UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA INFORMATIKY

KOMBINÁCIA AUTOMATICKÝCH A VIZUÁLNYCH METÓD DOLOVANIA DÁT

DIPLOMOVÁ PRÁCA

TOMÁŠ POLÁČEK

2008

Kombinácia automatických a vizuálnych metód dolovania dát.

DIPLOMOVÁ PRÁCA

Tomáš Poláček

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA INFORMATIKY

Vedúci diplomovej práce
Novotný Matej, Mgr.

BRATISLAVA 2008

Zadanie diplomovej práce

Kombinácia automatických a vizuálnych metód pre dolovanie dát. Opísať klady a zápory oboch prístupov, navrhnúť spôsob ako prístupy skombinovať čo najlepšie a ilustrovať takúto kombináciu na zvolenej implementácii.

Abstrakt

Automatické a vizuálne metódy dolovania dát sú pomerne preskúmané a často používané. Prirodzene vznikla požiadavka oba prístupy skombinovať. Vychádzali sme z predpokladu, že táto kombinácia by mala priniesť pozitívne výsledky vďaka využitiu dobrých vlastností oboch metód. V práci sme sa zamerali na segmentáciu dát pomocou K-Means algoritmu. Rozšírili sme ho pomocou vizuálneho zadávania stredov zhlukov a vizuálného definovania oddeľujúcich nadrovín. Podarilo sa nám zachytiť a vizualizovať proces segmentácie dát a umožniť jeho opätovné použitie na iné dáta. Všetky prezentované postupy sme implementovali v aplikácii, ktorá je súčasťou práce. Ukázalo sa, že zvolený spôsob rozšírenia K-Means algoritmu priniesol presnejšie vypočítanie zhlukov v dátach. Vďaka vizuálnej prezentácii analytického procesu sú vypočítané výsledky jasnejšie a dôveryhodnejšie. Aplikácia vďaka vizualizácii a interakcii využíva vo väčšej miere skúsenosti používateľa ako bežné automatické metódy.

Kľúčové slová: vizualizácia, dolovanie dát, K-Means, segmentácia dát, analýza zhlukov, scatterplot

Poďakovanie

Poďakovanie patrí:

- mojej rodine za trpezlivosť a vytvorenie podmienok na písanie práce
- slečne Lucii Sodomovej za podporu a inšpiráciu
- pánovi Petrovi Tomášovi za technickú pomoc
- pánovi Mgr. Matejovi Novotnému za povzbudivé slová a dobré rady

Obsah

1 Úvod

V dnešnej dobe je svet zaplavený miliónmi terabajtov dát. Väčšina firiem a spoločností zhromažďuje dáta o zákazníkoch, o svojich obchodných činnostiach a uskladňuje ich v dátových úložiskách. Veľké množstvo dát vzniká pri rôznych výskumoch, meraniach a vedeckých činnostiach. Často je množstvo nazbieraných dát také obrovské, že ich nie je možné skúmať obyčajným prezeraním nameraných hodnôt. Prirodzene tak vznikla požiadavka proces skúmania a získavania informácií z veľkého množstva dát zjednodušiť a v čo najväčšej miere automatizovať. V súčastnosti nám to umožňujú rýchle paralelné počítače s obrovským výkonom a pamäťovými kapacitami. **Dolovanie dát** sa dá definovať ako netriviálne získavanie implicitných, pred tým neznámych, a potencionálne užitočných informácií [8].

Dolovanie dát zahŕňa veľké množstvo techník z viacerých oblastí ako napríklad matematika, štatistika, umelá inteligencia alebo počítačová grafika. Dolovanie dát nám napomáha hľadať tie informácie, ktoré sú pre nás zaujímavé a niečím význačné.

Využite Dolovania dát môžeme nájsť v rôznych oblastiach života. Manažéri firiem ho môžu využiť ako pomôcku pri rozhodovaní o marketingových stratégiách spoločnosti a overiť si tak rôzne hypotézy a predikcie. V zdravotníctve ho môžme využiť na vyhľadávanie skupín príznakov a liekov. V bankovníctve má využitie napríklad pri hľadaní bankových podvodov. Veľké využitie vidíme pri predikcii správania sa zákazníka, odhadu predaja produktov alebo obchodných procesov [10].

Dolovanie dát nemôže úplne nahradiť skúseného analytika. Využíva sa skôr ako pomocný nástroj pri analýze, ktorej výsledky treba konfrontovať so skúsenosťami v oblasti jeho použitia. Z toho vyplýva, že znalosť aplikačnej domény je veľmi dôležitá.

2 Prehľad problematiky

Metódy dolovania dát môžeme rozdeliť do dvoch základných skupín. Prvú skupinu tvoria vizuálne metódy a druhú tvoria metódy automatické.

