VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Baigiamasis magistro darbas

Duomenų grupavimo taikymai transporto sistemose

Atliko: mag. 2 kurso, 1 grupės studentas
Marius Penikas

Darbo vadovas:
Asist. Alminas Čivilis

Vilnius, 2006

Turinys

Anotacija

Transporto srautų problemos šiuolaikiniuose didmiesčiuose kasdien darosi vis aktualesnės, todėl vis daugiau dėmesio skiriama sudėtingų transporto sistemų kūrimui, vis daugiau gilinamasi į transporto sistemose sukauptų duomenų tyrimą bei analizę. Kadangi tokios sistemos turi greitai apdoroti milžinišką ir nuolatos augantį duomenų kiekį iškyla duomenų minimizavimo problema. Problemos sprendimas - duomenų pertekliaus mažinimas grupuojant duomenis į didesnes grupes, pagal tam tikrus bendrus požymius. Šiame darbe pateikiama grupavimo algoritmų klasių analizė, jų tyrimas ir pritaikymas transporto sistemų duomenų apdorojimui. Taip pat pateikiama eksperimentinių bandymų metu gautų rezultatų suvestinė analizė.

Summary

Traffic flow control problems is getting more and more complicated and it is impossible to stop this growth. If we can not stop this process, we want to control it, so traffic analysis systems are getting more and more attention.

The object of investigation of this paper is data clustering and adjustment of data clustering algorithms to traffic flow control systems. The main goal is to analyze which class of clustering algorithms can perform better with specific traffic data, how much of this data is enough to forecast precise results, how much can we minimize and reduce our data and still get correct results.

To achieve these goals a new traffic database was created and two of a four types of clustering algorithms (K-means and Fuzzy C-means) were implemented and investigated with different amount of data. Algorithms were investigated two ways: time based traffic data clustering for a rush hour estimation and time and speed based traffic data clustering for the fastest route estimation.

In the first part of this paper clustering algorithms and the most important clustering problems are introduced. Second part or this work is reserved for all experiments which were done to achieve the goal determined for this work. Third section presents all the programming work which was done for the successful data analysis and algorithm testing.

Įvadas

Pastaraisiais dešimtmečiais informacinių transporto sistemų paklausa tapo labai didelė. Šalys nebepajėgia susidoroti su vis didėjančiais transporto srautais, transporto priemonių skaičiumi. Šio proceso neįmanoma sustabdyti ar bent kiek pristabdyti, nes dažnu atveju jis susijęs su visos šalies ekonomika, įvaizdžiu, patrauklumu ir kitais dalykais. Negalint šių procesų sustabdyti yra siekiama juos valdyti, tačiau tai padaryti nėra taip lengva, automatizuotos transporto informacinės sistemos susiduria su aibe įvairių problemų.

Viena pagringinių transporto informacinių sistemų problemų yra duomenų pertekliaus minimizavimas. Transporto informacinės sistemos privalo greitai apdoroti milžiniškus ir vis didėjančius duomenų kiekius. Kadangi skirtingiems tikslams pasiekti ar skirtingoms išvadoms padaryti reikalingų duomenų kiekis kartais gali skirtis kelias dešimtis ar net kelis šimtus kartų jį optimizavus pavyktų sutaupyti daug laiko ir resursų. Tam tikrais atvejais per didelis duomenų kiekis gali būti kritinis transporto informacinės sistemos veiksnys (pvz.: dėl techninių įrangos galimybių pajėgumų stokos, duomenų perdavimo kanalų spartos). Informacinei sitemai, kuriai keliamas uždavinys nustatyti transporto piko ir ne piko valandas ir informacinei sistemai skirtai greičiausio maršruto tam tikru laiku, tam tikroje vietoje nustatymui pateikiamų duomenų kiekis negali būti vienodas. Piko valandoms nustatyti reikės žymiai mažiau informacijos nei greičiausio maršruto paieškoms. Siekiant mažinti duomenų kiekį, neprarandant svarbios informacijos yra naudojamas duomenų grupavimas – objektų priskyrimas tam tikrom grupėm pabal bendrus požymius.

Šio darbo tikslas – išanalizuoti ir įvertinti grupavimo algoritmų klases, jų tipinius atstovus bei jų pritaikymą transporto sistemų duomenų grupavimui. Siekiant užsibrėžtų tikslų pristatomos grupavimo algoritmų klasės, jas atstovaujantys algoritmai, pristatoma informacinės sistemos dalis su realizuotais duomenų grupavimo algoritmais. Algoritmai vertinami 2 aspektais: dirbant su vienmačiais ir dvimačiais duomenimis, nes daugumoje transporto sistemų tiek duomenų pjūvių pilnai pakanka sistemos analizei atlikti. Vienmačiu atveju kelių apkrovoms nustatyti duomenys grupuojami tik pagal laiką, dvimačiu atveju kelių “kamščiams” bei realiems greičiams (greičiausiam atstumui skaičiuoti) nustatyti duomenys grupuojami pagal dvi komponentes - laiką ir greitį. Tyrimui panaudoti realūs, ne dirbtinai sugeneruoti duomenys ir sumodeliuotos situacijos, kad tiksliau įvertinti algoritmų galimybes, privalumus ir trūkumus realiame gyvenime. Galiausiai atlikus algoritmų analizę pateikiamos darbo išvados ir siūlymai.

1. Duomenų grupavimas

1.1. Duomenų grupavimo tikslai ir uždaviniai

Pačiu paprasčiausiu atveju duomenų grupavimas (*data clustering*) – tai objektų priskyrimas grupėms, kurių nariai yra “panašūs”. Grupė (*cluster*) yra objektų rinkinys, kurie yra “panašūs” tarpusavyje, bet “skiriasi” nuo objektų priklausančių kitoms grupėms.

Pagrindinis grupavimo tikslas yra – nesugrupuotų duomenų vidinis grupavimas, tačiau nėra jokio absoliučiai geriausio kriterijaus, kuris būtų nepriklausomas nuo galutinio rūšiavimo tikslo. Grupavimo kriterijų ir siekiamą galutinį tikslą privalo nustatyti pats vartotojas. Taigi, grupavimo esmė yra sugrupuoti duomenis į dideles panašias grupes, siekiant sumažinti duomenų kiekį.

Grupavimo algoritmai gali būti pritaikyti daugeliui uždavinių spręsti:

- Biologija: augalų ir gyvūnų grupavimas pagal tam tikrus požymius;
- Biblioteka: knygų grupavimas;
- Miestų planavimas: pastatų grupių nustatymas priklausomai nuo pastato tipo, panaudojimo, buvimo vietos;
- Kelių planavimas: kelių planavimas siekiant išvengti transporto kamščių;
- Kiti uždaviniai.

1.2. Grupavimo algoritmams keliami reikalavimai ir problemos

Pagrindiniai reikalavimai grupavimo algoritmui:

- Keičiamumas;
- Gebėjimas operuoti skirtingų tipų objektais;
- Gebėjimas aprašyti apibrėžtos formos grupes;
- Nepriklausomumas nuo pradinių duomenų;
- Gebėjimas susitvarkyti su triukšmu;
- Nepriklausomumas nuo pradinių duomenų tvarkos;
- Keliamačių duomenų palaikymas;
- Aiškumas ir panaudojamumas.

Pagrindinės grupavimo algoritmų problemos:

- Konkretus grupavimo algoritmas negali vienodai išspręsti visų jam keliamų reikalavimų;

- Darbas su kelių dimensijų duomenimis ir dideliu duomenų kiekiu gali būti “brangus” laiko atžvilgiu;
- Kartais gali būti sunku įvertinti atstumą tarp kelių keliamačių objektų;
- Grupavimo algoritmo rezultatas gali būti suprantamas skirtingais būdais;
- Kitos problemos.

1.3. Grupavimo algoritmų skirstymas

Grupavimas gali būti skirstomas į:

- Abstraktų grupavimą (*conceptual*);
- Grupavimą atstumo pagrindu (*distance-based*).

Du ar daugiau objektų priklauso tai pačiai grupei jei jie yra “arti” vienas kito duotu atstumu. Toks grupavimas yra vadinamas grupavimu atstumo pagrindu. Kai du ar daugiau objektų priklauso tai pačiai grupei jei jie turi bendrų savybių, būdingų visiems tos grupės objektams, tai toks grupavimas yra vadinamas abstrakčiuoju.

Grupavimo algoritmai taip pat gali būti:

- Hierarchiniai (*Hierarchical*);
- Nehierarchiniai(*Non-Hierarchical*);

Tokiu atveju nehierarchiniai algoritmai yra skirstomi į:

- Sutampančius (*Overlapping*);
- Tikimybinius (*Probabilistic*);
- Išskirtinius (*Exclusive*).

Išskirtiniai algoritmai klasifikuoja duomenis į grupes taip, kad vienas objektas gali priklausyti tik vienai konkrečiai grupei. Sutampantys algoritmai objektų grupavimui naudoja neapibrėžtas (*fuzzy*) aibes, kas leidžia kiekvienam objektui priklausyti dviem ar daugiau grupių. Tokiu atveju objektui kiekvienoje grupėje yra nurodomas priklausomybės laipsnis. Hierarchiniai algoritmai priklauso nuo dviejų artimiausių grupių sąjungos, todėl pradžioje kiekvienas objektas yra priskiriamas atskirai grupei, o po kelių iteracijų objektai yra suklasifikuojami į skirtingas grupes. Tikimybiniai algoritmai klasifikuoja objektus į grupes pagal tikimybę.

2. Pagrindinių grupavimo algoritmų apžvalga

2.1. K-means algoritmas

K-means (*MacQueen, 1967*) yra klasikinis išskirtinių (*exclusive*) grupavimo algoritmų pavyzdys. Šis algoritmas pateikia labai paprastą ir aiškų būdą, kaip suklasifikuoti objektų aibę į apibrėžtą grupių skaičių.

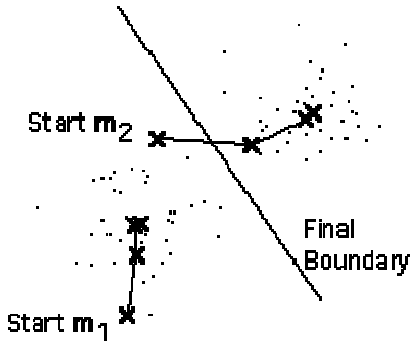
Algoritmas. Iš pradžių reikia apsibrėžti k centrinių taškų, po vieną kiekvienai grupei. Centrinių taškų teisingas parinkimas yra labai svarbus, nes nuo to priklausys galutinis grupavimo rezultatas. Todėl rekomenduojama juos išdėstyti kiek galima didesniu atstumu vienas nuo kito. Kitas žingsnis yra paimti visus objektų aibės taškus ir kiekvieną iš jų priskirti artimiausiam centriniam taškui. Kai nelieka nepriskirtų aibės taškų, tai reiškia jog pradinis grupavimas yra baigtas. Toliau mes turime imti kiekvieną objektų grupę ir iš ją sudarančių objektų paskaičiuoti naują šios grupės centrinį tašką. Matome, kad susidarė ciklas. Taip darbą tęsiame toliau, kol po kažkiek iteracijų jokių naujų pakitimų nebėra atliekama. Tai reiškia, jog grupavimas yra baigtas.

Taigi šis algoritmas minimizuoja tikslo funkciją, kurią mes galime užrašyti sekančiai:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

,

Šiuo atveju ši tikslo funkcija yra vadinama kvadratinės klaidos funkcija (*squared error*). $\left\| x_i^{(j)} - c_j \right\|^2$ yra pasiriktas atstumas tarp objekto $x_i^{(j)}$ ir grupės centro c_j .



Pav. 1 K-means grupavimas

Algoritmo schema:

1. Padėti K taškų klasifikuojamų objektų erdvėje. Šie taškai simbolizuos klasifikuojamų objektų grupių (*clusters*) centrus.

2. Kiekvieną objektą priskirti grupei, iki kurios centinio taško yra mažiausias atstumas.
3. Kai visi objektai suskirstyti į grupes, perskaičiuoti centrinių taškų pozicijas.
4. Kartoti 2 ir 3 žingsnius, kol centrinių taškų pozicijos išliks stabilios.

Išvados. Nors galima įrodyti, kad ši procedūra visada pasibaigs, t.y. tikslo funkcija – visada konverguos, algoritmas nebūtinai optimaliai suklasifikuos objektus į grupes taip, kad tai atitiktų tikslo funkcijos globalų minimumą. Algoritmas taip pat yra labai jautrus centrinių grupių taškų pradiniam parinkimui. Todėl, kad pasiekti gerų rezultatų gali prireikti ši algoritmą pakartoti kelis kartus ir su skirtingais pradiniais grupių centrais.

