STOCHASTINIS DAŽNŲ POSEKIŲ PAIEŠKOS ALGORITMAS

Loreta Savulionienė, Leonidas Sakalauskas
Vilniaus universiteto Matematikos ir informatikos institutas

Įvadas

Dabar bet kuri veikla susijusi su informacija, duomenimis, todėl vis svarbiau dideliuose informacijos ir duomenų kiekiuose rasti reikiamą elementą ar objektą, seką ar posekį. Be abejo, paieška yra vienas iš pagrindinių kompiuterio darbo operacijų. Dažnų posekių paieška labai svarbi šiandieninėje veikloje, kai apdorojami dideli informacijos ir duomenų kiekiai. Duomenų bazės pasiekė terabaitinį dydį, todėl tikslių paieškos algoritmų naudojimas reikalauja labai didelių kompiuterio laiko sąnaudų, nes, vykdant šiuos dažnų posekių paieškos algoritmus, tenka daug kartų perrinkti duomenų bazę. Tikslių algoritmų naudojimą dažnų posekių paieškai didelėse duomenų bazėse pakeičia apytikslių paieškos algoritmų taikymas. Aktualiausia nustatyti, kuris posekis dažnas, o ne tikslų dažnų posekių skaičių. Dažnų posekių paieška aktuali daugelyje veiklos sričių, t. y. tiek versle, tiek pramonėje, medicinoje ir t. t.

Šiame straipsnyje pasiūlytas stochastinis dažnų posekių paieškos algoritmas, pristatomi eksperimentinio tyrimo rezultatai bei išvados. Vykdant šį algoritmą, duomenų bazė skenuojama vieną kartą ir nustatomi dažni posekiai. Stochastinis algoritmas tinkamas pirkinių krepšelio analizės (market basket analysis), aptarnavimo kokybės, genetikos uždaviniams spręsti ir pan.

Tyrimo tikslas – sudaryti dažnų posekių paieškos algoritmą, įvertinti jo patikimumą.

Uždaviniai – didžiausio tikėtinumo metodas dažno posekio tikimybei įvertinti, dažno posekio nustatymo pirmos ir antros rūšies klaidų įvertinimas, pasikliovimo tikimybės rėžių nustatymas.

Tyrimo metodai – dažno posekio nustatymo statistinių hipotezių tikrinimo, tikimybės pasikliovimo rėžių, tikėtinumo funkcijos ir Monte Karlo metodai.

Tyrimo rezultatai – stochastinis dažnų posekių paieškos algoritmas. Šio algoritmo patikimumas įvertintas atlikus kompiuterinį eksperimentą.

Appriori algoritmų apžvalga

Sekų paieškos algoritmai pirmiausia buvo nagrinėjami Rakesh Agrawal ir Ramakrishnan Srikant darbuose. Šiuose darbuose buvo išnagrinėtas klasikinis Appriori algoritmas. Pirmojo Apriori algoritmo žingsnio metu nustatomi dažni vieno elemento rinkiniai. Vykdant šį algoritmo žingsnį, pereinama visa duomenų rinkmena ir nustatoma, kiek kartų kiekvienas elementas sutinkamas rinkmenoje, ir tolimesniam apdorojimui naudojami tik tie elementai, kurie tenkina nustatytą minimalų pasirodymų dažnį.

Tarkime, turime duomenų bazę A. Pirmą kartą analizuojama duomenų bazė. Gautas B rezultatas. Antrą kartą analizuojama duomenų bazė. Gautas C rezultatas. Trečią kartą analizuojama duomenų bazė. Gautas D rezultatas. Ketvirtą kartą analizuojama duomenų bazė. Gautas E rezultatas. Penktą kartą naujų kandidatų negauta. Tokiu būdu maksimalūs dažni posekiai yra <1, 2, 3, 4>, <1, 3, 5> ir <4, 5>. Šie posekiai ir yra ieškomi šablonai.

Kiti algoritmo žingsniai susideda iš dviejų dalių: potencialiai dažnų elementų rinkinių generavimo (jie vadinami *kandidatais*) ir rinkinių kandidatų dažnumo nustatymo. Apriori algoritmas generuoja kito žingsnio elementų rinkinius kandidatus 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 rinkiniai kandidatai, sudaryti k elementų, generuojami sujungiant dažnus elementų rinkinius, turinčius k–1 elementų, kurie tenkina minimalų pasikartojimų skaičių (Agrawal, Srikant, 1994).

