias ir statistinio dažnų posekių paieškos algoritmo aptiktas kaip dažniausias?
2. Kiek yra rinkmenų, kuriose posekis SIENA yra dažniausias ir statistinio dažnų posekių paieškos algoritmo neaptiktas kaip dažniausias, t. y. įvertinta pirmos rūšies klaida?
3. Kiek yra rinkmenų, kuriose posekis SIENA yra nedažnas ir statistinio dažnų posekių paieškos algoritmo nebuvo aptiktas kaip dažniausias, t. y. įvertinta antros rūšies klaida?

14 lentelė. ***Pirmos ir antros rūšies klaidos***

Statistinio algoritmo vykdymas, kai p	Pirmos rūšies klaida	Antros rūšies klaida
0	25 (2,63 %)	58 (6,10 %)
0,1	26 (2,74 %)	61 (6,41 %)
0,2	23 (2,42%)	63 (7,47 %)
0,3	23 (2,42%)	56 (5,89 %)
0,4	24 (2,53 %)	43 (4,52 %)
0,5	23 (2,42%)	61 (6,41 %)
0,6	30 (3,16 %)	49 (5,15 %)
0,7	24 (2,53 %)	59 (6,2 %)
0,8	24 (2,53 %)	53 (5,57 %)
0,9	21 (2,21 %)	48 (5,05 %)
1	27 (2,85 %)	44 (4,63 %)

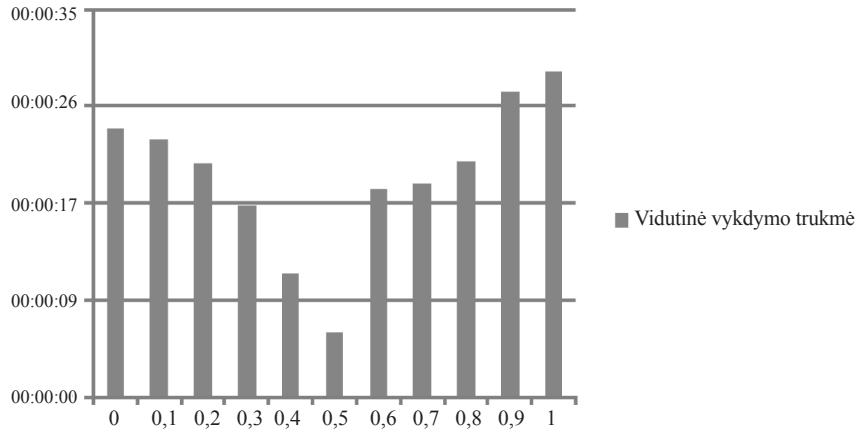
Atlikus eksperimentą pastebėta, kad dažnas posekis aptinkamas nepriklausomai nuo statistinio dažnų posekių paieškos algoritmo p reikšmės. Nuo statistinio dažnų posekių paieškos algoritmo p reikšmės priklauso tik dažno posekio aptikimo skaičius atskirose rinkmenų grupėse. Rinkmenų grupėse, kuriose posekis SIENA yra dažnas, pirmos rūšies klaida 2,59 %. Rinkmenų grupėse, kuriose posekis SIENA nedažnas, antros rūšies klaida 5,76 %.

Statistinis dažnų posekių paieškos algoritmas aptinka dažną seka 95,83 % visų rinkmenų.

Eksperimentiškai buvo tiriama, kokiomis pradinėmis sąlygomis paslėptas fragmentas SIENA tampa nedažnu posekiu. Visos rinkmenos sudarytos iš 100 000 sim-

15 lentelė. ***Paslėpto fragmento tikimybės įtaka fragmento dažniui***

Rinkmenų grupės pavadinimas	Paslėpto fragmento tikimybė p	Ar paslėptas fragmentas yra dažnas posekis?
0.1.*.txt	0,1	TAIP
0.09.*.txt	0,09	TAIP
0.08.*.txt	0,08	TAIP
0.07.*.txt	0,07	TAIP
0.06.*.txt	0,06	TAIP
0.05.*.txt	0,05	TAIP
0.04.*.txt	0,04	TAIP
0.03.*.txt	0,03	TAIP
0.02.*.txt	0,02	TAIP
0.01.*.txt	0,01	TAIP
0.009.*.txt	0,009	NE
0.008.*.txt	0,008	NE
0.007.*.txt	0,007	NE
0.006.*.txt	0,006	NE
0.005.*.txt	0,005	NE
0.004.*.txt	0,004	NE
0.003.*.txt	0,003	NE
0.002.*.txt	0,002	NE
0.001.*.txt	0,001	NE



4 pav. ***Statistinio dažnų posekių paieškos algoritmo vidutinė vykdymo trukmė***

bolių, paslėptas fragmentas SIENA. Paslėptas fragmentas SIENA yra dažnas, kai tikimybė $p \in [0{,}01; 0{,}1]$. Kai $p < 0{,}01$, paslėptas fragmentas tampa nedažnu posekiu.