Pastabos. K-means yra pakankamai “brangus” algoritmas. Šiam algoritmui trūksta tikslumo, todėl siekiant gerų rezultatų reikia jį “prasukti” kelis kartus. Praktinėje algoritmo realizacijoje reikia išskirti atvejį, kai po grupavimo kažkuri grupė gali likti tuščia. Rezultatai priklauso nuo to į kiek grupių mes klasifikuojame savo objektus, taip pat kokią metriką mes naudojame atstumo matavimui.

2.2. Fuzzy C-means algoritmas

Fuzzy C-means (Dunn, 1973; Bezdek 1981) yra klasikinis sutampančių (*overlapping*) grupavimo algoritmų pavyzdys, kurie leidžia objektams priklausyti kelioms grupėms. Šie algoritmai dažniausiai yra naudojami šablonų (*pattern*) atpažinimui.

Algoritmas. Šis algoritmas minimizuoja tikslo funkciją, kurią mes galime užrašyti sekančiai:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m}\left\|x_i - c_j\right\|^2$$, kur $1 \le m \ge \infty$

$u_{ij} – x_i$ objekto priklausomybės laipsnis grupėje j.

c_j – grupės centras.

|| * || - reikalingas įvertinti panašumą, tarp grupės centro ir išmatuotų duomenų.

Algoritmas realizuojamas iteraciniu tikslo funkcijos minimizavimu, kiekvieną kartą atnaujinant priklausomybės laipsnį u_{ij} (skaičiuojama narystės funkcija) ir grupės centrą c_j .

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\left\|x_i - c_j\right\|}{\left\|x_i - c_k\right\|}\right)^{\frac{2}{m-1}}} \quad , \quad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

$$\max_{ij}\left\{\left|u_{ij}^{(k+1)}-u_{ij}^{(k)}\right|\right\}<\varepsilon$$

Šis maksimumas realizuoja algoritmo pabaigos sąlygą. Čia $0 \le \varepsilon \le 1$ – pabaigos kriterijus. Tuo tarpu k žymi iteracijos numerį. Ši procedūra leidžia rasti lokalų tikslo funkcijos minimumą.

Algoritmo schema:

1. Inicializuoti $U=[u_{ij}]$ matricą, $U^{(0)}$.
2. K žingsnyje: suskaičiuoti centro vektorius $C^{(k)}=[c_j]$ su $U^{(k)}$

$$c_j=\frac{\sum_{i=1}^{N}u_{ij}^{m}\cdot x_i}{\sum_{i=1}^{N}u_{ij}^{m}}$$

3. Atnaujinti $U^{(k)}$, $U^{(k+1)}$

$$u_{ij}=\frac{1}{\sum_{k=1}^{C}\left(\frac{\|x_i-c_j\|}{\|x_i-c_k\|}\right)^{\frac{2}{m-1}}}$$

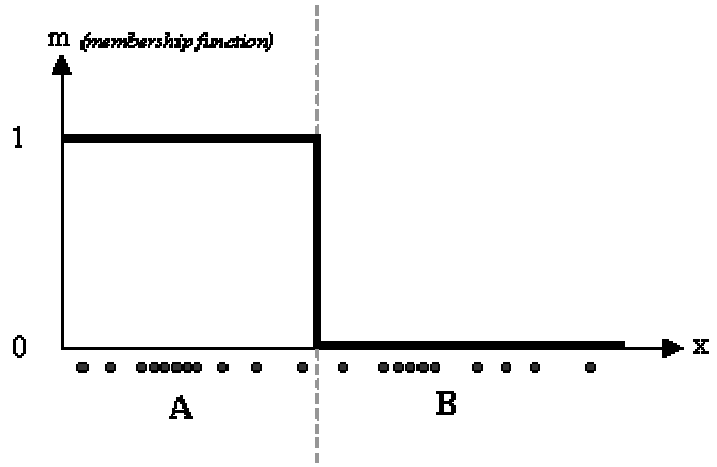
4. Jei $||U^{(k+1)}-U^{(k)}||<\varepsilon$ pabaigos sąlyga tenkinama – baigti, jei ne - grįžti į 2 punktą.

Pastabos. Objektų priklausymui vienai ar kitai grupei nustatyti yra skaičiuojama narystės funkcija, kuri ir lemia šio algoritmo neapibrėžtą elgesį. Tam sudaroma matrica U, kurios elementai priklauso intervalui [0,1] ir reiškia priklausomybės grupei laipsnį. Skirtingos pradinės algoritmo inicializacijos dažniausiai duoda tą patį galutinį rezultatą, tačiau gali pareikalauti daugiau iteracijų konvergavimui.

Pavyzdys. Pavaizduokime vienmatį grupavimo pavyzdį. Tarkime mes turime tokią duomenų aibę, kurią galime pavaizduoti sekančiai:

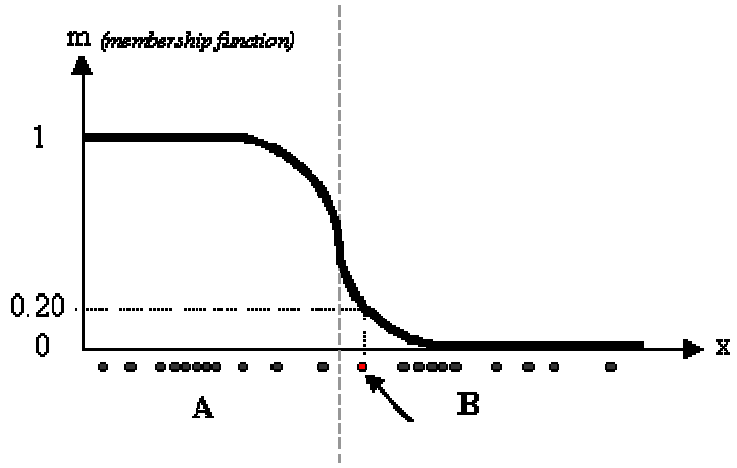
Pav. 2 Atsitiktinė vienmatė duomenų aibė

Žiūrėdami į šį vaizdą, galime nesunkiai išskirti du duomenų centrus. Jei šiuos duomenis suklasifikuotume K-means algoritmu gautume:



Pav. 3 Vienmatė duomenų aibė suklasifikuota K-means

O jei šiuos taškus suklasifikuotume Fuzzy C-means algoritmu gautume:



Pav. 4 Vienmatė duomenų aibė suklasifikuota Fuzzy C-means

Nuožulnus perėjimas iš grupės A į grupę B parodo, kad kai kurie taškai dabar priklauso abiem šiom grupėms, priešingai nei K-means algoritme. Taigi, galime daryti išvadą, kad Fuzzy C-means algoritmas nėra toks “griežtas” kaip K-means. Raudonai pažymėtas taškas parodo priklausomybės laipsnį tarp A ir B grupių.

Jei iš šių grafinių pavaizdavimų, pabandytume sudaryti narystės funkcijas atitinkančias matricas U vienam ir kitam atvejams gautume, kad :

$$U_{MC} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ .. & .. \\ 0 & 1 \end{bmatrix} \qquad U_{MC} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ .. & .. \\ 0.9 & 0.1 \end{bmatrix}$$

,

kur matrica kairėje atitinka K-means grupavimą, o dešinėje Fuzzy C-means grupavimą. Stulpelių skaičius priklauso nuo to į kiek grupių mes klasifikuosime duomenis, o eilučių skaičius priklauso nuo duomenų skaičiaus. Matome, kad K-means atveju koeficientai yra sveikieji skaičiai, o Fuzzy C-means atveju – realieji skaičiai. Būtent tai ir leidžia padaryti ne tokį "griežtą" atskyrimą tarp dviejų suklasifikuotų grupių.

2.3. Hierarchinio grupavimo algoritmas

Hierarchinis grupavimas (S.C.Johnson 1967) pagrįstas tuo, kad mes turime N duomenų aibę ir N*N atstumų (arba panašumų) matricą.

Algoritmas. Iš pradžių reikia priskirti kiekvieną aibės objektą atskirai grupei, taigi jei mes turime N objektų, mes turime ir N grupių po vieną objektą. Tarkim atstumai tarp grupių yra lygūs atstumams tarp objektų tose grupėse. Tada randame du artimiausius objektus (grupes) ir juos sujungiame į vieną grupę, taigi liks viena grupe mažiau. Randame atstumus tarp naujos grupės ir visų likusių senų grupių. Tada reikia kartoti 2 ir 3 žingsnius tol, kol liks tik viena objektų grupė.

Toks metodas vadinamas sulipimo metodu. Yra ir kitas priešingas metodas, kuris vadinamas dalomuoju metodu. Ten grupavimas atliekamas iš pradžių turint vieną didelę grupę ir toliau ją skaidant į mažesnes. Kadangi dalomieji metodai nėra tokie populiarūs, daugiau dėmesio skirsime sulipimo metodui.

Algoritmo schema (naudojant pavienio ryšio atstumo paieškos metodą)**:**

D = [d(i,j)] – atstumų matrica N*N.

Grupavimui priskiriami eilės numeriai 0, 1, 2, …, (n-1), o L(k) – grupavimo lygis.

1. L(0) ir eilės numeriui m priskirti 0. Kiekvieną objektą priskirti atskirai grupei.
2. Rasti trumpiausią atstumą tarp dviejų grupių, tarkim (r) ir (s).
   $d[(r),(s)] = min\ d[(i),(j)]$ minimumas skaičiuojamas tarp visų grupių
3. m = m + 1, sujungti (r) ir (s) grupes į vieną grupę, kad suformuoti sekančią suklasifikuotą m grupę. Priskiriame nauja grupavimo lygį:
   $L(m) = d[(r),(s)]$
4. Atnaujiname atstumų matricą D.
   $d[(k), (r,s)] = min\ d[(k),(r)], d[(k),(s)]$
5. Jei visi objektai priklauso vienai grupei, - baigti, jei ne - grįžti į 2 punktą.

Išvados. Toks algoritmas turi porą silpnybių:

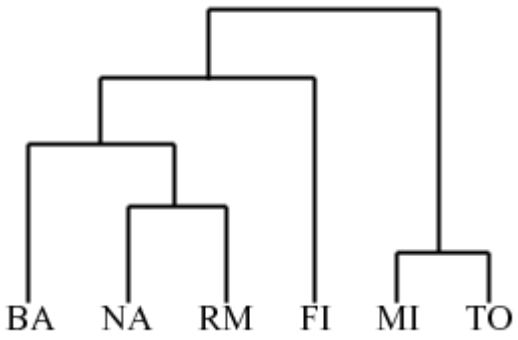
- Sudėtingumas ne mažesnis nei $O(n^2)$, kai n – objektų skaičius
- Negalima atstatyti prieš tai buvusių veiksmų

Pavyzdys. Tarkime turime tokią pradinę atstumų tarp miestų matricą. O grupavimo lygis L = 0:

	BA	**FI**	**MI**	**NA**	**RM**	**TO**
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

Pav. 5 Pradinė atstumų matrica

Tada sugrupavę turime:



Pav. 6 Hierarchinis grupavimas

3. Kitos su duomenų grupavimu susijusios problemos

3.1. Atstumo matavimas tarp grupės objektų

Kaip jau minėjome jungiant objektus į grupes grupavimo algoritmai skaičiuoja atstumus tarp objektų arba ieško apsibrėžtų panašumų ar skirtumų. Panašumus/skirtumus galime apibrėžti kaip taisyklių rinkinį, kuris mums leidžia išskirti objektus į grupes pagal joms būdingus, taisyklėse aprašytus požymius. Tiek atstumai tiek ir panašumai gali būti kelių dimensijų, kur kiekviena dimesija aprašo atskirą grupavimo taisyklę ar sąlyga. Pavyzdžiui jei mes turime maisto grupę, tai mes galime atkreipti dėmesį į to maisto kokybę, kainą, skonį, kalorijų kiekį ir pan. Taigi tiek kaina, tiek kokybė, tiek skonis yra atskiros dimensijos.

Galime išskirti kelis populiariausius metodus atstumams matuoti:

- Euklido atstumas (Euclidean distance);
- Kvadratinis Euklido atstumas (Squared Euclidean distance);
- Manheteno atstumas (Manhattan distance);
- Čebyševo atstumas (Chebychev distance);
- Svorių atstumas (Power distance);
- Procentinis nesutapimas (Percent disagreement).