Algoritmas AprioriAll kiekvieną kartą, kai analizuojama duomenų bazė, naudoja posekius, kurie gauti priešpaskutiniame analizavime. Tada generuojamos sekos – kandidatai bei atliekamas jų „palaikymo“ skaičiavimas, analizuojat duomenų bazę. Dažni rinkiniai, kurie aptinkami tikrinant pirmą kartą yra 1 – nario posekiai. Kartais šitas procesas vadinamas inicializacija. Šis algoritmas formuoja sekos visų galimų ilgių posekius. Tačiau jeigu tam tikro ilgio posekių yra mažai – tai šį ilgį galima praleisti. Algoritmas, kaip parametrą, naudoja posekių ilgį, kurie buvo analizuoti prieš tai buvusiame žingsnyje ir grąžina posekių ilgį, kurie bus analizuojami kitame žingsnyje (Huanyin, Jinsheng, 2009).

Jeigu raide t pažymėtas duomenų bazės tikrinimo numeris, tai tada galima užrašyti, kad $k(t + 1) = k(t) + p$. Tai reiškia, kad kitame žingsnyje bus analizuojami posekiai, kurie yra ilgesni p dydžiu. Tuo atveju, kai $p = 1$, tai algoritmas yra analogiškas algoritmui Apriori, t. y. bus analizuojami visų ilgių posekiai. Uždavinio tikslas – apibrėžti, kokio ilgio posekiai gali būti praleisti. Apibrėžkime dažnų

k – posekių ir posekių kandidatų santykį $h_k = F_k/C_k$. Intuityviai aišku, kad kandidatų kiekis, tenkinantis minimalius reikalavimus, rastų einamajame tikrinime didėja, tai laikas, kuris bus sugaištas analizuojat kandidatus (kurių ilgis yra nedidelis) nenagrinėjant visų galimų ilgių – mažėja.

Jeigu k-ajame tikrinime bus nustatyta, kad dalinių sekų posekis F_{k-1} yra tuščias, tai suformuoti galimų kandidatų posekį C_k bus neįmanoma. Tada galima elgtis taip: norit suformuoti C_k galima naudoti kandidatų aibę C_{k-1} (Srikant, Agrawal, 1996).

Algoritmas AprioriSome duomenų bazės analizuojamo metu apdoroja tik apibrėžto k ilgio posekius. Visų galimų ilgių posekių analizė yra laikui imli procedūra, todėl kyla klausimų, ar galima sutrumpinti šią procedūrą; jeigu bus atliekama ne visų posekių analizė – kas bus prarandama ir pan. (Agrawal, Srikant, 1997).

Algoritmas DymanicSome tiesioginėje eigoje praleidžia apibrėžto ilgio posekių kandidatų palaikymo skaičiavimą. Inicializavimo metu apskaičiuojamos visi L ilgio posekiai kandidatai. Po to apdorojami posekiai, kurių ilgis yra L kartotinis. Kai L = 3, tai inicializavimo metu bus apdorojami posekiai, kurių ilgiai 1, 2, tiesioginėje eigoje apdorojami posekiai, kurių ilgiai 3, 6, 9, 12, … (Toivonen, 1996).

Kai kuriais atvejais galima nevykdyti inicializavimo proceso. Posekiai, kurių ilgis yra lygus 6, gali būti sudaromi sujungiant dvi sekas, kurių ilgis lygus 3. Posekis, kurios ilgis 9, gali būti sudaromas jungiant posekius, kurių ilgiai 3 ir 6 ir t. t.

Appriori algoritmų efektyvumo palyginimas

Remiantis atliktų eksperimentų skaičiavimais, DynamicSome algoritmas, kai yra minimali palaikymo reikšmė, generuoja labai daug sekų kandidačių, o tai reikalauja didelių laiko sąnaudų. Net ir tuo atveju, kai pakanka kompiuterio operatyviosios atminties resursų, palaikymo skaičiavimų laiko sąnaudos yra didesnės negu AprioriSome algoritmo. Akivaizdu, kad visų algoritmų vykdymo laikas didėja, kai mažinamas dažno posekio palaikymas, nes tuo atveju didėja dažnų posekių kiekis. Pagrindinis algoritmo AprioriSome pranašumas, lyginant su algoritmu AprioriAll, tai, kad išvengiama trumpesnių nei nurodyto ilgio posekių skaičiavimo. Tačiau šis privalumas sumažėja dėl dviejų priežasčių. Pirmiausiai, kandidatai C_k yra aibės L_{k-1} poaibis, kandidatų skaičius, kurį generuoja algoritmas AprioriSome gali būti didesnis. Antra priežastis, nors algoritmas AprioriSome praleidžia kai kurių ilgių posekių analizę, mažesnio ilgio nei nurodytas ilgis posekiai vis tiek generuojami ir jie naudoja kompiuterio atmintį.