Atlikus eksperimentą, analizuota klaidos priklausomybė nuo rinkmenos turinio. Ši priklausomybė pateikta 2 priede. Klaidingai gali būti nustatytas dažnas posekis rinkmenose, kuriose paslėptas elementas nėra išskirtinai dažnas, t. y. jo tikimybė priklauso intervalui [0,001; 0,01]. Šiose rinkmenose nėra aiškiai išsiskiriančio dažno posekio, todėl ir atsiranda statistinio dažnų posekių paieškos algoritmo dažno posekio nustatymo klaida.

Statistinio dažnų posekių paieškos algoritmo rinkmenų grupių apdorojimo vidutinė trukmė, kai $p \in [0; 1]$, pateikiama 4 paveiksle.

Statistinio dažnų posekių paieškos algoritmo vidutinis vienos rinkmenos apdorojimo laikas yra 20 sekundžių. Statistinio dažnų posekių paieškos algoritmo rinkmenų apdorojimo laikas mažiausias, kai $p = 0{,}5$.

Po eksperimento buvo įvertinti pasikliovimo tikimybės γ rėžiai. Pasikliovimo tikimybės rėžiai esant skirtingoms p reikšmėms pateikti 16 lentelėje.

16 lentelė. ***Pasikliovimo tikimybės rėžiai***

Parametro p reikšmė	min p_1	max p_2
0	0,95275941	0,99936481
0,1	0,95377541	0,99931803
0,2	0,95309053	0,99933671
0,3	0,95177541	0,99938299
0,4	0,95277458	0,99928147
0,5	0,95211946	0,99932048
0,6	0,95232328	0,99925718
0,7	0,95997471	0,99923696
0,8	0,95339755	0,99921364
0,9	0,95358710	0,99917164
1	0,96340970	0,99914071

Pasikliovimo tikimybės intervalas yra [0,95377541; 0,99938299].

Išvados

Dažniausiai aktualu nustatyti, koks posekis yra dažnas, o ne tikslų dažnų posekių skaičių. Straipsnyje pasiūlytas statistinis dažnų posekių paieškos algoritmas, kuris duomenų bazę peržiūri vieną kartą, atsitiktinai paimdamas atsitiktinio ilgio posekius. Statistinio algoritmo tikslumo kriterijus yra fragmento radimo tikimybės pasikliovimo rėžis. Šis algoritmas yra apytikslis, tačiau leidžia suderinti du svarbius kriterijus, t. y. laiką ir tikslumą, atitinkamai parenkant parametrų p ir q reikšmes. Remdamasis atsitiktinai paimtų posekių analize algoritmas pateikia statistines išvadas apie dažnus posekius. Statistinis dažnų posekių paieškos algoritmas nėra tikslus, bet šio algoritmo veikimo laikas yra daug trumpesnis, palyginti su tiksliais *Apriori*, GSP ar rekursiniu algoritmais. Straipsnyje aprašyti du eksperimentai. Pirmajame eksperimente 100 rinkmenų apdorota statistiniu dažnų posekių paieškos algoritmu po 100 kartų. Nustatyta, kad algoritmo veikimo laikas priklauso nuo p reikšmės: kuo p reikšmė didesnė, tuo ilgesnis algoritmo vykdymo laikas. Visuose bandymuose dažnų posekių pasirinktas ilgis – penki simboliai. Apdorojus eksperimentų rezultatus buvo apskaičiuotos šios charakteristikos: fragmentų skaičiaus ir imčių skaičiaus santykis, imčių skaičiaus ir simbolių skaičiaus rinkmenoje santykis, rastų fragmentų skaičiaus ir paslėptų fragmentų skaičiaus santykis, paslėptų fragmentų skaičiaus ir simbolių skaičiaus rinkmenoje santykis. Atlikus eksperimentą nustatyta, kad jei $p \in (0; 0,5)$ – fragmentų skaičiaus ir imčių skaičiaus santykis yra apytiksliai lygus rastų fragmentų skaičiaus ir paslėptų fragmentų skaičiaus santykiui, o jei $p \in (0,5; 1)$ – rastų fragmentų skaičiaus ir paslėptų fragmentų skaičiaus santykis yra apytiksliai lygus paslėptų fragmentų skaičiaus ir simbolių skaičiaus rinkmenoje santykiui. Antrajame eksperimente buvo generuota 19 rinkmenų grupių po 100 rinkmenų, t. y. iš viso apdorota 1900 rinkmenų. Apdorojus šio eksperimento rezultatus, prieita prie išvados, kad klaidų tikimybė padidėja apdorojant šiuo algoritmu rinkmenas, kuriose nėra vieno dažnumu smarkiai išsiskiriančio posekio. Statistinis dažnų posekių paieškos algoritmas aptinka dažną seką 95,83 % tikslumu, pasikliovimo tikimybės intervalas yra [0,95377541; 0,99938299].