Euklido atstumas. Bene pats patogiausias, geriausias ir populiariausias metodas atstumams tarp objektų matuoti yra Euklido atstumas. Euklido atstumas – tai geometrinis atstumas tarp objektų keliamatėje erdvėje. Euklido atstumas yra skaičiuojamas:

$$d(x,y) = \{\Sigma_i (x_i - y_i)^2 \}^{½}$$

Reikia atkreipti dėmesį, kad Euklido algoritmas dažniausiai skaičiuojamas neapdorotiems duomenims. Šio metodo privalumas yra tai, kad naujo objekto atsiradimas niekaip nepaveikia atstumo tarp kitų jau buvusių objektų. Tačiau šis metodas yra jautrus dimensijų skirtumams, todėl naudojant ši metodą reikėtų operuoti vienodomis dimensijomis siekiant gauti teisingus rezultatus.

Kvadratinis Euklido atstumas. Kvadratinis Euklido metodas naudojamas yra labai panašus į Euklido atstumo matavimo metodą, tik jis yra naudojamas kai norima toliau esantiems objektams suteikti didesnius svorius ir taip labiau atskirti atskiras grupes. Kvadratinis Euklido atstumas yra skaičiuojamas:

$$d(x,y) = \Sigma_i (x_i - y_i)^2$$

Manheteno atstumas. Šis atstumas išreiškiamas matuojant dimensijų skirtumų sumą. Iš esmės jis yra labai panašus į Euklido atstumą.

$$d(x,y) = \Sigma_i |x_i - y_i|$$

Čebyševo atstumas. Šis atstumo matavimo būdas dažniausia aptinkamas tada, kai norima atskirti du objektus, jei jie skiriasi bent viena dimensija. Čebyševo atstumas išreiškiamas tokia formule:

$$d(x,y) = \text{Maksimumas}|x_i - y_i|$$

Svorių atstumas. Kartais mes norime sumažinti ar padidinti svorį tų dimensijų, kur objektai yra labai skirtingi. Tai yra atliekama skaičiuojant svorių atstumą, kuris išreiškiamas formule:

$$d(x,y) = (\Sigma_i |x_i - y_i|^p)^{1/r}$$

kur r ir p yra vartotojo apibrėžti parametrai. Parametras p išreiškia svorį ant atskirų objektų dimensijų skirtumų, o r išreiškia svorį ant didesnių skirtumų tarp objektų. Jei r ir p lygūs 2, tai šis atstumas yra lygus Euklido atstumui.

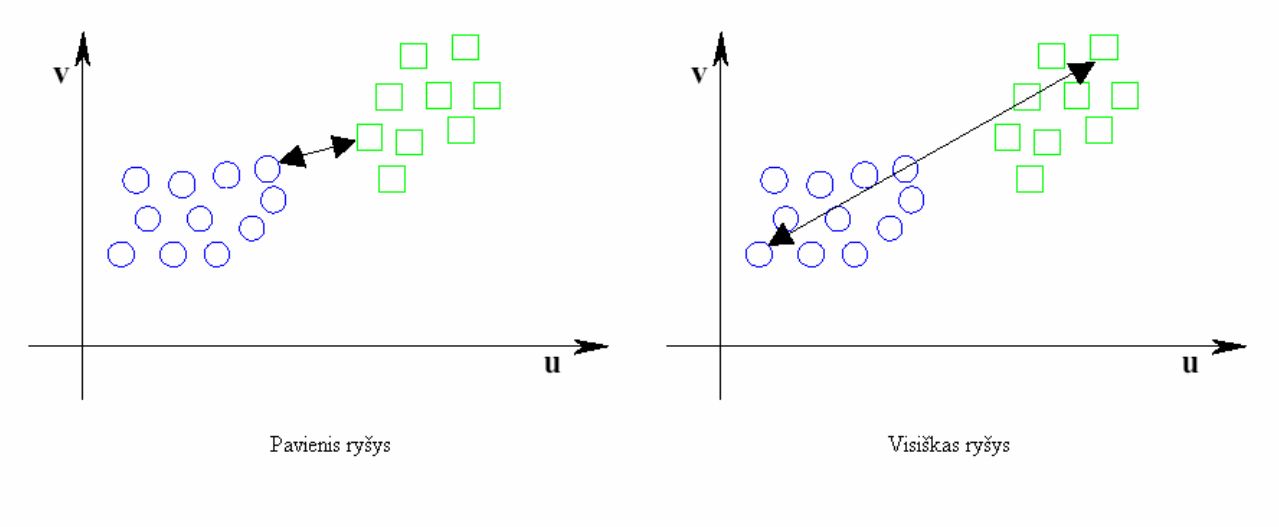
Procentinis nesutapimas. Šis atstumo skaičiavimo metodas yra naudojamas, jei dimensijų duomenys gamtoje yra besąlyginiai. Atstumas yra skaičiuojamas:

$$d(x,y) = (\text{Suma } x_i \neq y_i)/ i$$

3.2. Grupių jungimo metodai

Kol kiekvienas objektas egzistuoja atskirai, tol atstumus tarp objektų gali išmatuoti labai paprastais būdais, tačiau jei objektai sudaro grupes, išmatuoti atstumus tarp grupių jau nebėra taip paprasta . Šiems atstumams matuoti yra sukurti keli metodai. Populiariausi būdai yra:

- Pavienio ryšio (*single-linkage, nearest neighbor*)
- Visiško ryšio (*complete-linkage, furthest neighbor*)
- Vidutinio ryšio (*average-linkage, UPGMA*)
- Centrinio ryšio (*UPGMC*)



Pav. 7 Ryšių pavyzdžiai

Pavienis ryšys. Pavienis ryšys dar kartais vadinamas minimumo metodu. Atstumas tarp grupių yra skaičiuojamas kaip trumpiausias atstumas nuo bet kurio vienos grupės objekto iki bet kurio kitos grupės objekto.

Visiškas ryšys. Visiškas ryšys dar yra vadinamas maksimumo metodu. Tokiu atveju atstumas tarp dviejų grupių yra skaičiuojamas kaip ilgiausias atstumas nuo bet kurio vienos grupės objekto iki bet kurio kitos grupės objekto.

Vidutinis ryšys. Vidutinio ryšio būdu atstumas tarp dviejų grupių yra skaičiuojamas kaip visų atstumų tarp vienos grupės objektų ir kitos grupės objektų vidurkis.

Centrinis ryšys. Centrinio ryšio būdu atstumas tarp dviejų grupių yra skaičiuojamas kaip atstumas tarp dviejų grupių centro taškų. Grupių centro taškai suprantami, kaip grupės gravitacijos centrai.

4. Grupavimo algoritmų tyrimas

Grupavimo algoritmų tyrimui buvo sukurta informacinės sistemos duomenų bazės dalis, su pilnai realizuotais K-means ir Fuzzy C-means algoritmais, vienmačiam ir dvimačiam duomenų grupavimui. Ši darbo dalis bus aprašyta sekančiame skirsnyje.

Grupavimo algoritmai vertinami 2 aspektais: dirbant su vienmačiais ir dvimačiais duomenimis, nes daugumoje transporto sistemų tiek duomenų pjūvių pilnai pakanka sistemos analizei atlikti. Vienmačiu atveju kelių apkrovoms nustatyti duomenys gali būti grupuojami tik pagal laiką, dvimačiu atveju spūstims keliuose bei faktiniams greičiams (kad būtų imanoma rasti greičiausią kelionės maršrutą ir prognozuoti sugaištą laiką) nustatyti duomenys grupuojami pagal dvi komponentes - laiką ir greitį.

Vykdant šį tyrimą sieksime išsiaiškinti, kuris grupavimo algoritmas geresnis, efektyvesnis. Kaip didinant ar mažinant grupių skaičių keičiasi skaičiavimų iš sugrupuotų duomenų paklaidos. Kiek duomenų pakanka, kad pasiekti norimų rezultatų ir kaip nustatyti šią ribą.

4.1. Duomenys naudojami tyrimui

Kaip jau buvo minėta, duomenų bazei užpildyti panaudoti duomenys iš Aalborgo universitete (Danija) vykdyto “INFATI” projekto. “INFATI” projekto metu duomenys buvo renkami siekiant ištirti vairuotojų psichologinę reakciją į informaciją apie viršijamą greitį. Duomenys buvo gaunami kas 1 sekundę iš GPS imtuvų (Pav. 8) įmontuotų automobiliuose, laikotarpyje nuo 2000 m. gruodžio iki 2001 m. kovo.

Visi duomenys, galintys atskleisti vairuotojų tapatybę yra pašalinti: pašalinti duomenys 2 km spinduliu nuo vairuotojo namų, taip pat duomenys, galintys atskleisti projekte dalyvavusių vairuotojų darbo vietas, bei kita “jautri” informacija. Dėl to šio darbo metu vykdant kai kuriuos tiriamuosius darbus, dėl duomenų trūkumo (iškarpymų) nebuvo galima gauti tikslių rezultatų. Atvaizdavus tokius duomenis grafiškai matosi, kad grafikas eina per nulines reikšmes. Atliekant šį darbą tokias vietas ignoruosime.

Pav. 8 Automobiliuose sumontuoti GPS imtuvai

Duomenų bazėje iš viso yra apie 1,9 milijono įrašų, iš kurių šiam tyrimui mes naudosime tik mažą dalį duomenų (naudosime tik iš vieno automobilio gautus duomenis, dėl per mažų techninių resursų). Visi duomenys surinkti projekto dalyviams važinėjant Aalborgo miesto rajone, Danijoje. Šiais duomenimis užpildyta T_LOG lentelė. Iš INFATI projekto vykdytojų gauti tokie duomenys:

- automobilio identifikatorius;
- vairuotojo identifikatorius;
- data;
- laikas;
- x koordinatė;
- y koordinatė;
- su skaitmeninių žemėlapiu suderinta x koordinatė;
- su skaitmeninių žemėlapiu suderinta y koordinatė;
- palydovų kiekvienu laiko momentu užfiksavusių automobilį skaičius;
- maksimalus kelyje leidžiamas greitis;
- realus automobilio greitis;
- gatvės kodas.

Minėti duomenys ne komerciniams tikslams, moksliniams tyrimams yra pateikiami nemokamai projekto vykdytojų Internetinėje svetainėje[1]. Kai kurie projekto duomenys, kaip antai skaitmeninio žemėlapio koordinatės, nėra viešai skelbiami, todėl jų gauti nepavyko.

Duomenų importo į duomenų bazę skriptų išeities kodai pateikiami CD diske.

4.2. Tyrimo platforma

Grupavimo algoritmų tyrimui buvo panaudota sekanti platforma:

- Procesorius: Intel Pentium Centrino -1,6 GHz;
- RAM: 512 Mb;
- HDD: 40 Gb;
- Operacinė sistema: Windows Xp Professional;
- DBVS: Oracle 9i, Standart edition;
- Duomenų analizės priemonės: Microsoft Exel 2000.

[1] *http://www.cs.auc.dk/TimeCenter/software.htm*

4.3. Nagrinėjimo objektai

Kaip jau minėjome toliau bus tiriami du grupavimo algoritmai, kaip tipiniai savo algoritmų klasių atstovai: K-means ir Fuzzy C-means. Algoritmai bus tiriami dviem aspektais:

- grupuojant vienmačius duomenis;
- grupuojant dvimačius duomenis.

Pirmuoju (vienmačių duomenų grupavimo) atveju duomenis skirstysime į grupes pagal laiko komponentę – tai leis apskaičiuoti kelių apkrovas tam tikrais paros laiko tarpais bei nustatyti transporto spūstis (kelių eismo piko valandas). Kelių apkrovas tam tikru laikotarpiu galėsime nustatyti atvaizdavę vidutinio greičio tam tikru paros laikotarpiu grafikus bei atlikę maksimalaus leistino ir faktinio greičio tam tikrais laiko tarpais palyginimą. Šis tyrimas turėtų būti mažiau jautrus duomenų kiekio pokyčiams. O piko ir ne piko laikotarpius galėsime įvertinti nustatę grafikų minimumo ir maksimumo taškus.

Antruoju (dvimačių duomenų grupavimo) atveju duomenis skirstysime į grupes pagal 2 komponentes: laiko ir greičio. Toks duomenų grupavimas leis mums konkrečiais paros laiko tarpais tiksliai žinoti realų vidutinį greitį tam tikroje kelio atkarpoje ir bus galima nesunkiai nustatyti patį greičiausią ir optimaliausią kelią nuo taško A iki taško B. Toks grupavimas turėtų būti tikslesnis, bet tuo pačiu metu ir jautresnis duomenų kiekio pokyčiams bei pradinių grupių centrų parinkimui.