Stochastinis dažnų posekių paieškos algoritmas

Dažnų posekių paieška aktuali daugelyje veiklos sričių, t. y. tiek versle, tiek pramonėje, medicinoje ir t. t. Stochastinis algoritmas gali būti taikomas pirkinių krepšelio analizės (market basket analysis), aptarnavimo kokybės, genetikos uždaviniams spręsti ir pan. Kuriamo stochastinio algoritmo tikslas –nustatyti dažnus posekius didelėse duomenų bazėse.

Tegul *M* ilgio duomenų bazėje *D* dažnai pasitaiko tam tikras vienas fragmentas, kurio ilgis iš anksto nėra žinomas. Šiam dažnam fragmentui nustatyti analizuojami atsitiktinai pasirinkti atsitiktinio ilgio *l* posekiai. Tegul analizuojamų posekių ilgis yra pasiskirstęs pagal geometrinį dėsnį su parametru *q*, o tarpų tarp dviejų analizuojamų posekių ilgiai taip pat pasiskirstę pagal geometrinį dėsnį su parametru *p*.

Nesunku apskaičiuoti, kad vidutinis analizuojamo posekio ilgis yra $l = q/(1-q)$, o vidutinis tarpo tarp gretimų posekių ilgis yra lygus $t = p/(1-p)$. Tegul, analizuojant duomenų bazę D, atsitiktinai pasirinkta *N* (imčių skaičius) įvairaus ilgio posekių. Atitinkamo ilgio posekių dažniai c_i apskaičiuojami pagal šią formulę:

$$c_i = N_i/N, \text{ kur i} = 1, 2, \ldots, \text{n},$$

N_i – atitinkamo ilgio posekių skaičius, N – visų posekių skaičius, i – posekio ilgis, n – maksimalus posekio ilgis.

Tegul nagrinėjamos dvi nepriklausomos posekių imtys, jų dydžiai yra n_1 ir n_2; pirmojoje imtyje dažniausias posekis pasitaikė k_1 kartų, o antrojoje – k_2 kartų.

Nulinė hipotezė tvirtina, kad dažniausių posekių proporcijos duomenų bazėje, iš kurios imtys paimtos, yra vienodos, o alternatyva tvirtina, kad tos tikimybės nelygios:

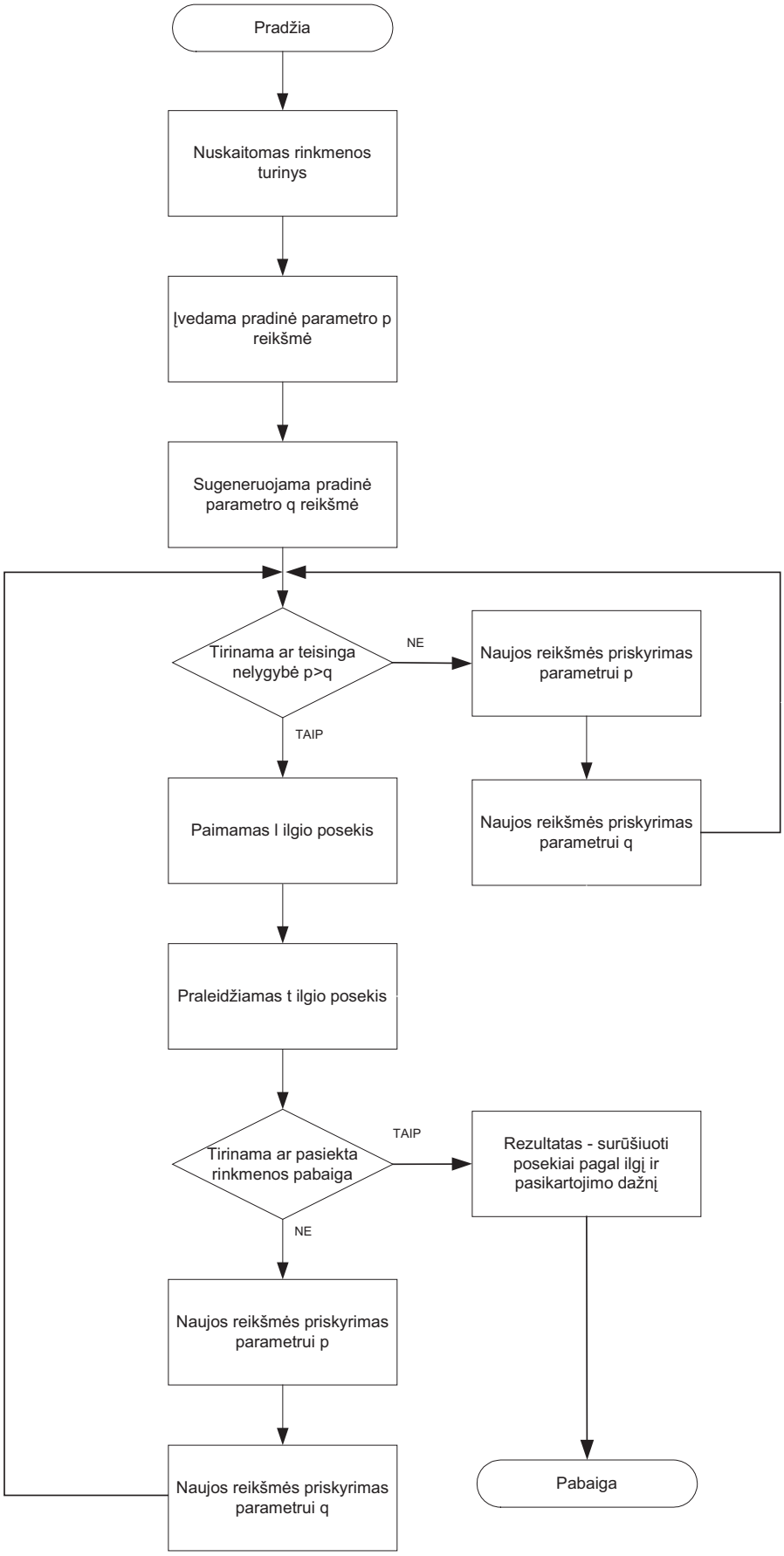
$$H_0 : p_1 = p_2$$
$$H_1 : p_1 \neq p_2$$