LITERATŪRA

AYRES, Jay; FLANNICK, Jason; GEHRKE, Johannes; YIU, Tomi (2002). Sequential Pattern mining using a bitmap representation. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 429–435.

AGRAWAL, Rakesh; SRIKANT, Ramakrishnan (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*.

CAI-YAN, Jia; XIE-PING, Gao (2005). Multi-scaling sampling: an adaptive sampling method for discovering approximate association rules. *Journal of Computer Science and Technology*, Vol. 20, p. 309–318.

CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford (2009). *Introduction to Algorithms*. 3th edition. London: The MIT Press Cambridge. ISBN 978-0-262-53305-8.

GYENESEI, Attila; TEUHOLA, Jukka (2003). Probabilistic Iterative Expansion of Candidates in Mining Frequent Itemsets. *CEUR*, Vol. 90.

HUANYIN, Zhou; JINSHENG, Liu (2009). The Research of A-Priori Algorithm Candidates Based

on Support Counts. In *International Conference on Information Technology and Computer Science*, Vol. 1, p. 192–195.

JUOZAPAVIČIUS, Algimantas (2007). *Duomenų struktūros ir efektyvūs algoritmai.* Vilnius: TEV. ISBN 978-9955-680-87-1.

PRAGARAUSKAITĖ, Julija; DZEMYDA, Gintautas (2009). Tikimybinis dažnų posekių paieškos algoritmas. *Informacijos mokslai*, t. 50, p. 352–357.

TOIVONEN, Hannu (1996). Sampling Large Databases for Association Rules. In *Proceedings of the 22nd International Conference on Very Large Databases,* Mumbai, India, p. 134–145.

TUMASONIS, Romanas; DZEMYDA, Gintautas (2004). The Probabilistic Algorihm for Mining Frequent Sequences. In *Proceedings ADBIS'04 Eight East-European Conference on Advances in Databases and Information Systems*, p. 89–98.

1 priedas

Rinkmenų grupės ir jų generavimo charakteristikos

Rinkmenų grupės pavadinimas	Simbolių skaičius rinkmenoje	Paslėptas fragmentas	Paslėpto fragmento tikimybė	Kiti simboliai ir jų generavimo tikimybės
0.1.*.txt	100000	SIENA	0,1	S – 0,18 I – 0,18 E – 0,18 N – 0,18 A – 0,18
0.09.*.txt	100000	SIENA	0,09	S – 0,19 I – 0,18 E – 0,18 N – 0,18 A – 0,18
0.08.*.txt	100000	SIENA	0,08	S – 0,19 I – 0,19 E – 0,18 N – 0,18 A – 0,18
0.07.*.txt	100000	SIENA	0,07	S – 0,19 I – 0,19 E – 0,19 N – 0,18 A – 0,18
0.06.*.txt	100000	SIENA	0,06	S – 0,19 I – 0,19 E – 0,19 N – 0,19 A – 0,18
0.05.*.txt	100000	SIENA	0,05	S – 0,19 I – 0,19 E – 0,19 N – 0,19 A – 0,19