Atlikdami tyrimą pasistengsime įvertinti santykį tarp duomenų kiekio reikalingo apsibrėžto tikslumo prognozei ateityje atlikti ir paklaidos, kurią galime sau leisti laikyti nereikšminga ir neturinčia įtakos galutiniams rezultatams. Toks įvertinimas leis nustatyti priklausomybę tarp duomenų kiekio ir paklaidos. Atskirais atvejais (pvz.: piko valandų nustatymui) galima tikėtis pakankamai tikslios prognozės drastiškai sumažinus duomenų kiekį reikalingą jai atlikti.

Siekiant užsibrėžtų tikslų reikės išmatuoti kaip duomenų kiekis grupėse priklauso nuo grupių į kurias grupuojame turimus duomenis kiekio. Taip pat reikės išmatuoti paklaidą gautą tarp realių duomenų ir prognozuojamų duomenų. Tam naudosime apibendrintą metodą ir skaičiuosime skirtumą tarp realaus greičio tam tikrais laiko momentais nubrėžto grafiko ploto ir prognozuojamo greičio atitinkamais laiko momentais nubrėžto grafiko ploto padidinus atitinkamos komponentės ribas ir sumažinus grupių skaičių.

Apjungus šiuos grafikus nagrinėjamiems algoritmams bus galima įvertinti jų taikymo transporto sistemose galimybes, gerąsias ir blogąsias savybes, minimalius nagrinėjamų grupavimo komponenčių kiekius, prognozuojamų duomenų tikslumą, jų tinkamumą tikslioms ateities prognozėms daryti.

4.4. Tiriami algoritmai

Atsižvelgdami į tai, kad hierarchiniai grupavimo algoritmai turi anksčiau minėtų silpnybių ir nėra tinkami transporto sistemų duomenims grupuoti tirsime tik nehierarchinių algoritmų savybes. Kaip jau anksčiau teigėme, nehierarchiniai algoritmai yra skirstomi į:

- Sutampančius (*Overlapping*);
- Tikimybinius (*Probabilistic*);
- Išskirtinius (*Exclusive*).

Dėl tikimybinių grupavimo algoritmų sudėtingumo juos paliksime nuošalyje ir toliau nagrinėsime tik sutampančius ir išskirtinius algoritmus. Sutampantiesiems algoritmams atstovaus Fuzzy C-means algoritmas, o išskirtiniams K-means algoritmas. Juos pasirinkome, kaip tipinius ir geriausiai žinomus grupavimo algoritmus atstovaujančius šioms algoritmų klasėms. Šių algoritmų aprašymai pateikti atitinkamai 2.1. ir 2.2. poskyriuose.

Esminis skirtumas tarp šių algoritmų yra tas, kad K-means algoritmas objektus grupuoja į grupes taip, kad objektas gali priklausyti tik vienai grupei, o Fuzzy C-means algoritmas leidžia objektams priklausyti kelioms grupėms.

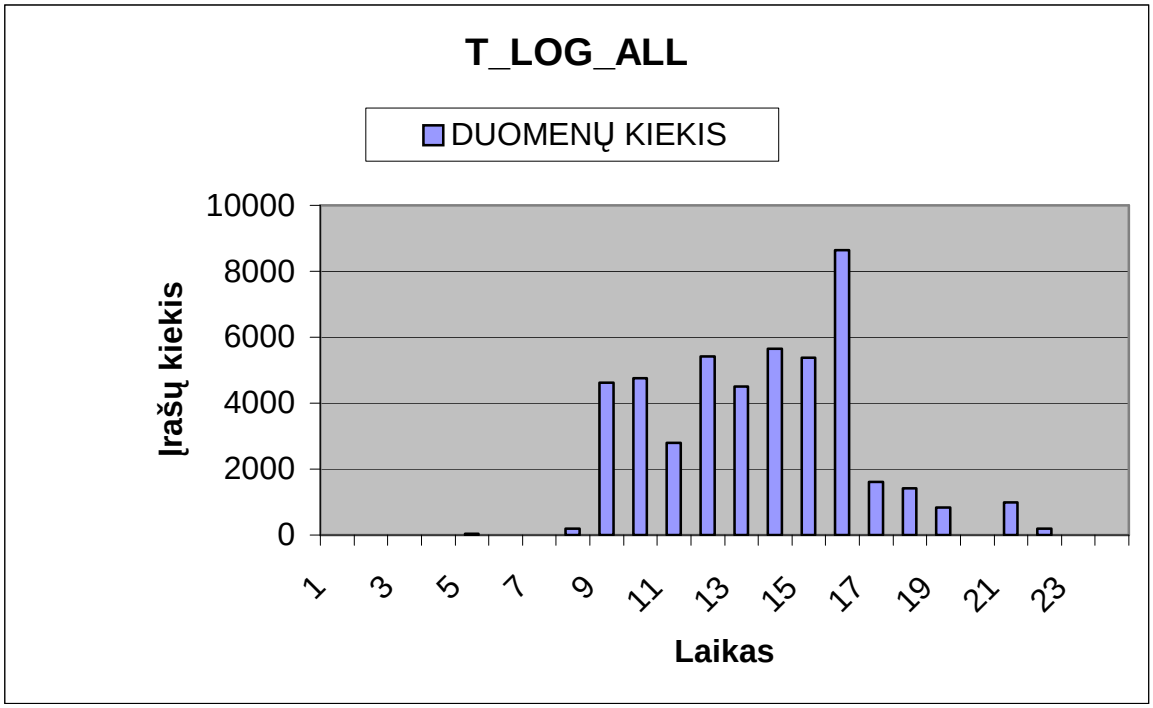
Atstumams tarp objektų matuoti naudosime Euklido atstumą, kurį pritaikysime vienmačiam ir dvimačiam atvejams, o atstumu tarp grupių laikysime atstumą tarp jų centrinių taškų.

4.5. Rezultatai gauti nenaudojant grupavimo metodų

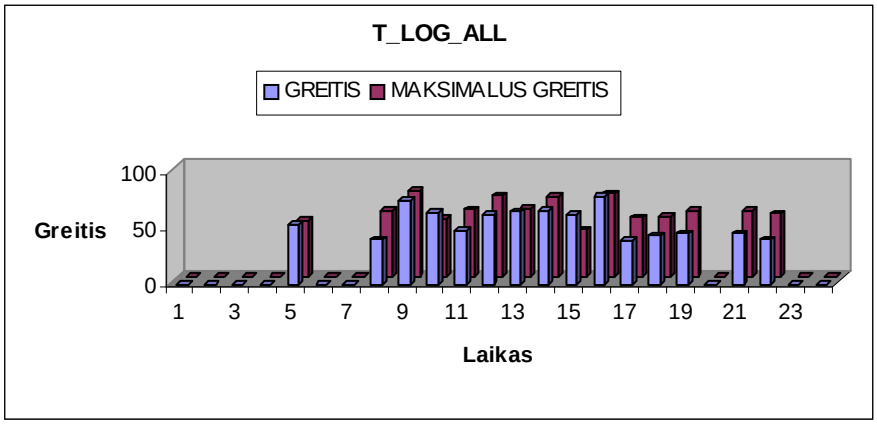
Šiame darbe daugiau aptarsime bandymus atliekamus su visa duomenų aibe, o informaciją apie savaitgalių ir darbo dienų duomenų poaibius galima rasti pridedamame CD diske. Šią informaciją nagrinėsime tik išimtiniais, “įdomiais” atvejais. Visa informacija, kiek tai yra galima bus pateikta grafiniame pavidale. Toliau pateiksime bendrą pradinę informaciją.

9 paveikslėlyje pateiktame grafike matome kiek kartų GPS imtuvas gavo duomenis iš palydovų tam tikrais laiko tarpais. Paprastumo dėlei ėmėme 1 valandos laiko intervalus. Iš grafiko nesunku nustatyti, kad didžiausias eismas yra 17 valandą, o didžiausios kelių apkrovos yra nuo 9 iki 20 valandų. Laiko tarpai, kada automobilis nevažinėjo specialiai palikti tušti.

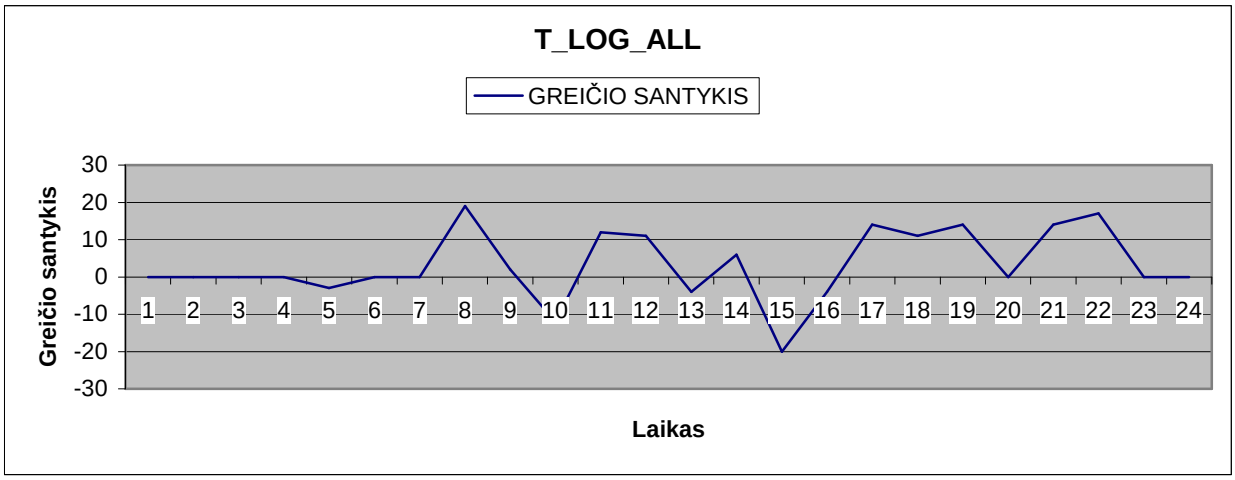
10 ir 11 paveikslėliuose pavaizduotuose grafikuose atsispindi realaus ir maksimalaus greičio santykiai tam tikrais paros laiko tarpais. 12 grafike greičiai išskaidyti į greičius darbo dienomis, savaitgaliais ir bendrus greičius visai savaitei. Iš šio grafiko nulinių greičio reikšmių galime spręsti kada automobilis visai nevažinėjo ir kiek bendro greičio skaičiavimui turi įtakos atskiros duomenų grupės.