Priimant arba atmetant hipotezę H_0, galimos dviejų rūšių klaidos. Jos vadinamos pirmosios ir antrosios rūšies klaidomis. Pirmosios rūšies klaida: hipotezė H_0 atmetama, kai ji teisinga. Antrosios rūšies klaida: hipotezė H_0 priimama, kai ji klaidinga. Stochastinis dažnų posekių paieškos algoritmas yra apytikslis, todėl galimos pirmos ir antros rūšies klaidos. Fiksuojamas posekis $c_{i1}, c_{i2}, \ldots, c_{ik}$.

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.

Stochastinio dažnų posekių paieškos algoritmo veikimo schema pateikta 1 paveiksle.



1 pav. Stochastinio dažnų posekių paieškos algoritmo veikimo schema

Stochastinio dažnų posekių paieškos algoritmo statistikų įvertinimas

1. Pasikliovimo tikimybės rėžis

Intervalas $[p_1; p_2]$ vadinamas parametro p pasikliautinuoju intervalu, jei $P(p_1 < p < p_2) = \alpha$, skaičius α vadinamas pasikliovimo lygmeniu, p_1 ir p_2 vadinami pasikliovimo tikimybės rėžiais. Stochastinio algoritmo tikslumo kriterijus – tai fragmento radimo tikimybė.

Fragmento radimo tikimybės pasikliovimo rėžiai įvertinami pagal šias formules:

$$p_1 = 1 - \text{BetaInv}(\frac{1-\alpha}{2}, n-k, k+1);$$

$$p_2 = 1 - \text{Betanv}(1-(\frac{1-\alpha}{2}), n-k+1, k).$$

n – visų fragmentų skaičius,
k – duoto fragmento pasirodymų skaičius,
BetaInv – beta skirstinio kvantilis.

2. Kriterijaus statistika (žymima raide *u*)

Turimos dvi nepriklausomos imtys, jų dydžiai yra n_1 ir n_2. Pirmojoje imtyje rasta k_1, o antrojoje k_2 objektų, turinčių rūpimąją požymio reikšmę.

Kriterijaus hipotezėje apie požymio radimo tikimybių lygybę imtyse statistika gali būti įvertinama įvairiais būdais. Kriterijaus statistika *u* konstruojama taip, kad H_0 esant teisingai ji būtų pasiskirsčiusi pagal standartinį normalųjį skirstinį. Kriterijaus statistika *u* apskaičiuojama pagal šią formulę (Čekanavičius, Murauskas, 2000):

$$u = \frac{d_1 - d_2}{\left(\frac{k_1+k_2}{n_1+n_2}\right)\cdot\left(1-\frac{k_1+k_2}{n_1+n_2}\right)\cdot\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}.$$

Pažymėjus $d = (k_1 + k_2) / (n_1 + n_2)$, gaunama tokia formulė:

$$u = \frac{d_1 - d_2}{d\cdot(1-d)\cdot\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}.$$

Kriterijaus statistiką *u* galima įvertinti ir taip (Čekanavičius, Murauskas, 2000):

$$u = \left(2\arcsin\sqrt{d_1} - 2\arcsin\sqrt{d_2}\right)\sqrt{\frac{n_1\cdot n_2}{n_1+n_2}}.$$

Nagrinėjamos dvi nepriklausomos posekių imtys, jų dydžiai yra n_1 ir n_2 ir pirmojoje imtyje dažniausias posekis pasitaikė k_1 kartų, o antrojoje – k_2 kartų.