0.04.*.txt	100000	SIENA	0,04	S – 0,2 I – 0,19 E – 0,19 N – 0,19 A – 0,19
0.03.*.txt	100000	SIENA	0,03	S – 0,2 I – 0,2 E – 0,19 N – 0,19 A – 0,19
0.02.*.txt	100000	SIENA	0,02	S – 0,2 I – 0,2 E – 0,2 N – 0,19 A – 0,19
0.01.*.txt	100000	SIENA	0,01	S – 0,2 I – 0,2 E – 0,2 N – 0,2 A – 0,19
0.009.*.txt	100000	SIENA	0,009	S – 0,2 I – 0,2 E – 0,2 N – 0,2 A – 0,191
0.008.*.txt	100000	SIENA	0,008	S – 0,2 I – 0,2 E – 0,2 N – 0,2 A – 0,192
0.007.*.txt	100000	SIENA	0,007	S – 0,2 I – 0,2 E – 0,2 N – 0,2 A – 0,193
0.006.*.txt	100000	SIENA	0,006	S – 0,2 I – 0,2 E – 0,2 N – 0,2 A – 0,194
0.005.*.txt	100000	SIENA	0,005	S – 0,2 I – 0,2 E – 0,2 N – 0,2 A – 0,195
0.004.*.txt	100000	SIENA	0,004	S – 0,2 I – 0,2 E – 0,2 N – 0,2 A – 0,196

0.003.*.txt	100000	SIENA	0,003	S – 0,2 I – 0,2 E – 0,2 N – 0,2 A – 0,197
0.002.*.txt	100000	SIENA	0,002	S – 0,2 I – 0,2 E – 0,2 N – 0,2 A – 0,198
0.001.*.txt	100000	SIENA	0,001	S – 0,2 I – 0,2 E – 0,2 N – 0,2 A – 0,199

2 priedas

Klaidos priklausomybė nuo rinkmenos turinio

Rinkmenų grupės pavadinimas	Paslėpto fragmento tikimybė	Statistinio algoritmo vykdymas, kai p	Rinkmenų skaičius grupėje, kai klaidingai rastas dažnas posekis	Klaida procentais
0.01.*.txt	0,01	0	38	38 %
0.009.*.txt	0,009	0	35	35 %
0.008.*.txt	0,008	0	14	14 %
0.007.*.txt	0,007	0	5	5 %
0.006.*.txt	0,006	0	9	9 %
0.005.*.txt	0,005	0	7	7 %
0.004.*.txt	0,004	0	5	5 %
0.003.*.txt	0,003	0	4	4 %
0.002.*.txt	0,002	0	4	4 %
0.001.*.txt	0,001	0	8	8 %
0.01.*.txt	0,01	0,1	38	38 %
0.009.*.txt	0,009	0,1	32	32 %
0.008.*.txt	0,008	0,1	16	16 %
0.007.*.txt	0,007	0,1	8	8 %
0.006.*.txt	0,006	0,1	7	7 %
0.005.*.txt	0,005	0,1	3	3 %
0.004.*.txt	0,004	0,1	2	2 %
0.003.*.txt	0,003	0,1	6	6 %
0.002.*.txt	0,002	0,1	6	6 %
0.001.*.txt	0,001	0,1	6	6 %
0.02.*.txt	0,02	0,2	1	1 %
0.01.*.txt	0,01	0,2	37	37 %
0.009.*.txt	0,009	0,2	33	33 %
0.008.*.txt	0,008	0,2	12	12 %
0.007.*.txt	0,007	0,2	8	8 %