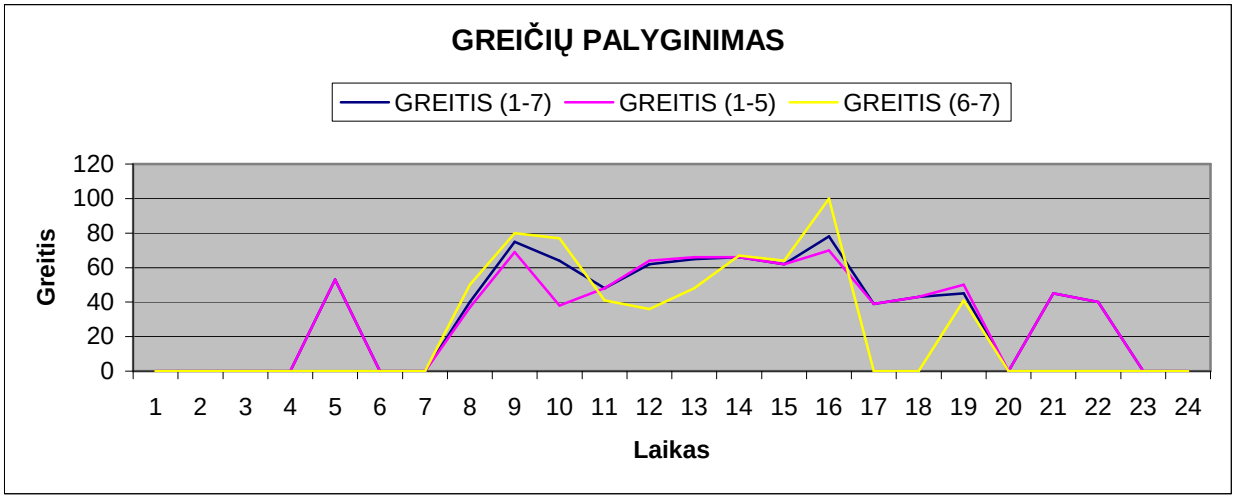
Pav. 9 Duomenų kiekis DB, kas 1 h



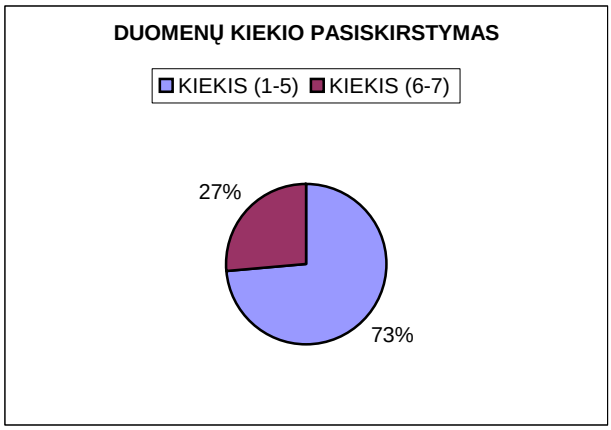
Pav. 10 Visų duomenų greičių ir maksimalių greičių palyginimas, kas 1 h



Pav. 11 Maksimalaus greičio ir realaus greičio skirtumai (maksimalus greitis - greitis), kas 1h



Pav. 12 Greičių bendro vidurkio, greičio savaitgaliais ir darbo dienomis palyginimas



Pav. 13 Duomenų kiekio pasiskirstymas tarp darbo dienų ir savaitgalių

Jei nagrinėsime tik laiko atkarpą tarp 8 ir 23 valandų (nes kitur mažoka duomenų) galime nesunkiai nustatyti, kad didžiausios kelių apkrovos yra 8, 10-12, 17-20 valandomis. Bet įvertinus duomenis pateiktus 9 grafike galime teigti, kad didžiausios spūstys keliuose yra apie 17 val., nes tuo laiku gauta daugiausia pranešimų ir bene didžiausias maksimalaus leistino ir realaus greičio santykis. Mažiausiai keliai apkrauti prieš ir iškart po spūsčių meto. Iš kitos pusės 12 grafikas rodo, kad savaitgaliais kelių apkrovos yra didžiausios ne vakare, bet apie vidurdienį.

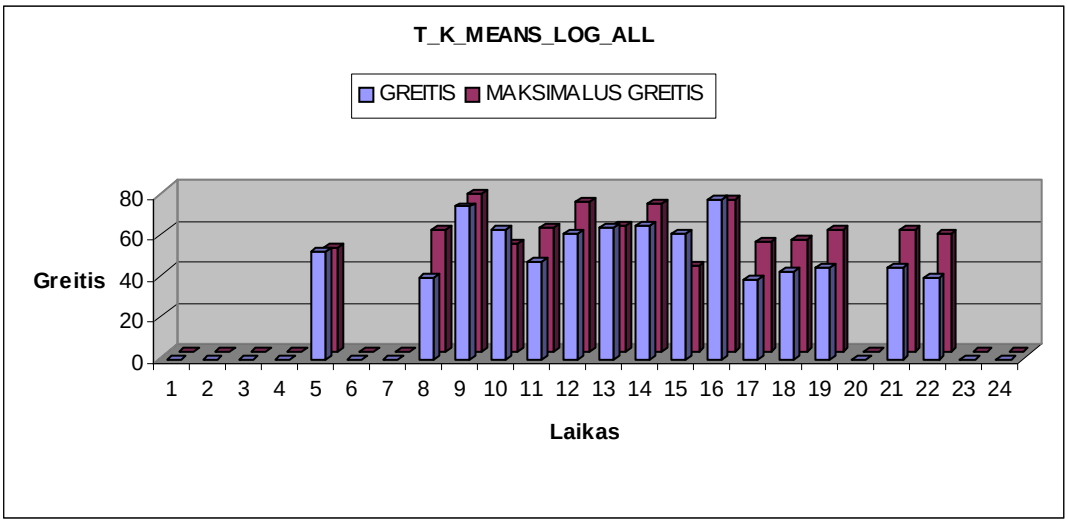
Iš 13 grafiko, kuriame pavaizduotas duomenų gautų darbo dienomis ir duomenų gautų savaitgaliais santykis, nesunku nustatyti, kad apkrovos didžiausios yra darbo dienomis.

Daugiau duomenų, lentelių ir grafikų galima rasti pridedamame CD diske.

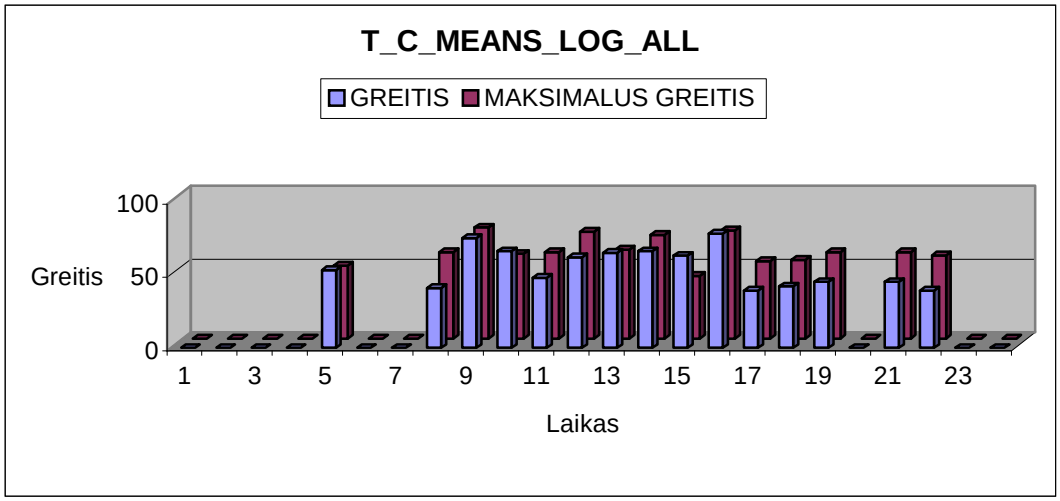
4.6. Vienmatis duomenų grupavimas

Kaip jau buvo minėta anksčiau, vienmačių duomenų grupavimo atveju duomenis skirstysime į grupes pagal laiko komponentę – tai leis apskaičiuoti kelių apkrovas tam tikrais

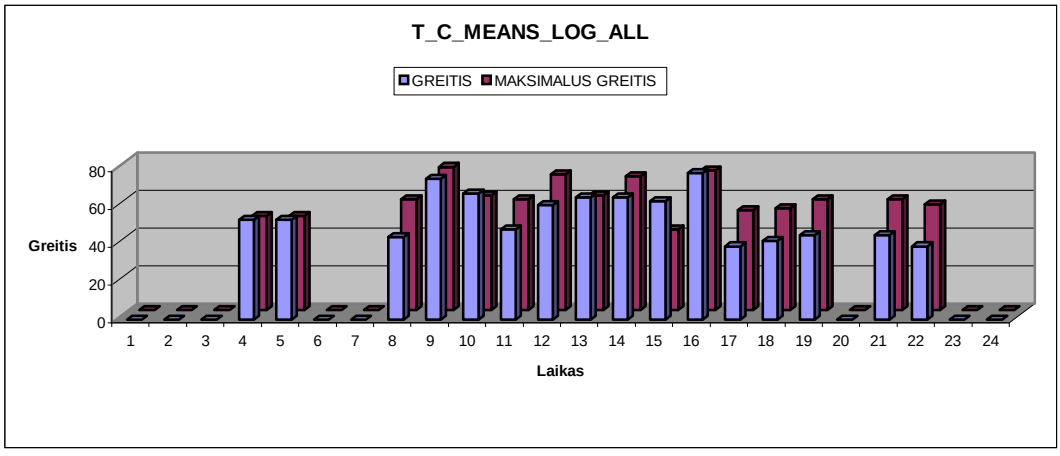
paros laiko tarpais bei nustatyti transporto spūstis (kelių eismo piko valandas). Toliau pabandysime palyginti K-means ir Fuzzy C-means algoritmų grupavimo rezultatus vienmačiams duomenims.



Pav. 14 K-means algoritmu sugrupuoti duomenys (24 grupės)



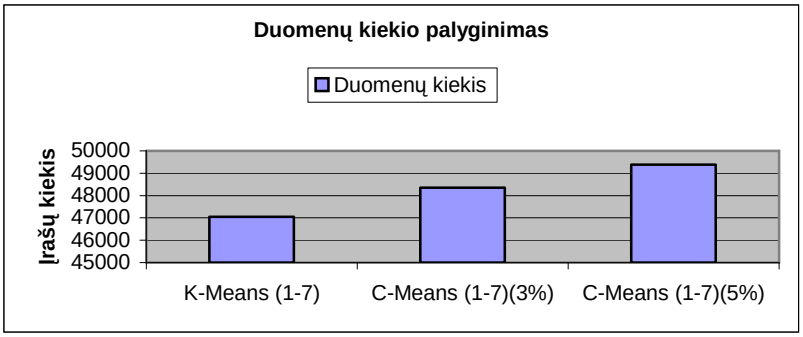
Pav. 15 Fuzzy C-means algoritmu sugrupuoti duomenys, kai narystės koef. = 3% (24 grupės)



Pav. 16 Fuzzy C-means algoritmu sugrupuoti duomenys, kai narystės koef. = 5% (24 grupės)

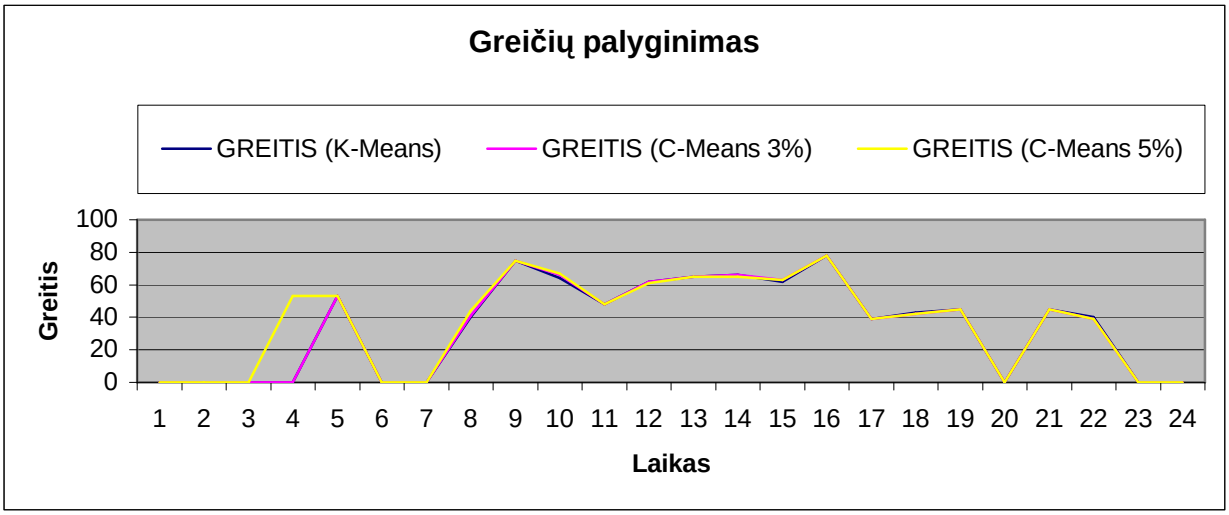
Kaip buvo minėta 2.1. skirsnyje praktikoje pasitaiko atvejų, kai po grupavimo kuri nors grupė lieka tuščia, todėl realizuojant vienmačius grupavimo algoritmus specialiai tuščios grupės buvo paliktos. Jos leis identifikuoti laiko tarpus, kada duomenų apie transporto srautus nėra gauta. Pavyzdžiui 14 ir 15 grafikuose matome, kad nuo 0 valandų iki 4 valandos ryte duomenų apie transporto srautus nėra gauta. Kai kur tokios informacijos spragos atsirado dėl to, kad techniniai resursai neleido išnagrinėti daugiau nei iš vieno vairuotojo gautus duomenis, o kai kur dėl privačios informacijos apie vairuotojus dalyvavusius projekte slėpimo.