3. Prielaidų vertinimas

Įvertinus kriterijaus statistiką, atliekamas prielaidų tikimybių vertinimas. Kai alternatyva dvipusė (H_1: $p_1 \neq p_2$), gautąją *u* reikšmę atitinkanti *p*-reikšmė apskaičiuota taip:

$$p = 2 - (1\text{-NORMSDIST(ABS(u))}).$$

p-reikšmė reiškia tikimybę rizikos, kad atmetant H_0 bus padaryta pirmos rūšies klaida, todėl H_0 pagrįstai atmesti galima tik tada, kaip *p*-reikšmė gaunama nedidelė, nežymi, mažesnė už įprastinius, tradicinius reikšmingumo lygmėms (0,1; 0,05; 0,01 ar 0,001). *p*-reikšmė išreiškia hipotezės H_0 tikėtinumą, t. y. tikimybę, kad joje išsakytas teiginys atitinka tikrovę. Todėl kuo didesnė *p*-reikšmė, tuo labiau nulinė hipotezė pasikliautina.

Sekų charakteristikų pasikeitimo momentui nustatyti sukurtas modelis, kuriame dažniausiai pasitaikantis fragmentas nustatomas didžiausio tikėtinumo būdu. Dažno fragmento ilgis nustatomas remiantis monotoniškumo taisykle – dažnų posekių poaibis yra dažnas posekis. Sekų charakteristikų pasikeitimo momentų nustatymo modelyje apskaičiuojamas dažniausio fragmento dažnis, priklausomai nuo fragmento ilgio. Toliau pasinaudojama tuo, kad kol fragmento ilgis yra mažesnis už paslėpto fragmento ilgį, tai nagrinėjamo fragmento dažnis yra beveik pastovus, ir kai tiriamo fragmento ilgis yra didesnis už tiriamo dažnaiusio fragmento, tai tikimybė pradeda mažėti. Tokiu atveju sudaromas binarinis procesas, kurio reikšmė lygi vienetui, jei statistinis kriterijus neprieštarauja hipotezei apie dviejų gretimo ilgio fragmentų pasirodymo tikimybės sutapimą ir lygi nuliui, jei dviejų gretimo ilgio fragmento pasirodymo tikimybės reikšmingai skiriasi. Atpažįstant paslėptą fragmentą, manoma, kad dviejų gretimo ilgio fragmentų dažniai sutampa tol, kol fragmento ilgis neviršija paslėpto fragmento ilgio (4 pav.). Šio binarinio proceso charakteristikų pasikeitimo momentas turi sutapti su paslėpto fragmento ilgiu. Kadangi dviejų gretimo ilgio fragmentų pasirodymų skaičius imtyse yra pasiskirstęs pagal binominį dėsnį, galima sudaryti logaritminę tikėtinumo funkciją dažniausiai pasitaikančio fragmento pasirodymo tikimybės pasikeitimui nustatyti:

$$f(k) = \frac{k!}{k_1!\cdot(k-k_1)!}\cdot\frac{(N-k)!}{(K-k_1)!\cdot(N-k-K+k_1)!}\cdot p_1^{k_1}\cdot(1-p_1)^{k-k_1}\cdot p_2^{K-k_1}\cdot(1-p_2)^{N-k-K+k_1}$$

Iš čia gauname logaritminę tikėtinumo funkciją:

$$\ln(f(k)) = \sum_{i=1}^{k}\ln i - \sum_{i=1}^{k_1}\ln i - \sum_{i=1}^{k-k_1}\ln i + \sum_{i=1}^{N-k}\ln i - \sum_{i=1}^{K-k_1}\ln i - \sum_{i=1}^{N-k-K+k_1}\ln i + \sum_{i=1}^{k_1}p_1 + \sum_{i=1}^{k-k_1}\ln(1-p_1) + \sum_{i=1}^{K-k_1}\ln p_2 + \sum_{i=1}^{N-k-K+k_1}(1-p_2),$$