2.1 Vizuálne metódy dolovania dát

Dolovanie dát pomocou ich vizualizácie umožňuje hľadať skryté informácie prostredníctvom ľudského zraku. Vizualizácia sa snaží prezentovať dáta formou obrazu tak, aby boli ľahko čitateľné pre ľudské oko. Vizualizácia by mala potlačiť menej dôležité informácie v dátach a zvýrazniť tie, ktoré sú podstatné a zaujímavé. Vizualizovanie dát sa opiera o schopnosť ľudského oka vnímať veľké množstvo vizuálnych informácií. Ľudský zrak je dobre trénovaný a citlivý na vnímanie vizuálnej informácie. V spojení s ľudským mozgom a znalosťami v oblasti, kde sa dolovanie plánuje použiť, tvorí vynikajúci nástroj na odhaľovanie skrytých informácií.

Daniel A. Keim, odborník v oblasti vizualizácie a dolovania dát, rozdelil vo svojom článku [14] množinu vstupných dát pre vizualizáciu do niekoľko kategórií.

Jednodimenzionálne dáta: Tieto dáta majú len jeden. Typickým príkladom sú časovo závislé dáta. K jednému časovému bodu je priradená jedna hodnota nejakého atribútu. Príkladom môže byť postupnosť cien nejakej komodity na burze alebo postupnosť nameraných teplôt na nejakom mieste v rôznych časových okamihoch.

Dvojdimenzionálne dáta: Tieto dáta majú dva rôzne rozmery, môžu to byť napríklad geografické dáta, ktoré obsahujú geografickú výšku a šírku. Môže sa zdať, že vizualizovať dvojrozmerné dáta je jednoduché, ale treba byť opatrný. Ak je počet záznamov príliš veľký, môže sa stať, že výsledný obraz bude príliš chaotický a nebude z neho jasne vidieť štruktúru dát.

Multidimenzionálne dáta: Do tejto skupiny patria tie dáta, ktoré majú veľa záznamov a väčšina z nich pozostáva z viac ako troch rozmerov. Je zrejmé, že takéto dáta sa nemôžu vizualizovať pomocou klasických 2D alebo 3D grafov. Príkladom multidimenzionálnych dát sú záznamy z relačnej databázy, ktoré často pozostávajú z desiatok alebo stoviek stĺpcov. Kedže takéto dáta sa nedajú jednoduchým spôsobom zobraziť na obrazovku počítača, treba použiť oveľa sofistikovanejšie metódy. Jednou z nich sú napríklad paralelné

súradnice. V práci sa zameriame na tento typ dát.

Text a hypertext: Nie všetkým dátam sa dá určiť počet rozmerov. V dnešnej dobe internetu a technológie www je text veľmi dôležitý typ dát. Bohužiaľ, text sa dá veľmi ťažko popísať číslami, preto nemôžme na jeho vizualizáciu aplikovať väčšinu bežných vizualizačných techník. Text je nutný pred samotnou vizualizáciou najprv transformovať. Ako príklad môžeme zobrať počítanie slov alebo počítanie výskytu písmen.

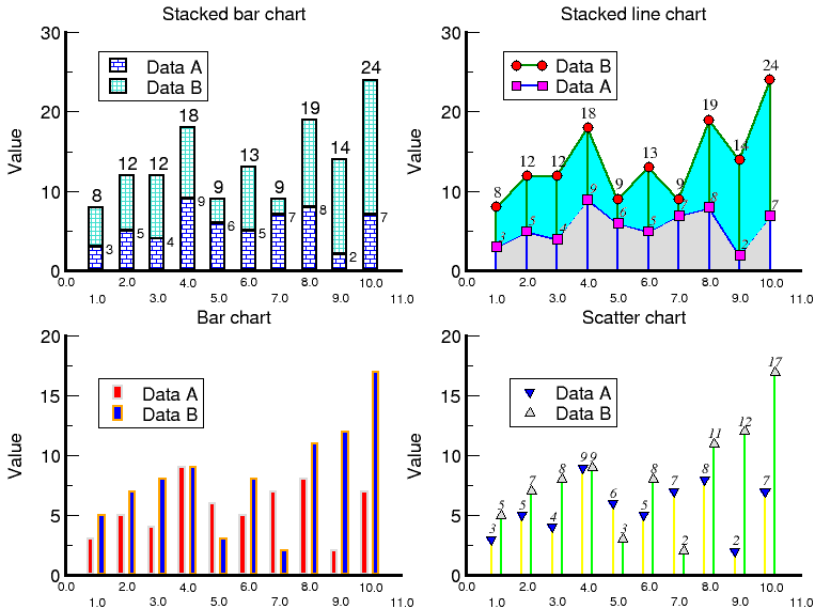
Hierarchie a grafy: Dátové záznamy v tejto skupine majú zvyčajne nejaký vzťah k dalším záznamom a objektom. Na vizualizáciu takýchto vzťahov sa zvyčajne používajú grafy. Graf pozostáva z vrcholov, ktoré sú pospájané hranami. Tieto hrany reprezentujú sémantické prepojenie medzi objektami, ktoré daná hrana spája. Príkladom môže byť napríklad štruktúra súborového systému alebo linky v sieti www. Podobnú vizualizáciu použijeme aj neskôr v našej práci na prezentáciu stromovej štruktúry analytického sedenia v časti 4.3.2.