Kaip galima matyti iš 15 ir 16 grafikų didinant Fuzzy C-means narystės koeficiento reikšmę duomenimis užsipildė daugiau grupių. Taip įvyko dėl grupių persidengimo. 5% narystės koeficientas duoda 10% procentų grupių persidengimą. 17 grafike pavaizduotos algoritmų operacijų duomenų aibės. Galime matyti, kad 1% narystės koeficiento padidinimas atitinkamai pridėjo beveik 1% naujų (pasikartojančių) duomenų aibės elementų.



Pav. 17 Operacijų aibių palyginimas

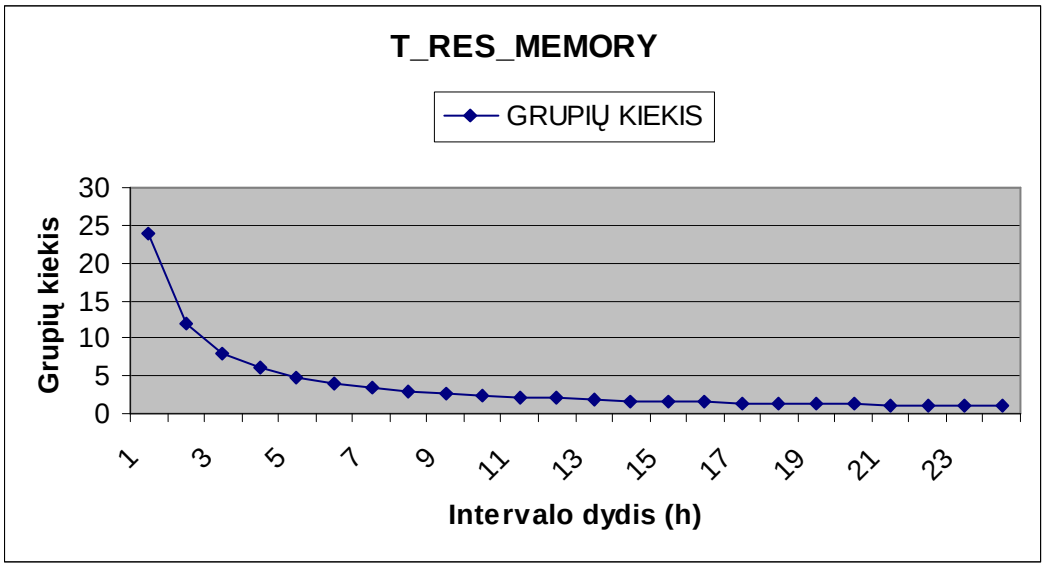
18 paveikslėlyje pavaizduotame grafike vaizduojami atskirų grupavimo algoritmų rezultatai grupuojant vienmačius duomenis.



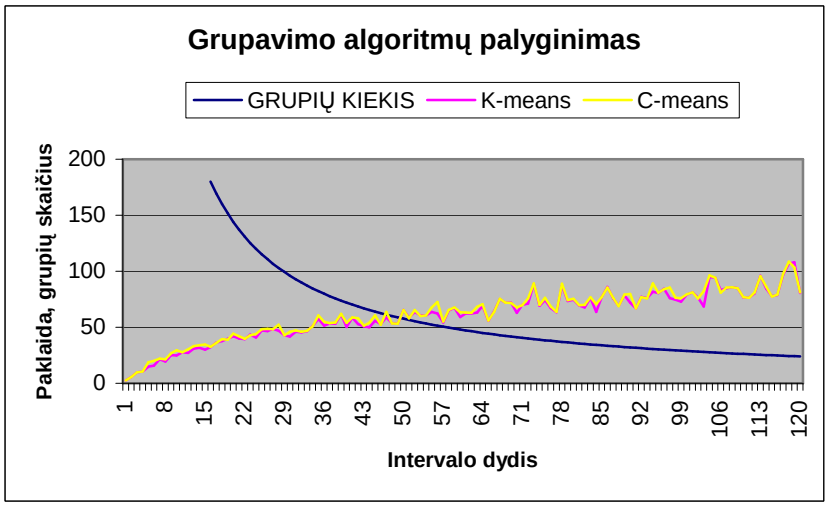
Pav. 18 Grupavimo algoritmų palyginimas

Kaip matome iš 18 grafiko abu grupavimo algoritmai parodė beveik vienodus rezultatus. Tik Fuzzy C-means algoritmas didinant narystės koeficientą atlieka atitinkamai daugiau skaičiavimų ir gali pateikti iškreiptus rezultatus (gali sugrupuoti duomenis į daugiau grupių nei iš tikro reikia), todėl galime teigti, kad K-means algoritmas vienmačiams grupavimams yra ir paprastesnis ir greitesnis ir tikslesnis.

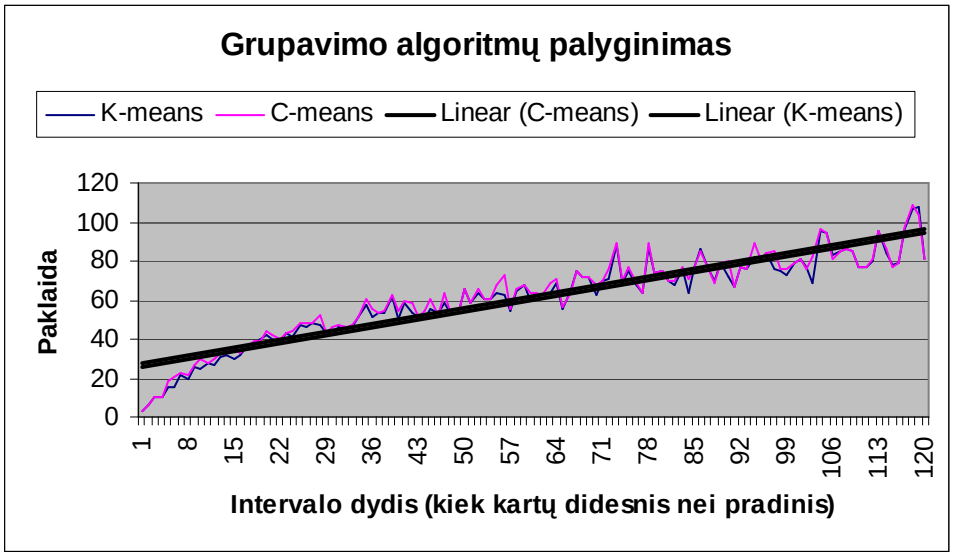
Pabandykime nustatyti priklausomybę tarp duomenų kiekio ir paklaidos tiek K-means, tiek Fuzzy C-means algoritmams. 19 grafike pavaizduota kreivė parodo grupių kiekio ir jų dydžio (pločio) tarpusavio priklausomybę. Šiame grafike paprastumo dėlei visi skaičiavimai buvo atlikti laikant, kad minimalus grupės dydis 1 valanda, bet analogiški rezultatai gaunami ir sumažinus grupės dydį.



Pav. 19 Grupių skaičiaus ir jų pločio (dydžio) santykis



Pav. 20 Priklausomybę tarp duomenų kiekio ir paklaidos



Pav. 21Grupavimo algoritmų įvertinimas

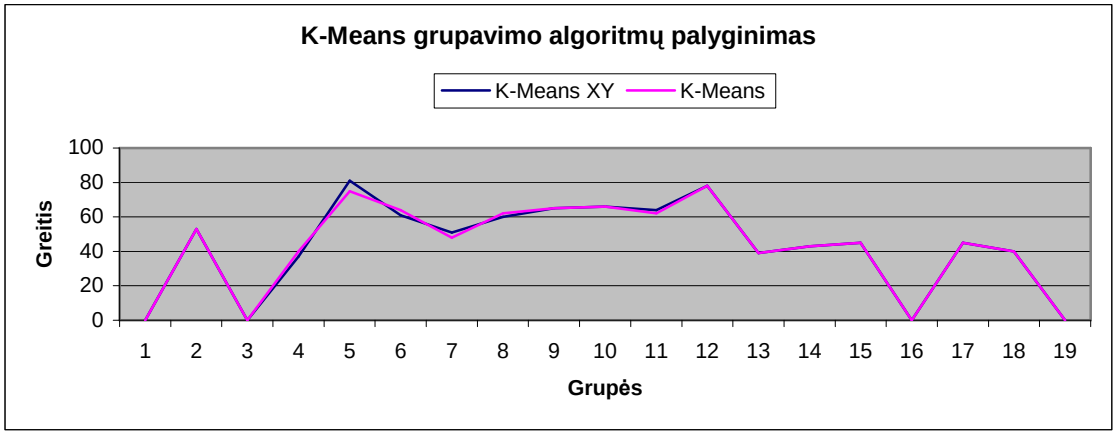
Kaip matyti iš 20 ir 21 grafikų galima laikyti, kad grupavimo rezultatų paklaida ir intervalų kiekis tarpusavyje susiję tiesiškai, nes mažinant grupių skaičių ir didinant jų dydį atitinkamai didėja ir paklaida.

Jei pvz. mums reikia pusės valandos tikslumu nustatyti piko valandas gatvėse, paėmę pradinę duomenų aibę iš kelių, keliasdešimt ar daugiau tūkstančių įrašų, ją galime sugrupuoti į 48 grupes, po vieną reikšmę. Nubrėžus tokį grafiką jo minimumai ir maksimumai parodys laiko tarpus, kada kelias daugiausiai ir mažiausiai apkrautas. Taip mūsų duomenų aibė gali sumažėti šimtus ar tūkstančius kartų, kas atitinkamai pagreitins rezultatų skaičiavimo laiką, duomenų apsikeitimą laiką, sumažins reikiamų resursų kiekį. Nes mes galime kiekvieną dieną, n metų saugoti po 1 mln. įrašų ir turėti n * 365 * 1 mln. įrašų arba saugoti po 48 įrašus ir turėti n * 365 * 48 įrašų.

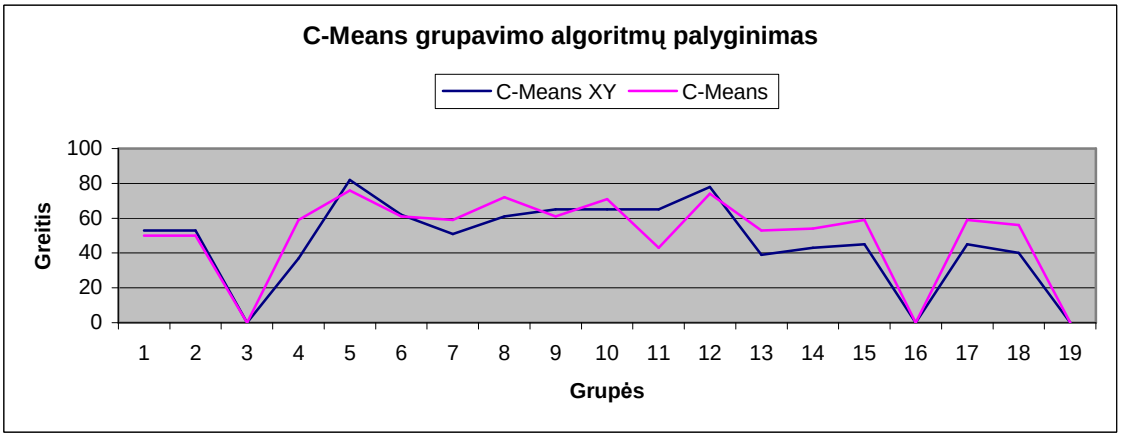
4.7. Dvimatis duomenų grupavimas

Dvimačių duomenų grupavimo atveju duomenis skirstysime į grupes pagal 2 komponentes: laiko ir greičio. Toks duomenų grupavimas leis mums konkrečiais paros laiko tarpais tiksliai žinoti realų vidutinį greitį tam tikroje kelio atkarpoje ir bus galima nesunkiai nustatyti patį greičiausią ir optimaliausią kelią nuo taško A iki taško B. Taip pat pabandysime palyginti duomenis gautus vienmačio ir dvimačio grupavimo atvejais.

Kaip matome iš 22 ir 23 grafikų K-means algoritmas tiek vienmačiu tiek dvimačiu atveju davė beveik identiškus rezultatus, o tuo tarpu Fuzzy C-means algoritmo rezultatai truputį išsiskyrė.