k – binarinio proceso charakteristikų pasikeitimo momentas, k_1 – tikimybių dažnių sutapimų skaičius iki pasikeitimo momento, k_2 – tikimybių dažnių sutapimų skaičius po pasikeitimo momento, N – maksimalus posekio ilgis. Dažniausio fragmento ilgis atitinka logaritminės tikėtinumo funkcijos minimumą. Minimizuojančią funkciją yra patogu įvesti kaip dviejų gretimų šios funkcijos reikšmių skirtumą, kuris yra lygus:

$$\ln(f(k)\ / f(k-1)\ = \text{h}\ k - \ln(k-k_1) - \ln(N-k+1) + \ln(N-k-k_1+1) + \ln(1-p_1) + \ln(1-p_2),$$

jeigu k-oji binarinio proceso reikšmė yra lygi 0. Jei ši reikšmė lygi 1, tai yra gretimų tikėtinumo funkcijos reikšmių skirtumas yra lygus:

$$\ln(f(k))/f(k-1)) = \ln k - \ln k_1 - \ln(N-k+1) + \ln(K-k_1+1) + \ln p_1 + \ln p_2,$$

čia $p_1 = \frac{k_1}{k}$, $p_2 = \frac{k_2}{k}$. Tikėtinumo funkcijos minimumas sutampa su pirmąja kintamojo k reikšme, kuriai dviejų gretimų šios funkcijos reikšmių skirtumas yra teigiamas. Skaičiavimai pradedami pradinės reikšmės k = 0. Nesunku pastebėti, kad pradinė tikėtinumo funkcijos reikšmė yra:

$$\ln(f(0)) = \sum_{i=1}^{N} \ln i - \sum_{i=1}^{K} \ln i - \sum_{i=1}^{N-K} \ln i + K \cdot \ln p_2 + \ln(1-p_2)\cdot(N-K).$$

Logaritminės tikėtinumo funkcijos reikšmėms apskaičiuoti, galima naudotis rekurentinėmis formulėmis:

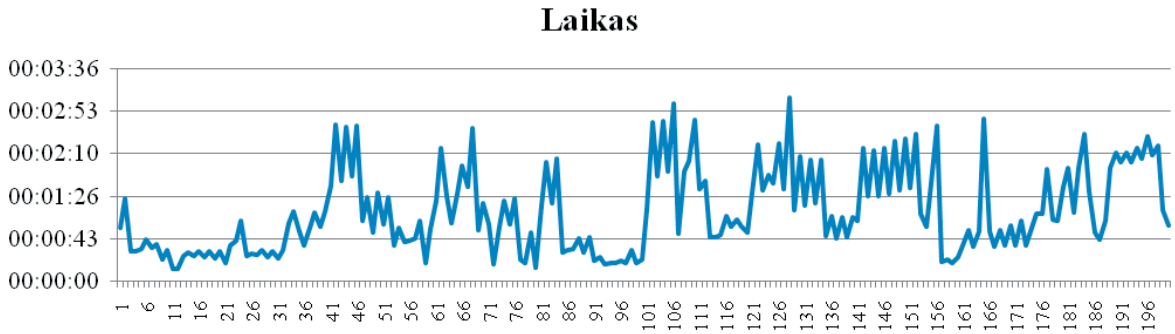
$k_1(k+1)=k_1(k)$, $k_2(k+1)=k_2(k)$, jei k-oji binarinio proceso reikšmė yra lygi nuliui ir $k_1(k+1)=k_1(k)+1$, $k_2(k+1)=k_2(k)-1$.

Šis algoritmas leidžia suderinti du svarbius kriterijus, t. y. laiką ir tikslumą, atitinkamai parenkant parametrų p ir q reikšmes.

Kompiuterinis modeliavimas

Eksperimentui buvo sugeneruota duomenų bazė. Duomenų bazės ilgis 100000 simbolių, paslėptas fragmentas SIENA su tikimybe 0,2. Kiti simboliai Q, W, Z, X, kurių generavimo tikimybės atitinkamai lygios 0,3; 0,2; 0,2 ir 0,1. Stochastiniu dažnų posekių paieškos algoritmu duomenų bazė analizuota 200 kartų.

Vidutinis duomenų bazės analizavimo laikas 00:01:14. Standartinis nuokrypis yra 00:00:57.



2 pav. Duomenų bazės apdorojimo laikas

Mažiausias imčių skaičius 9436, didžiausias imčių skaičius 18361, vidutinis imčių skaičius 14283. Imčių skaičius priklauso nuo parametrų p ir q parinkimo, t. y. kai p ≥ q, tai imčių skaičius beveik du kartus didesnis už imčių skaičių, kai p < q. Kiekvienoje imtyje sugeneruoti posekiai buvo sugrupuoti pagal ilgį ir apskaičiuoti posekių dažniai, įvertinant anksčiau aprašytas tikimybines charakteristikas.