Algoritmy a softvér: Tieto dáta tvoria špecifickú skupinu. Cieľom ich vizualizácie je podporiť vývoj softvéru pomocou lepšieho pochopenia kódu. Vďaka vizualizácii sa ľahšie pochopí štruktúra a tok informácií v softvérovom diele. Iný spôsob využitia je vizualizácia chýb, čím sa napomáha ľahšiemu a presnejšiemu ladeniu aplikácie.

Kaim vo svojom článku [14] rozdelil metódy vizualizácie na niekoľko nasledovných kategórií.

Štandardné 2D a 3D zobrazenia: Jedná sa o klasické 2D a 3D stĺpcové alebo bodové grafy [21]. Ukážka stĺpcových grafov je na obrázku 1. Podobné vizualizácie sú nepoužiteľné na dáta s vysokou dimenzionalitou, ale sú veľmi ľahko vnímateľné pre človeka a ľahko sa interpretujú. Človek sa s takýmito vizualizáciami stretáva denne a je na ne dobre trénovaný. Pre tieto nesporné výhody sme si tento typ vizualizácie zvolili aj v našej práci. Budeme intenzívne využívať bodové grafy.

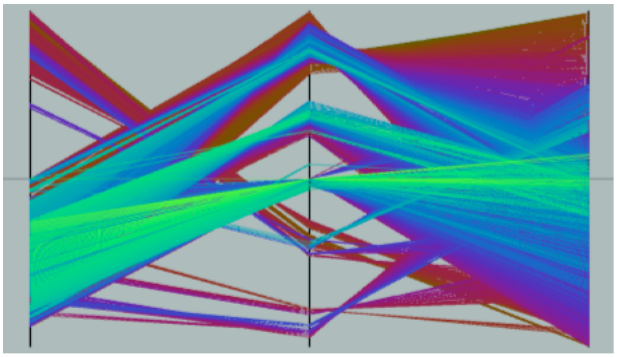
Geometricky transformované zobrazenia: Techniky sú založené na hľadaní transformácií viacrozmerných dát, ktoré nám uľahčia rozpoznať skryté informácie [18]. Patria sem napríklad paralelné súradnice (Obr. 2). Táto technika mapuje k rozmerné dáta na dva rozmery obrazovky pomocou k ekvidištantne rozmiestnených osí. Viac o tejto technike sa dá dočítať napríklad v diplomovej práci Mateja Novotného [19].



Obr. 1: Stĺpcové grafy

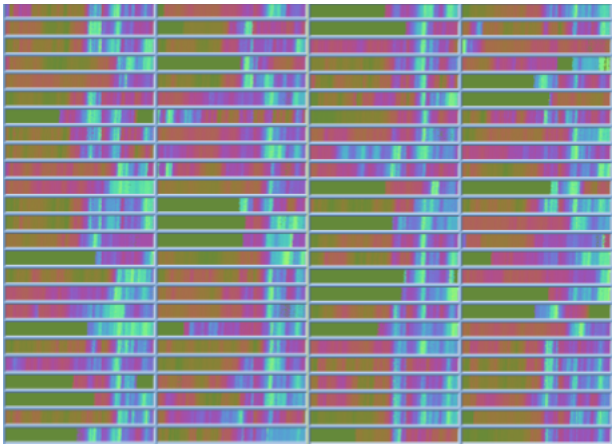
Ikonické zobrazenia: Myšlienka tejto techniky je mapovať atribúty multidimezionálnych dát na parametre ikonického zobrazenia [7]. Ikony môžu byť rôznych druhov, malé zjednodušené ľudské tváre, hviezdičky, alebo ikony v tvare ihly. V prípade ikon pripomínajúcich ľudské tváre sa zistilo, že ľudské oko je dobre trénované v ich rozpoznávaní a vnímaní ich výrazu. Pri tejto technike sa využíva mapovanie atribútov dátového záznamu do jednotlivých čŕt ľudskej tváre, napríklad do tvaru úst, nosa, obočia, hlavy atď. Výsledný graf je veľmi ľahko vnímateľný a čitateľný aj keď zobrazuje veľa rozmerov. Nevýhodou tohto zobrazenia je, že dokáže zobraziť len malé množstvo dát. Ikony väčšinou zaberajú veľa miesta a preto veľmi rýchlo preplnia obrazovku.