Pav. 22 Vienmačio ir dvimačio grupavimo K-means algoritmu palyginimas

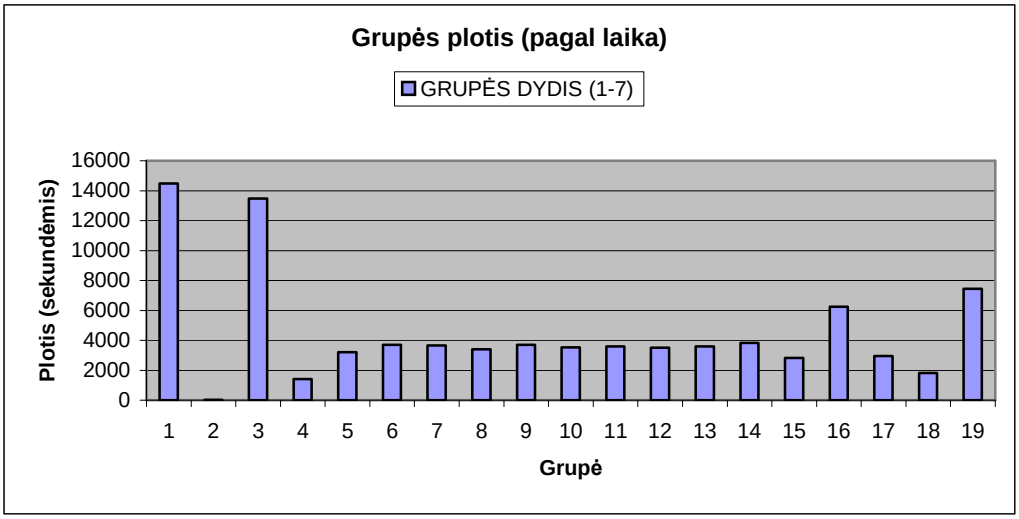


Pav. 23 Vinmačio ir dvimačio grupavimo Fuzzy C-means algoritmu palyginimas

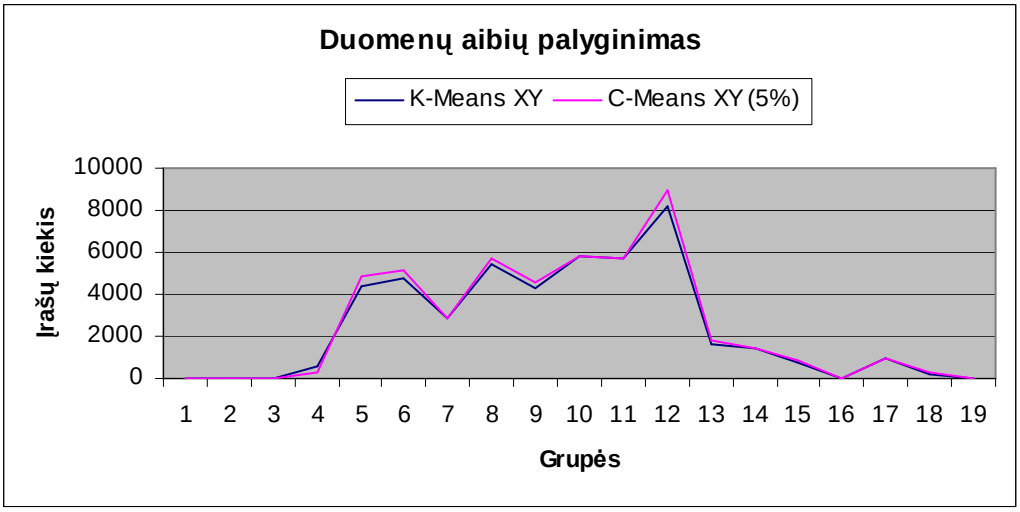
Iš 24 grafiko galime matyti kaip pasiskirstė grupių dydžiai (palyginti laiko intervalo dydžiais) sugrupavus duomenis K-means dvimačio grupavimo algoritmu. Iš grafiko matyti, kad naktį (para prasideda 0 val. 00 min. 00 s.) kai eismas nėra intensyvus grupės yra labai didelės, o tuo tarpu dieną darbo metu ir vakare eismas keliuose yra gerokai intensyvesnis, todėl ir sugrupavus duomenis, grupių dydžiai išlieka panašūs. Siekiant geresnių grupavimo K-means algoritmu rezultatų yra patariama grupių centrus išdėstyti kuo didesniais atstumais vienas nuo kito. Tokiu atveju tikimasi tikslesnio grupavimo. Kaip matome iš 24 grafiko transporto sistemoms tai nėra labai tinkanti rekomendacija. Grupavimas taptų efektyvesnis ir tikslesnis, jei grupių centrus pirminiame grupavime išdėstytume kuo arčiau centro, o dar geriau jei kuo daugiau grupių centrų atsirastų iš anksto planuojamuose piko taškuose (juos apytiksliai nustatyti yra gana nesunku, tai galima padaryti iš 9, 10, 11 grafikų). Toks pradinis centrų parinkimas būtų tikslingas siekiant gauti kuo tikslesnę informaciją apie greičius tam tikrose kelio atkarpose pačiu intensyviausiu paros metu, kai transporto srautai yra didžiausi (gaunama daugiausiai duomenų), labiausiai aktualūs ir yra sukaupiama daugiausiai informacijos. Iš tokio grafiko būtų galima

identifikuoti daugiau lokalių minimumų ir maksimumų, kurie tuo pačiu metu padėtų stebėti ir kelio apkrovas.

Iš 22 ir 23 grafikų taip pat matyti, kad intensyviausias eismas yra dienos viduryje ir norėtųsi toje laiko atkarpoje turėti kuo tikslesnę ir detalesnę informaciją.



Pav. 24 Dvimačio K-means algoritmo grupių dydžių pasiskirstymas



Pav. 25 Dvimačių grupavimo algoritmų duomenų aibių palyginimas

Kaip matyti iš 25 grafiko, C-means dvimačio grupavimo algoritmui reikia truputį didesnės duomenų aibės, kaip ir vienmačiu atveju, tačiau grupavimo rezultatai dėl to tikslesni nepasidaro.

Kaip ir vienmačiu atveju, dvimačiu atveju tikslingiau būtų naudoti K-means algoritmą, nes jis paprastesnis, greitesnis, reikalauja mažiau resursų, o galutinis grupavimo rezultatas tarp šio ir Fuzzy C-means algoritmų yra faktiškai vienodas (neturintis didelės įtakos prognozei).

5. Praktinis duomenų grupavimo algoritmų realizavimas

Kaip jau buvo minėta anksčiau šiam darbui atlikti buvo sukurta ir duomenimis užpildyta Oracle 9i duomenų bazių valdymo sistemoje veikianti duomenų bazė. Buvo suprogramuoti funkcijų ir pricedūrų paketai realizuojantys aptartus algoritmus.

5.1. Duomenų bazė

Šio darbo praktiniam realizavimui pasirinkta Oracle 9i duomenų bazių valdymo sistema. Sukurta MIF duomenų bazė. Duomenų bazės ir visų jos sudėtinių dalių kūrimo skriptų išeities kodai pridedami CD diske.

5.1.1. Tablespace

Tablespace’ai:

- AUTO – skirtas duomenų bazei ir duomenims siunčiamiems automobiliams per GPS imtuvus saugoti.
- AUTO_IDX – skirtas visiems schemos AUTO indeksams saugoti (tame tarpe ir pirminiams lentelių (santykių) raktams).

5.1.2. Vartotojai

Vartotojai:

- AUTO – vartotojas darbui su transporto sistemos duomenimis. Šio vartotojo vardu bus sukurta schema transporto sistemos duomenims saugoti.

5.1.3. Schemos

Schemos:

- AUTO – schema skirta vartotojo AUTO sukurtai transporto sistemos informacijai saugoti.

5.1.4. Rolės

Rolės:

- R_AUTO_READ – rolė, kuri leidžia tik skaityti iš schemos AUTO.
- R_AUTO_MODIFY – rolė, kuri leidžia skaityti ir rašyti shemoje AUTO.

5.1.5. Lentelės

Lentelės:

- T_LOG – lentelė gaunamiems “žaliems” (*raw*) duomenims iš GPS imtuvų sumontuotų automobiliuose saugoti.
- T_LOG_DATE – lentelė skirta visoms datoms ir jas atitinkančioms savaitės dienoms naudojamoms transporto sistemos duomenų bazėje saugoti.
- T_LOG_ALL – lentelė skirta visų turimų duomenų greičių vidurkių saugojimui.
- T_LOG_WEEK – lentelė skirta visų turimų duomenų greičių vidurkių darbo dienomis saugojimui.
- T_LOG_WEEKEND – lentelė skirta visų turimų duomenų greičių vidurkių savaitgaliais saugojimui.
- T_K_MEANS_LOG_ALL – lentelė skirta visų turimų duomenų suklasifikuotų vienmačiu K-means algoritmu saugojimui.
- T_K_MEANS_LOG_WEEK – lentelė skirta visų turimų duomenų darbo dienomis suklasifikuotų vienmačiu K-means algoritmu saugojimui.
- T_K_MEANS_LOG_WEEKEND – lentelė skirta visų turimų duomenų savaitgaliais suklasifikuotų vienmačiu K-means algoritmu saugojimui.
- T_K_MEANS_LOG_ALL_XY – lentelė skirta visų turimų duomenų suklasifikuotų dvimačiu K-means algoritmu saugojimui.
- T_K_MEANS_LOG_WEEK_XY – lentelė skirta visų turimų duomenų darbo dienomis suklasifikuotų dvimačiu K-means algoritmu saugojimui.
- T_K_MEANS_LOG_WEEKEND_XY – lentelė skirta visų turimų duomenų savaitgaliais suklasifikuotų dvimačiu K-means algoritmu saugojimui.
- T_C_MEANS_LOG_ALL – lentelė skirta visų turimų duomenų suklasifikuotų vienmačiu Fuzzy C-means algoritmu saugojimui.
- T_C_MEANS_LOG_WEEK – lentelė skirta visų turimų duomenų darbo dienomis suklasifikuotų vienmačiu Fuzzy C-means algoritmu saugojimui.

- T_C_MEANS_LOG_WEEKEND – lentelė skirta visų turimų duomenų savaitgaliais suklasifikuotų vienmačiu Fuzzy C-means algoritmu saugojimui.
- T_C_MEANS_LOG_ALL_XY – lentelė skirta visų turimų duomenų suklasifikuotų dvimačiu Fuzzy C-means algoritmu saugojimui.
- T_C_MEANS_LOG_WEEK_XY – lentelė skirta visų turimų duomenų darbo dienomis suklasifikuotų dvimačiu Fuzzy C-means algoritmu saugojimui.
- T_C_MEANS_LOG_WEEKEND_XY – lentelė skirta visų turimų duomenų savaitgaliais suklasifikuotų dvimačiu Fuzzy C-means algoritmu saugojimui.
- T_RES_MEMORY – lentelė skirta saugoti duomenims apie intervalų skaičiaus ir jų ilgio santykį.
- T_RES_AREA – lentelė skirta klasifikavimo algoritmų grafikų plotų ir jų santykių su nustatytu tikslaus grafiko plotu procentinių santykių skaičiavimui.

5.1.6. Indeksai

Siekiant pagreitinti duomenų paiešką, visų lentelių, visi stulpeliai (atributai) yra indeksuoti. Indeksų ir pirminių lentelių raktų (*primary key*) pavadinimai sudaromi, pagal schemą:

I_ + ‘lentelės pavadinimas’ + ‘stulpelio pavadinimas’

Kaip jau buvo minėta anksčiau visi indeksai saugomi AUTO_IDX tablespace’e.

5.2. Procedūrų ir funkcijų paketai duomenų grupavimui

5.2.1. Pkg_time_and_date

PKG_TIME_AND_DATE – paketas skirtas darbui su datomis ir laikais, jų vertimui iš NUMBER į DATE Oracle duomenų formatus.

- get_hours – funkcija, kuri iš NUMBER formato laiko išskiria valandas.
- get_minutes – funkcija, kuri iš NUMBER formato laiko išskiria minutes.
- get_seconds – funkcija, kuri iš NUMBER formato laiko išskiria sekundes.
- get_time_in_seconds – funkcija, kuri NUMBER formato laiką paverčia į sekundes.
- get_date – funkcija, kuri NUMBER formato datą paverčia į DATE formato datą.
- fill_t_log_date – procedūra, kuri užpildo lentelę skirtą visoms duomenų bazėje esančioms datoms ir jas atitinkančioms savaitės dienoms naudojamoms duomenų bazėje saugoti.

- clear_t_log_date – procedūra, kuri išvalo T_LOG_DATE lentelę ir atlaisvina visą jos užimtą atmintį.

5.2.2. Pkg_auto

PKG_AUTO – paketas skirtas lentelės T_LOG duomenų apdorojimui.