Eksperimento metu buvo įvertintas vidutinis dažnų posekių skaičius pagal posekių ilgį (3 pav.).

3 pav. Vidutinis dažnų posekių skaičius

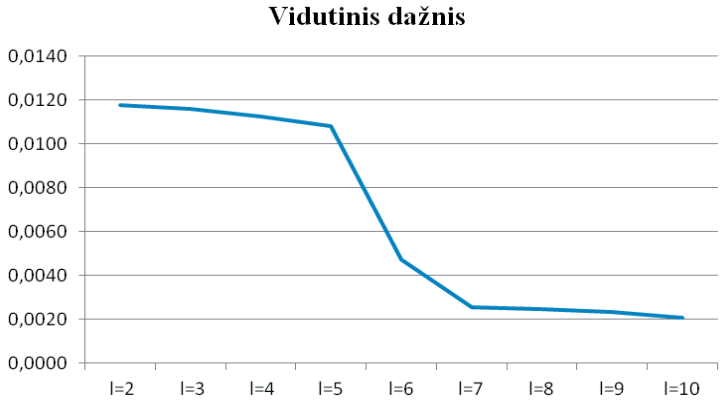
Iš pateiktos priklausomybės matyti, kad dažnų posekių skaičius žymiai sumažėja, kai posekio ilgis viršija fragmento ilgį.

Dažniausi posekiai pagal ilgį pateikti 1 lentelėje:

1 lentelė. ***Dažniausi posekiai***

Posekio ilgis l	Dažniausias posekis	Bandymų skaičius, kai posekis buvo dažniausias	Procentai
1	Q	200	100 %
2	NA	52	26 %
3	IEN	77	38,5 %
4	SIEN	104	52 %
5	SIENA	200	100 %
6	QSIENA	118	59 %
7	NASIENA	52	26 %
8	ENASIENA	72	36 %
9	IENASIENA	109	54,5 %
10	SIENASIENA	196	98 %

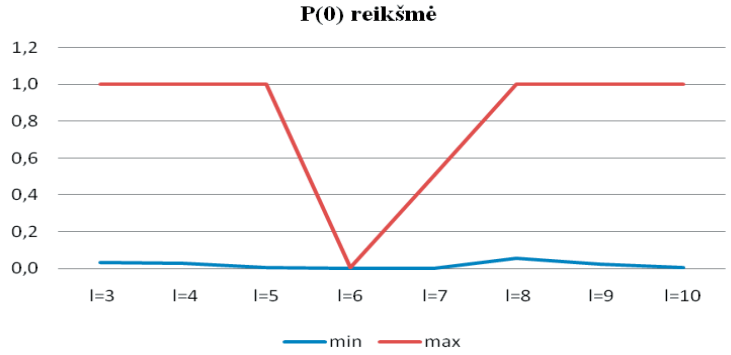
Per eksperimentą buvo įvertintas vidutinis posekių dažnis, priklausomai nuo dažniausio posekio ilgio (4 pav.).



4 pav. Vidutinis posekių dažnis

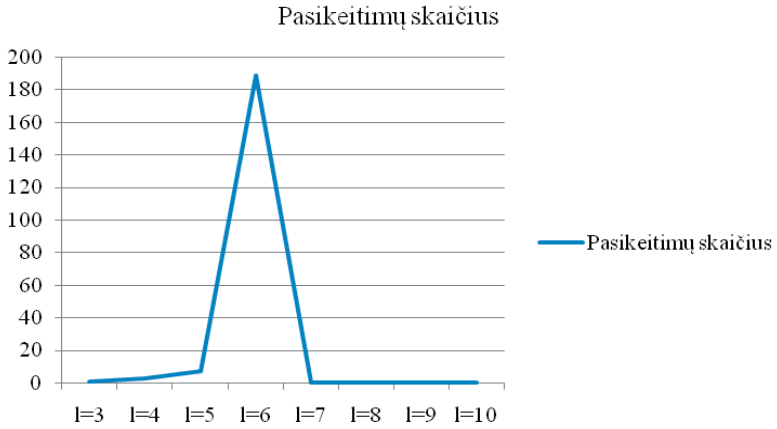
Teisingo dažniausio posekio nustatymo pasikliovimo tikimybės intervalas yra [0,9531; 0,9993]. Pirmos rūšies klaida buvo padaryta 2,61 %, antros rūšies klaida – 6,08 %.

Kriterijaus statistikos *u* didžiausia ir mažiausia reikšmės pavaizduotos grafike:



5 pav. Kriterijaus statistikos reikšmės

Pasinaudojus dviejų gretimo ilgio fragmentų dažnio sutapimo kriterijaus statistika, įvertinta prielaida – ar dažnas posekis priklauso fragmentui?