Sieť pixlov: Základný princíp týchto techník je namapovať každú hodnotu záznamu do farebného bodu na obrazovke a zoskupiť body patriace jednému rozmeru do oddelenej oblasti [13]. Kedže v základných zobrazeniach tohto typu sa mapuje jedna hodnota na jeden bod, takáto technika umožňuje zobraziť naraz veľké množstvo dát (Obr. 3). Rôzne techniky používajú rôzne usporiadanie bodov, aby lepšie zobrazili korelácie a vzťahy medzi dátami. Príkladom sú techniky rekurzívneho vzoru, alebo technika kruhových



Obr. 2: Paralelné súradnice

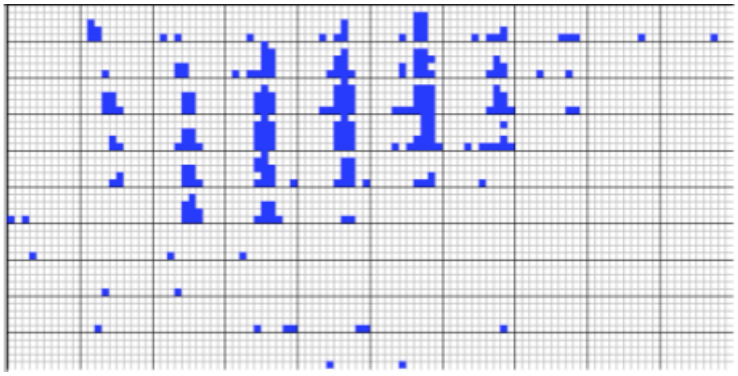
segmentov.



Obr. 3: Sieť pixlov

Zložené zobrazenia: Tieto techniky sú určené na zobrazenie dát, ktoré majú isté hierarchické usporiadanie. V prípade multidimezionálnych dát sa musia rozmery, ktoré sú určené na vybudovanie hierarchie, určovať veľmi pozorne. Príkladom môže byť technika skladania rozmerov, ktorá vkladá jeden súradnicový systém do druhého [16]. Ukážka takéhoto zobrazenia je na obrázku 4.

Veľmi dôležitou súčasťou vizualizácie je interakcia s používateľom. Používateľ oveľa ľahšie pochopí vizualizáciu, ak nie je statická môže s ňou interagovať. Základný prehľad techník uviedli vo svojom článku [15] Kosara,



Obr. 4: Zložené zobrazenie

Hauser a Gresh. V článku skúmajú vizualizáciu dát z hľadiska interakcie s používateľom. Daniel A. Keim vo svojom článku [14] rozdelil metódy vizualizácie na základe interakcie na nasledovné skupiny:

Dynamické projekcie: Metódy v tejto kategórii sú založené na dynamickej zmene projekcií viacrozmerných dát. Projekcie sa môžu meniť automaticky alebo pričinením používateľa.

Interaktívne filtrovanie: Do tejto skupiny patrí napríklad priama selekcia podmnožiny dát používateľom alebo určenie podmnožiny pomocou atribútov, ktoré má daná podmnožina spĺňať. Podobné techniky využijeme aj v našej práci. Umožníme používateľovi selektovať podmnožiny dát definovaním obdĺžnikových oblastí, ktoré budú dátové záznamy do selekcie pridávať alebo z nej odoberať.

Interaktívna zmena veľkosti: Pri veľkom množstve dát je zmena veľkosti veľmi dôležitá. Pri prvom pohľade musia byť dát zobrazené v kompaktnej forme a používateľ si následne sám určí tie oblasti, ktoré ho zaujímajú. Pri náhľade na dáta sa zámerne vynechávajú niektoré detaily, aby nebol obraz príliš zložitý. Pri postupnom zväčšovaní sa postupne vynárajú pred tým skryté detaily.

Interaktívne skreslenie a deformácia: Tieto techniky vykreslia niektorú časť dát s vyšším rozlíšením a dôrazom na detaily, zatiaľ čo zvyšok dát je vykreslený menej detailne s menším dôrazom. Výhoda je, že aj keď sme zameraný na určitú časť dát, nestrácame prehľad o zvyšných dátach, ktoré takto nezmiznú z obrazovky, ale sa iba vykreslia v menšom rozlíšení.

Interaktívne prepojenie a selekcia: Používateľ môže na jednu množinu dát aplikovať viacero techník vizualizácie. Aby sa v nich nestratil, je treba dodržať princíp Focus + Context. Používateľ musí mať možnosť zamerať sa detailnejšie na podmnožinu vstupných dát ale pritom nesmie stratiť celkový prehľad o dátach. Ak používateľ selektuje nejakú podmnožinu dát v jednej vizualizácii, mala by sa rovnaká podmnožina dát selektovať aj vo všetkých ostatných vizualizáciach. Túto metódu nazývame interaktívne prepojenie pohľadov. Využili sme ju aj v našej práci. Podarilo sa nám tak udržať konzistentný pohľad na zobrazované dáta.

2.2 Automatické metódy dolovania dát

Prehľadový článok [22] o automatických metódach dolovania dát zverejnil na svoje stránke Kurt Thearling. Metódy rozdelil do dvoch hlavných kategórií. Do prvej kategórie patria klasické techniky a do druhej techniky novej generácie.

Klasické techniky: Tieto techniky sú používané už dlhšiu dobu. Založené sú predovšetkým na matematike a štatistike.

- **Histogramy:** Pri výpočte histogramu sa zvolí jeden parameter (prediktor) a následne sa vypočíta výskyt jeho jednotlivých hodnôt. Táto technika nám podá vysoko úrovňový náhľad na skúmané dáta. Pohľadom na histogram získame intuitívnu predstavu o dátach a ich usporiadaní a o niektorých dôležitých štatistických vlastnostiach skúmaných dát.