- fill_t_log_all – procedūra, kuri užpildo T_LOG_ALL lentelę.
- fill_t_log_week – procedūra, kuri užpildo T_LOG_WEEK lentelę.
- fill_t_log_weekend – procedūra, kuri užpildo T_LOG_WEEKEND lentelę.
- clear_table – procedūra, kuri ištrina duomenis iš nurodytos lentelės ir atlaisvina visą jos anksčiau užimtą atmintį.

5.2.3. Pkg_clust

PKG_CLUST – paketas skirtas vienmačiam duomenų grupavimui K-means ir Fuzzy C-means algoritmais.

- fill_t_k_means_log_all – procedūra, užpildanti T_K_MEANS_LOG_ALL lentelę duomenimis sugrupuotais vienmačiu K-means algoritmu.
- fill_t_k_means_log_week – procedūra, užpildanti T_K_MEANS_LOG_WEEK lentelę duomenimis sugrupuotais vienmačiu K-means algoritmu.
- fill_t_k_means_log_weekend – procedūra, užpildanti T_K_MEANS_LOG_WEEKEND lentelę duomenimis sugrupuotais vienmačiu K-means algoritmu.
- fill_t_c_means_log_all – procedūra, užpildanti T_C_MEANS_LOG_ALL lentelę duomenimis sugrupuotais vienmačiu Fuzzy C-means algoritmu.
- fill_t_c_means_log_week – procedūra, užpildanti T_C_MEANS_LOG_WEEK lentelę duomenimis sugrupuotais vienmačiu Fuzzy C-means algoritmu.
- fill_t_c_means_log_weekend – procedūra, užpildanti T_C_MEANS_LOG_WEEK lentelę duomenimis sugrupuotais vienmačiu Fuzzy C-means algoritmu.

5.2.4. Pkg_clust_xy

PKG_CLUST_XY – paketas skirtas dvimačiam duomenų grupavimui K-means ir Fuzzy C-means algoritmais.

- minimum_value – funkcija randanti mažiausią iš trijų skaičių.

- find_distance – funkcija skaičiuojanti euklido atstumą dvimatėje erdvėje tarp taško ir grupės centro.
- fill_t_k_means_log_all_xy – procedūra, užpildanti T_K_MEANS_LOG_ALL lentelę duomenimis sugrupuotais dvimačiu K-means algoritmu.
- fill_t_k_means_log_week_xy – procedūra, užpildanti T_K_MEANS_LOG_WEEK lentelę duomenimis sugrupuotais dvimačiu K-means algoritmu.
- fill_t_k_means_log_weekend_xy – procedūra, užpildanti T_K_MEANS_LOG_WEEKEND lentelę duomenimis sugrupuotais dvimačiu K-means algoritmu.
- fill_t_c_means_log_all_xy – procedūra, užpildanti T_C_MEANS_LOG_ALL lentelę duomenimis sugrupuotais dvimačiu Fuzzy C-means algoritmu.
- fill_t_c_means_log_week_xy – procedūra, užpildanti T_C_MEANS_LOG_WEEK lentelę duomenimis sugrupuotais dvimačiu Fuzzy C-means algoritmu.
- fill_t_c_means_log_weekend_xy – procedūra, užpildanti T_C_MEANS_LOG_WEEKEND lentelę duomenimis sugrupuotais dvimačiu Fuzzy C-means algoritmu.

5.2.5. Pkg_results

PKG_RESULTS – paketas skirtas apdoroti

- fill_t_res_memory – procedūra, kuri užpildo T_RES_MEMORY lentelę.
- fill_t_res_memory_part – procedūra, kuri dalinai (nurodytą žingsnių skaičių) užpildo T_RES_MEMORY lentelę
- calc_log_all_area – funkcija, kuri suskaičiuoja iš lentelės T_LOG_ALL duomenų nubrėžto grafiko apribotą plotą.
- calc_log_week_area – funkcija, kuri suskaičiuoja iš lentelės T_LOG_WEEK duomenų nubrėžto grafiko apribotą plotą.
- calc_log_weekend_area – funkcija, kuri suskaičiuoja iš lentelės T_LOG_WEEKEND duomenų nubrėžto grafiko apribotą plotą.
- calc_k_means_all_area – funkcija, kuri suskaičiuoja iš lentelės T_K_MEANS_LOG_ALL duomenų nubrėžto grafiko apribotą plotą.
- calc_k_means_week_area – funkcija, kuri suskaičiuoja iš lentelės T_K_MEANS_LOG_WEEK duomenų nubrėžto grafiko apribotą plotą.

- calc_k_means_weekend_area – funkcija, kuri suskaičiuoja iš lentelės T_K_MEANS_LOG_WEEKEND duomenų nubrėžto grafiko apribotą plotą.
- calc_c_means_all_area – funkcija, kuri suskaičiuoja iš lentelės T_C_MEANS_LOG_ALL duomenų nubrėžto grafiko apribotą plotą.
- calc_c_means_week_area – funkcija, kuri suskaičiuoja iš lentelės T_C_MEANS_LOG_WEEK duomenų nubrėžto grafiko apribotą plotą.
- calc_c_means_weekend_area – funkcija, kuri suskaičiuoja iš lentelės T_C_MEANS_LOG_WEEKEND duomenų nubrėžto grafiko apribotą plotą.
- calc_k_means_all_xy_area – funkcija, kuri suskaičiuoja iš lentelės T_K_MEANS_LOG_ALL_XY duomenų nubrėžto grafiko apribotą plotą.
- calc_k_means_week_xy_area – funkcija, kuri suskaičiuoja iš lentelės T_K_MEANS_LOG_WEEK_XY duomenų nubrėžto grafiko apribotą plotą.
- calc_k_means_weekend_xy_area – funkcija, kuri suskaičiuoja iš lentelės T_K_MEANS_LOG_WEEKEND_XY duomenų nubrėžto grafiko apribotą plotą.
- calc_c_means_all_xy_area – funkcija, kuri suskaičiuoja iš lentelės T_C_MEANS_LOG_ALL_XY duomenų nubrėžto grafiko apribotą plotą.
- calc_c_means_week_xy_area – funkcija, kuri suskaičiuoja iš lentelės T_C_MEANS_LOG_WEEK_XY duomenų nubrėžto grafiko apribotą plotą.
- calc_c_means_weekend_xy_area – funkcija, kuri suskaičiuoja iš lentelės T_C_MEANS_LOG_WEEKEND_XY duomenų nubrėžto grafiko apribotą plotą.
- fill_t_res_area_k_means_all – procedūra skirta T_RES_AREA lentelei užpildyti ir vienmačio K-means algoritmo apskaičiuojamų plotų santykių su T_LOG_ALL lentelėje esančiais tiksliais duomenimis palyginimui
- fill_t_res_area_c_means_all – procedūra skirta T_RES_AREA lentelei užpildyti ir vienmačio C-means algoritmo apskaičiuojamų plotų santykių su T_LOG_ALL lentelėje esančiais tiksliais duomenimis palyginimui
- clear_table – procedūra, kuri ištrina duomenis iš nurodytos lentelės ir atlaisvina visą jos anksčiau užimtą atmintį.

Rezultatų aptarimas

Šio darbo rezultatai galėtų būti panaudoti transporto srautų analizės informacinėse sistemose. Pavyzdžiui, dabar daugelyje automobilių montuojamos palydovinės navigacijos sistemos, internetiniuose portaluose, galima pasižiūrėti skaitmeninių žemėlapių, susiplanuoti kelionių maršrutus, apskaičiuoti tų kelionių trukmę ir kelionės ilgį. Tačiau šitie skaičiavimai nėra visiškai tikslūs ir dažnu atveju nesutampa su realybe. Dėl to, kad tokios sistemos neanalizuoja realaus laiko duomenų, o pateikia tik teorinius skaičiavimus. Paprastai tokios sistemos atlieka skaičiavimus naudodamos faktiškai kelio atkarpose leistinus greičius (nustatytus kelio ženklais), tačiau kaip jau matėme iš šio darbo realybė dažnai būna kitokia ir reikia įvertinti kad faktiškai leidžiamas greitis ne visada sutampa su realiu greičiu. Realūs skaičiavimai atima daug laiko, reikalauja daug resursų, labai sunku nepaklysti duomenų “jūroje”. Tačiau naudojant duomenų grupavimą galima smarkiai sumažinti duomenų kiekį ir vistiek atlikti tikslią rezultatų prognozę. Iš centrinio informacijos centro per palydovinę įrangą automobilių navigacinėms sistemoms galima greitai nusiųsti atnaujintą informaciją, kuri galėtų būti apdorota, palyginta su praeities duomenimis ir vairuotojui galėtų būti pateikta tiksliausia įmanoma prognozė apie padėtį keliuose.

Išvados

Atlikus darbą, galima daryti tokias išvadas:

- Grupavimo algoritmų galimybių panaudojimas įvairiose transporto sistemose leistų kelis šimtus, tūkstančius ar daugiau kartų sumažinti galutiniams transporto srautų analizės rezultatams skaičiuoti reikalingų duomenų kiekį.
- Atliekant transporto informacinių sistemų duomenų grupavimą reikia iš anksto žinoti kokių rezultatų ir kokio duomenų tikslumo yra siekiama, tai leidžia duomenų kiekį sumažinti iki optimalios ribos.
- Vienmatis duomenų grupavimas nėra informatyvus. Analogiškų tikslų su papildoma informacija galima pasiekti naudojant keliamatį grupavimą ir tai nekainuoja daug papildomų resursų ir laiko.
- Ne visos grupavimo algoritmų šeimos vienodai gerai tinka informacinėms transporto sistemoms ir jų duomenų grupavimui.
- Realizuoti grupavimo algoritmai leidžia teigti, kad transporto sistemoms labiausiai tinka išskirtinių grupavimo algoritmų šeimos algoritmai.
- Transporto sistemose, norint pasiekti tikslesnių analizės rezultatų, parenkant pirminius grupių centrus reikia stengtis juos parinkti nutolusius ne didžiausiu atstumu vienas nuo kito, o tose vietose, kurios reikalingos tyrimui atlikti.
- Per didelė Fuzzy C-means algoritmo narystės koeficiento reikšmė iškreipia grupavimo rezultatus. Grupavimui naudojant sutampančius algoritmus, šio koeficiento reikšmė neturėtų būti daugiau nei keli procentai.
- Faktiškai galima laikyti, kad grupavimo rezultatų paklaida ir grupių kiekis tarpusavyje yra tiesiškai susiję. Ši priklausomybė yra atvirkštinė - kiek kartų sumažinsime grupių kiekį, tiek procentų padidinsime paklaidą.

Literatūros sąrašas

[1] **J. B. MacQueen**: Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297. 1967

[2] **Brian T. Luke**. K-Means Clustering. *http://fconyx.ncifcrf.gov/%7Elukeb/kmeans.html*

[3] **Osmar Zaiane**. Principles of knowledge discovery in databases. 1999. *http://www.cs.ualberta.ca/%7Ezaiane/courses/cmput690/slides/Chapter8/index.html*

[4] **Andrew Moore**. K-means and Hierarchical Clustering. 2001. *http://www.autonlab.org/tutorials/kmeans11.pdf*

[5] **Hans-Joachim Mucha, Hizir Sofyan**. Cluster Analysis. 2003. *http://www.quantlet.com/mdstat/scripts/xag/html/xaghtmlframe142.html*

[6] **Stephen P. Borgatti**. How to Explain Hierarchical Clustering. 1994. *http://www.analytictech.com/networks/hiclus.htm*

[7] **Miranda Maria Irene**. Clustering methods and algorithms. 1999. *http://www.cse.iitb.ac.in/dbms/Data/Courses/CS632/1999/clustering/dbms.html*

[8] A Tutorial on Clustering Algorithms *http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/index.html*

[9] **C. S. Jensen, H. Lahrmann, S. Pakalnis, and J. Runge**. The INFATI data. 2004. *http://www.cs.aau.dk/research/DP/tdb/TimeCenter/TimeCenterPublications/TR-79.pdf*

[10] **P. Heide**. INFATI Hardware og Software. Technical Report. 2004. *http://www.trg.dk/projekter/infati/notat2.pdf*

[11] **J. Juhl**. INFATI Mapmatching. Technical Report. 2004. *http://www.trg.dk/projekter/infati/notat3.pdf*.

Priedas Nr. 1

CD diskas su šiame darbe panaudotais duomenimis, duomenų importo skriptų kodais, sukurtos programinės įrangos išvesties kodais, vykdyto tyrimo rezultatais bei šio darbo elektroniniu variantu.