0.006.*.txt	0,006	0,2	4	4 %
0.005.*.txt	0,005	0,2	3	3 %
0.004.*.txt	0,004	0,2	4	4 %
0.003.*.txt	0,003	0,2	2	2 %
0.002.*.txt	0,002	0,2	3	3 %
0.001.*.txt	0,001	0,2	4	4 %
0.01.*.txt	0,01	0,3	42	42 %
0.009.*.txt	0,009	0,3	21	21 %
0.008.*.txt	0,008	0,3	9	9 %
0.007.*.txt	0,007	0,3	9	9 %
0.006.*.txt	0,006	0,3	5	5 %
0.005.*.txt	0,005	0,3	4	4 %
0.004.*.txt	0,004	0,3	5	5 %
0.003.*.txt	0,003	0,3	4	4 %
0.002.*.txt	0,002	0,3	2	2 %
0.001.*.txt	0,001	0,3	3	3 %
0.01.*.txt	0,01	0,4	38	38 %
0.009.*.txt	0,009	0,4	24	24 %
0.008.*.txt	0,008	0,4	10	10 %
0.007.*.txt	0,007	0,4	5	5 %
0.006.*.txt	0,006	0,4	3	3 %
0.005.*.txt	0,005	0,4	2	2 %
0.004.*.txt	0,004	0,4	4	4 %
0.003.*.txt	0,003	0,4	4	4 %
0.002.*.txt	0,002	0,4	2	2 %
0.001.*.txt	0,001	0,4	3	3 %
0.01.*.txt	0,01	0,5	38	38 %
0.009.*.txt	0,009	0,5	31	31 %
0.008.*.txt	0,008	0,5	11	11 %
0.007.*.txt	0,007	0,5	7	7 %
0.006.*.txt	0,006	0,5	1	1 %
0.005.*.txt	0,005	0,5	4	4 %
0.004.*.txt	0,004	0,5	3	3 %
0.003.*.txt	0,003	0,5	3	3 %
0.002.*.txt	0,002	0,5	1	1 %
0.001.*.txt	0,001	0,5	1	1 %
0.01.*.txt	0,01	0,6	44	44 %
0.009.*.txt	0,009	0,6	23	23 %
0.008.*.txt	0,008	0,6	8	8 %
0.007.*.txt	0,007	0,6	4	4 %
0.006.*.txt	0,006	0,6	1	1 %
0.005.*.txt	0,005	0,6	2	2 %
0.004.*.txt	0,004	0,6	2	2 %
0.003.*.txt	0,003	0,6	1	1 %
0.002.*.txt	0,002	0,6	1	1 %
0.01.*.txt	0,01	0,7	40	40 %

0.009.*.txt	0,009	0,7	26	26 %
0.008.*.txt	0,008	0,7	14	14 %
0.007.*.txt	0,007	0,7	4	4 %
0.006.*.txt	0,006	0,7	2	2 %
0.005.*.txt	0,005	0,7	2	2 %
0.004.*.txt	0,004	0,7	1	1 %
0.002.*.txt	0,002	0,7	1	1 %
0.001.*.txt	0,001	0,7	2	2 %
0.01.*.txt	0,01	0,8	34	34 %
0.009.*.txt	0,009	0,8	28	28 %
0.008.*.txt	0,008	0,8	10	10 %
0.007.*.txt	0,007	0,8	7	7 %
0.006.*.txt	0,006	0,8	1	1 %
0.004.*.txt	0,004	0,8	2	2 %
0.003.*.txt	0,003	0,8	1	1 %
0.002.*.txt	0,002	0,8	2	2 %
0.001.*.txt	0,001	0,8	2	2 %
0.01.*.txt	0,01	0,9	34	34 %
0.009.*.txt	0,009	0,9	31	31 %
0.008.*.txt	0,008	0,9	3	3 %
0.007.*.txt	0,007	0,9	2	2 %
0.005.*.txt	0,005	0,9	1	1 %
0.004.*.txt	0,004	0,9	1	1 %
0.002.*.txt	0,002	0,9	1	1 %
0.01.*.txt	0,01	1	41	41 %
0.009.*.txt	0,009	1	27	27 %
0.008.*.txt	0,008	1	3	3 %
0.007.*.txt	0,007	1	1	1 %
0.002.*.txt	0,002	1	1	1 %
0.001.*.txt	0,001	1	2	2 %

STATISTICAL ALGORITHM FOR MINING FREQUENT SEQUENCES

Loreta Savulioniene, Leonidas Sakalauskas

S u m m a r y

Modern life involves large amounts of data and information. Search is one of the major operations performed by a computer. Search goal is to find a sequence (element) in large amounts of data or to confirm that it does not exist. Amounts of data in databases have reached terabytes, and therefore data retrieval, analysis, rapid decision-making become increasingly complicated. Large quantities of information cover both important and void information. The main goal of data mining is to find the meaning in data, i.e. a relationship between the data, their reproducibility, etc. This technology applies to business, medicine and other areas where large amounts of information are processed and a relationship among data is detected, i.e. new information is obtained from large amounts of data. The paper proposes a new statistic algorithm for repeated sequence search. The essence of this statistic algorithm is to identify repeated sequences quickly. During the algorithm all contents of the file are not checked several times. During the algorithm, the file is checked once according to the chosen probability *p*. This algorithm is inaccurate, but its execution time is shorter than of the accurate algorithms.