- **Lineárna regresia:** Jedná sa o druh štatistickej predpovede. Najjednoduchšia forma regresie je lineárna regresia, ktorá obsahuje len jeden prediktor a predpoveď. Lineárna regresia sa dá triviálne rozšíriť do viacerých rozmerov pridávaním ďalších prediktorov. Iné, nelineárne regresie môžeme získať umocňovaním alebo násobením a delením prediktorov.

- **Najbližší sused a delenie do zhlukov:** Tieto techniky využívajú podobnosť záznamov v databáze vzhľadom na určené parametre. Metóda delenia do zhlukov zaradí jednotlivé záznamy do skupín, pričom záznamy v jednej skupine sú si v istom zmysle podobné. Princíp najbližšieho suseda spočíva v tom, že k danému neúplnému záznamu nájdeme záznam, ktorý je mu najviac podobný a na základe neho odhadneme

(predpovedáme) chýbajúce parametre. Delenie do zhlukov aj princíp najbližšieho suseda môžu slúžiť na predpovedanie procesov. Rozdelením záznamov do zhlukov môžeme veľmi ľahko odhaliť hrubú štruktúru a rozloženie dát v dátovom súbore. Viac informácií môžete nájsť v článku [12]. Z tejto skupiny automatických metód dolovania dát sme si v našej práci zvolili K-Means algoritmus [3] a [1]. Je to pomerne rozšírená a často používaná metóda na rozdelenie dát do zhlukov.

Techniky novej generácie: Techniky, ktoré boli vyvinuté v posledných dvadsiatich rokoch.

- **Rozhodovacie stromy:** Je to prediktívny model, ktorý predpovedá na základe niekoľkých rozhodnutí. Tieto rozhodnutia prebiehajú na stromovej štruktúre. V listoch stromu sú finálne kategórie. Uzly stromu reprezentujú nejaké rozhodovacie kritérium na základe ktorého sa pri kategorizáciu nejakého prvku pokračuje buď do ľavého alebo pravého podstromu. Táto technika sa využíva napríklad pri chemickej diagnostike chorôb.
- **Neuronové siete:** Sú siete na rozpoznávanie vzorov. Dokážu sa učiť a meniť svoju štruktúru. Neuronové siete napodobňujú ľudský mozog. Väčšinou sa učia postupne jeden záznam za druhým, ale niektoré algoritmy dokážu spracovať všetky trénovacie dáta naraz. Neuronové siete majú veľké využitie vo viacerých odboroch.

2.3 Kombinované techniky

Kombinácia automatických a vizuálnych metód je v súčastnosti intezívne skúmaná. Ukazuje sa, že skombinovaním oboch prístupov získame presnejšie dôveryhodnejšie výsledky. Porovnaním oboch metód a možnosťou ich vzájomnej integrácie sa zaoberal Thomas Rongitsch vo svojej diplomovej práci [20].

Naša práca svojim obsahom spadá tiež do tejto kategórie. Pokúsime sa nájsť kombinovanú techniku, ktorá bude poskytovať lepšie výsledky ako samostatne použité metódy, z ktorých sa skladá.

3 Kombinácia automatických a vizuálnych metód dolovania dát

Automatické a vizuálne metódy dolovania dát sú v dnešnej dobe pomerne dosť preskúmané a často využívané. Prirodzeným pokusom ako ich vylepšiť je ich vzájomné spojenie, využitie výhod oboch prístupov a potlačenie ich nedostatkov.

Častým problémom automatických metód je ich konfigurácia a prezentácia výsledkov. Príkladom môžu byť algoritmy pre segmentáciu dát. Ich výstupom sú množiny záznamov pre jednotlivé vypočítané zhluky. Z takéhoto výstupu je však veľmi obtiažne odhadnúť napríklad tvar vypočítaného zhluku.

V práci sa pokúsime sa nájsť spôsob, ako vhodne zobraziť vstupné dáta pre automatické metódy a ako ich vhodne vizuálne nakonfigurovať. Využitím kombinácie by mali byť vypočítané výsledky presnejšie a výpočet by mal byť rýchlejší. Využitím vizualizácie a interakcie sa pokúsime využiť skúsenosti a vedomosti analytika, ktorý bude aplikáciu používať.

Ku kombinácii automatických a vizuálnych metód sa dá pristupovať viacerými spôsobmi. Kombinácia môže byť symetrická alebo nesymetrická. Pri symetrickej kombinácii sú oba spôsoby dolovania dát zastúpené viacmenej v rovnakej miere a vzájomne previazané. Obe metódy sú touto kombináciou vylepšené. Nesymetrická kombinácia je taká, kde sa jedna metóda využije na doplnenie druhej. Tento prístup sme zvolili aj v našej práci. Zoberieme automatickú metódu K-Means algoritmus určenú na segmentáciu dát a rozšírime ju pomocou rôznych vizualizácií a vizuálnych interakcií.