6 pav. Tikimybinių charakteristikų pasikeitimas

Įvertinus prielaidą, nustatyta, kad dažno posekio dažnis apytiksliai lygus fragmento dažniui. Dažno posekio ilgis buvo nustatomas didžiausio tikėtinumo metodu.

Išvados

1. Stochastinis dažnų posekių algoritmas tinkamas didelėms duomenų bazėms analizuoti.
2. Algoritmas, remiantis atsitiktinai paimtų posekių analize, pateikia išvadas apie dažnus posekius.
3. Įvertinus tikėtinumo funkcijos pasikeitimą bei prielaidų tikimybes, nustatyta, kad dažno posekio dažnis apytiksliai lygus paslėpto fragmento dažniui.

Literatūra

1. Agrawal R., Srikant R., 1994, Fast *algorithms for mining association rules in large databases.* Proceedings of the 20th International Conference on Very Large Data Bases.
2. Agrawal R., Srikant R., 1997, Mining Sequential Patterns. *Journal Intelligent Systems*. Vol. 9. No. 1. P. 33–56.
3. Čekanavičius V., Murauskas G., 2000, *Statistika ir jos taikymai*. I tomas. Vilnius: TEV.
4. Huanyin Z., Jinsheng L., 2009, The Research of A-Priori Algorithm Candidates Based on Support Counts. *International Conference on Information Technology and Computer Science*. Vol. 1. P. 192–195.
5. Srikant R., Agrawal R., 1996, Mining Sequential Patterns: Generalizations and Performance Improvements. *In Proc. Int'l Conf Extending Database Technology*. Springer. P. 3–17.
6. Toivonen H., 1996, Sampling Large Databases for Association Rules. *Proceedings of the 22nd International Conference on Very Large Databases*. India: Mumbai. P. 134–145.

STOCHASTIC ALGORITHM FOR MINING FREQUENT SUBSEQUENCES

Loreta Savulioniene, Leonidas Sakalauskas

Summary

The article introduces a stochastic algorithm for mining frequent subsequences, which randomly separates subsequences of different length when a database is being scanned. The distribution of the length of subsequences depends on a geometric law with parameter p and the distribution of the distance between chosen subsequences also depends on a geometric law with parameter q. The designed algorithm was tested using computer modelling including methods of statistical hypothesis testing, probability confidence limits, likelihood functions and Monte Carlo for finding frequent subsequences. This algorithm is approximate, but it enables combining two important criteria (time and accuracy) and choosing values of parameters p and q accordingly. Using analysis of random subsequences, the algorithm gives statistical conclusions about frequent subsequences. Therefore, the designed stochastic algorithm for mining frequent subsequences can be used for search for frequent subsequences in large databases.

Keywords: subsequence, candidate sequence, frequent subsequence, first class error, second class error, confidence interval, criterion statistics, Apriori algorithm, stochastic frequent subsequence mining, fragment, length of a

STOCHASTINIS DAŽNŲ POSEKIŲ PAIEŠKOS ALGORITMAS

Loreta Savulionienė, Leonidas Sakalauskas

Santrauka

Straipsnyje pristatytas stochastinis dažnų posekių paieškos algoritmas, kuris, skenuojant duomenų bazę, atsitiktinai atrenka įvairiaus ilgio posekius. Posekių ilgis pasiskirstęs pagal geometrinį dėsnį su parametru p, o atstumas tarp pasirenkamų posekių taip pat pasiskirstęs pagal geometrinį dėsnį su parametru q. Sukurtas algoritmas buvo testuojamas kompiuterinio modeliavimo būdu, pritaikant dažnam posekiui nustatyti statistinių hipotezių tikrinimo, tikimybės pasikliovimo rėžių, tikėtinumo funkcijos ir Monte Karlo metodus. Šis algoritmas yra apytikslis, tačiau leidžia suderinti du svarbius kriterijus, t. y. laiką ir tikslumą, atitinkamai parenkant parametrų p ir q reikšmes. Algoritmas, pasinaudojus atsitiktinai paimtų posekių analize, pateikia statistines išvadas apie dažnus posekius. Taigi, sukurtas stochastinis dažnų posekių algoritmas gali būti taikomas dažnų posekių paieškai didelėse duomenų bazėse.

Prasminiai žodžiai: posekis, seka kandidatė, dažnas posekis, pirmos rūšies klaida, antros rūšies klaida, pasikliautinumo intervalas; kriterijaus statistika, Apriori algoritmas, stochastinis dažnų posekių paieškos algoritmas, fragmentas, posekio ilgis.

Įteikta 2011-11-16