Na základe týchto úvah sa teraz pokúsime sformulovať ciele pre našu prácu.

3.1 Ciele práce

Hlavným cieľom bude snaha o rozšírenie a vylepšenie K-Means algoritmu pomocou vizuálnych metód dolovania dát. O rozšírenie sa pokúsime v rôznych fázach výpočtu algoritmu:

- V konfiguračnej fáze umožníme presnejšiu konfiguráciu vstupov a nastavení algoritmu. Pokúsime sa to docieliť vizuálnym zadávaním počiatočných stredov pre jednotlivé zhluky. To by malo výpočet algoritmu

urýchliť a priniesť presnejšie výsledky ako bežne používané náhodné počiatočné rozloženie stredov zhlukov. Vizuálne zadávanie stredov by malo uľahčiť určenie počtu zhlukov. Predpokladáme, že vďaka zobrazeniu skúmaných dát bude pre analytika jednoduchšie odhadnúť počet zhlukov, ktoré chce v dátach nájsť.

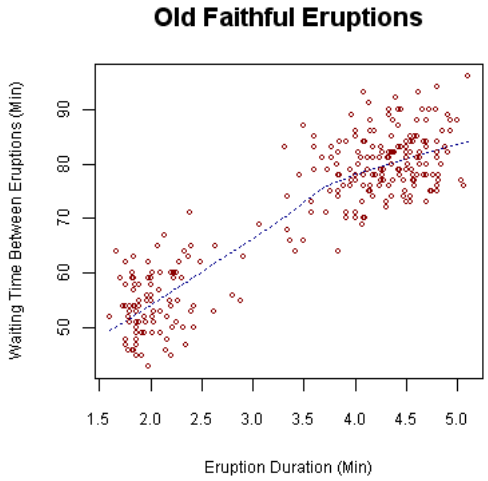
- Konfiguráciu K-Means algoritmu rozšírime aj o možnosť vizuálneho zadávania oddeľujúcich nadrovín. Jedná sa o nadroviny, ktoré vytvoria hranicu, cez ktorú "nepreskočí" dátový záznam pri výpočte algoritmu z jedného zhluku do druhého. To znamená, že pokiaľ by mal pri výpočte nejaký dátový záznam prejsť z jedného zhluku do druhého, stane sa tak iba ak medzi stredmi týchto zhlukov neexistuje oddeľujúca nadrovina.

- Vizualizáciu využijeme aj v konečnej fáze algoritmu. Vypočítané zhluky sa pokúsime prehľadne zobraziť tak, aby boli ľahko interpretovateľné a čitateľné používateľom.

Vizualizácie, ktoré použijeme na rozšírenie K-Means algoritmu budú založené na scatterplot grafe. Jedná sa o dvojdimenzionálny graf, kde sú jednotlivé záznamy umiestnené na pozícii určenej hodnotami tohto záznamu v zobrazovaných rozmeroch (Obr. 5). Viac informácii v [6] alebo [2]. Do tejto vizualizácie pridáme ďalšie grafické elementy, ktoré budú prezentovať stredy zhlukov a oddeľujúce nadroviny. Scatterplot bol zvolený pre jeho jednoduchosť a ľahké vnímanie ľudským okom. Pokúsime sa ho použiť tak, aby sme túto jeho dôležitú vlasnosť nepokazili.

Výsledkom našej práce bude analytický nástroj určený na segmentáciu dát. Počas procesu segmentácie analytik postupne definuje rôzne zhluky vstupných dát, ktoré majú nejaké zaujímavé vlastnosti. Pokúsime sa tento proces analýzy v našej aplikácii zachytiť a vizualizovať. Chceli by sme tým uľahčiť prezentáciu jednotlivých krokov a výsledky analytického procesu iným osobám, ktoré sa na analýze nepodieľali. Vhodnou vizualizáciou by sme mali uľahčiť interakciu požívateľa s jednotlivými krokmi v zachytenom analytickom procese.

Aplikácia by mala umožniť opätovné použitie vytvoreného analytického procesu na iné kompatibilné dáta. Používateľ bude môcť aplikovať celý proces na inú množinu vstupných dát a overiť si na nich analytické závery vytvorené na prvej množine dát. Opätovné použitie by malo ušetriť čas a umožniť ľahšie porovnanie podobných dátových súborov (napr. merania z rovnakého zdroja).



Obr. 5: Ukážka scatterplot vizualizácie.

3.2 Systémové požiadavky na softvérové dielo

Výsledné softvérové dielo by malo spĺňať požiadavky na moderné počítačové aplikácie. Aplikácia by mala byť spustiteľná na rôznych platformách. Modulárna architektúra by mala umožniť jednoduché rozšírenie aplikácie o ďalšiu funkcionalitu. Rozšíriteľnosť aplikácie by mala byť podporovaná aj vhodnou licenciou kompatibilnou s GNU GPL [5].

4 Riešenie

4.1 Náčrt riešenia

Po hlbšej analýze cieľov práce z časti 3.1 a počas samotnej implementácie sa ukázalo, že potrebnú funkcionalitu aplikácie môžeme rozdeliť do dvoch väčších a čiastočne nezávislých častí. Prvá časť sa zaoberá **rozšírením a vylepšením K-Means algoritmu**. Druhá časť je zameraná na **zachytenie, vizualizáciu a opätovné použitie analytického procesu**.

Vylepšenie K-Means algoritmu sme dosiahli jeho previazaním s rozšírenou scatterplot vizualizáciou. Používateľovi sme umožnili interaktívnym naklikávaním v grafe zadávať počiatočné stredy zhlukov pre výpočet algoritmu. Následne sa dajú tieto stredy jednoducho upravovať a dá sa spresňovať ich pozícia v ďalších rozmeroch. Aplikácia umožňuje vizuálne (nakreslením úsečky) špecifikovať oddeľujúce nadroviny. Po skončení výpočtu algoritmu aplikácia zobrazí vypočítané zhluky rôznymi farbami.

Druhá časť funkcionalitu aplikácie rieši vizualizáciu analytického sedenie pomocou stromu. Jednotlivé uzly stromu umožňujú interakciu používateľa s príslušnými operáciami a dátami, ktoré reprezentujú. Aplikácia zároveň umožňuje vizualizovať sedenia pomocou grafu. Jeho výhodou je ľubovoľné, používateľom definované rozmiestnenie uzlov v rovine, čo uľahčuje prezentáciu analytického sedenia.

Pre ľahšie vytváranie scatterplot vizualizácií aplikácia umožňuje zobraziť maticu scatterplotov, v ktorej používateľ vidí náhľad grafu pre všetky dvojice rozmerov. Ukážka takejto matice je na obrázku 20.

4.2 Vylepšenie K-Means algoritmu

Z dostupných automatických metód pre dolovanie dát sme si zvoli K-Means algoritmus [1]. Rozhodli sme sa tak pre jeho jednoduchosť a rozšírenosť. Úlohou algoritmu je rozdeliť vstupnú množinu dátových záznamov na n zhlukov. Pseudkód algoritmu je uvedený na obrázku 6.

Algoritmus sa počíta opakovane vo viacerých kolách, pričom v každom kole sa prepočíta vzdialenosť každého dátového záznamu vzhľadom na všetky stredy zhlukov. Podľa vypočítanej vzdialenosti sa dátový záznam priradí tomu zhluku, ku ktorému je najbližšie. Následne sa prepočítajú súradnice

stredov zhlukov tak, aby sa nachádzali v strede celého zhluku. Výpočet sa opakuje pevne zvolený počet kôl. Na konci výpočtu pre každý zhluk platí, že vzdialenosť jeho prvkov ku stredu zhluku je menšia ako ich vzdialenosť k ľubovoľnému inému stredu zhluku.

Inputs:
$I = \{i_1, ..., i_k\}$ (Instances to be clustered)
n (Number of clusters)
Outputs:
$C = \{c_1, ..., c_n\}$ (cluster centroids)
$m : I \rightarrow C$ (cluster membership)

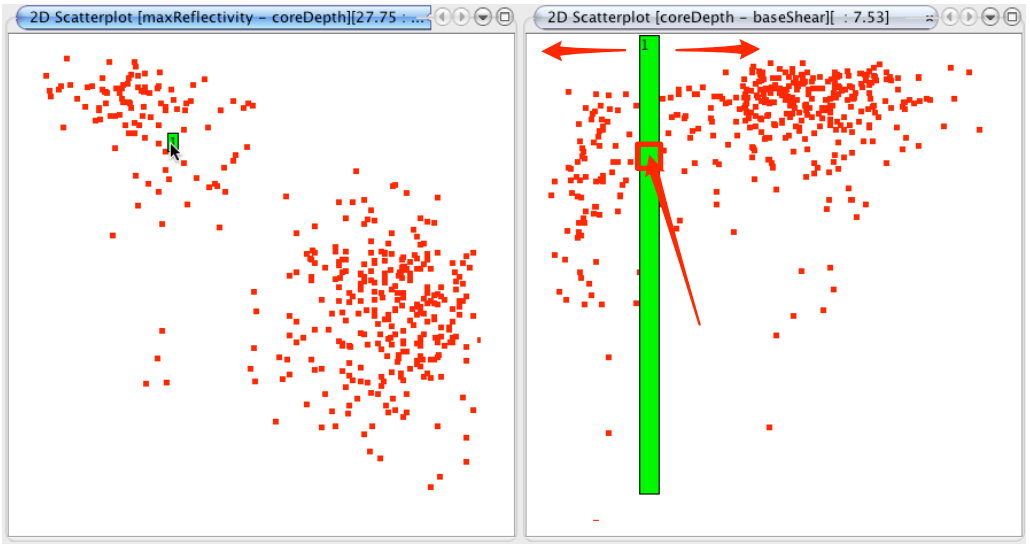
procedure **KMeans**
Set C to initial value (e.g. random selection of I)
For each $i_j \in I$
$m(i_j) = \underset{k \in \{1..n\}}{\operatorname{argmin}} distance(i_j, c_k)$
End
While m has changed
For each $j \in \{1..n\}$
Recompute i_j as the centroid of $\{i | m(i) = j\}$
End
For each $i_j \in P$
$m(i_j) = \underset{k \in \{1..n\}}{\operatorname{argmin}} distance(i_j, c_k)$
End
End
return C
End

Obr. 6: Pseudokód K-Means algoritmu.

Tento pomerne jednoduchý výpočet algoritmu sme v našej práci rozšírili o možnosť určenia počiatočnej pozície stredov zhlukov a o možnosť zadefinovania oddeľujúcich nadrovín.

4.2.1 Zadávanie stredov zhlukov

Vizualizácia, na ktorej je aplikácia postavená je dvojrozmerný scatterplot a preto aj interakcia používateľa s vizualizáciou je v dvoch rozmeroch. Zadávanie stredov pre zhluky sme však potrebovali vyriešiť tak, aby bolo možné určiť súradnice stredov vo všetkých rozmeroch. Ako vhodné riešenie sa ukázala metóda **postupného spresňovania pozície stredu zhluku**. Používateľ najprv určí pozíciu stredu v dvoch rozmeroch kliknutím do scatterplotu (Obr. 7 vľavo). Následne postupne pridáva ďalšie súradnice tohto stredu klikaním do scatterplotov s inými rozmermi. V prípade, že nejaký scatterplot zobrazuje rozmery, v ktorých má stred už určenú jednu súradnicu, tak príslušný stred v zobrazíme ako úsečku (Obr. 7 vpravo). Posúvaním tejto úsečky môže používateľ upraviť hodnotu už zadanej súradnice. Dvojklikom na úsečku sa dodefinuje chýbajúca súradnica. Kliknutím do scatterplotu, ktorý zobrazuje rozmery v ktorých upravovaný stred ešte nemá určenú ani jednu súradnicu, sa dodefinujú obidve súradnice stredu podľa miesta kliknutia. V prípade potreby sa zadané stredy zhlukov dajú pomocou kontextovej ponuky jednoducho z grafu odstrániť.



Obr. 7: Zadávanie stredov zhlukov. Vľavo stred so zadanými súradnicami v oboch zobrazovaných rozmeroch. Vpravo ten istý stred so zadanou len jednou súradnicou v zobrazovaných rozmeroch a vyznačenými možnosťami interakcie.

Pri výpočte K-Means algoritmu nemusia mať zadané všetky stredy súradnice vo všetkých rozmeroch. V takom prípade sa výpočet vykonáva len

na tých rozmeroch, v ktorých majú všetky stredy definovanú súradnicu.

4.2.2 Zadávanie oddeľujúcich nadrovín

Pri určovaní zhlukov pomocou K-Means algoritmu hrajú dôležitú úlohu **outliers dáta**. Jedná sa o také dátové záznamy, ktoré sú významne vzdialené od väčšiny ostatných záznamov. Pri analytických výpočtoch sa snažíme takéto dáta redukovať alebo ich vôbec do výpočtu nezaradiť [11].

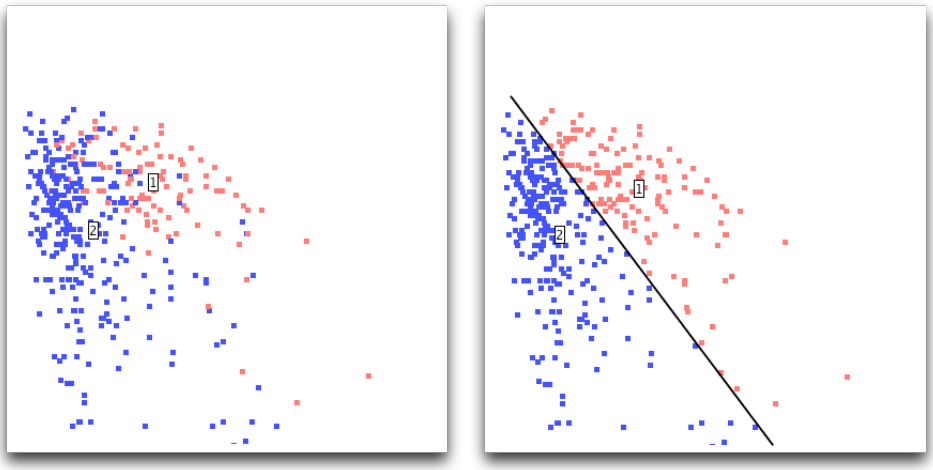
Oddeľujúce nadroviny upravujú výpočet algoritmu tak, že vo fáze kde sa rozhoduje ku ktorému stredu zhluku bude dátový záznam priradený, sa berie do úvahy aj poloha vzhľadom na oddeľujúcu nadroviny. Dátový záznam sa môže priradiť stredu zhluku iba ak neleží v opačnej polrovine ako stred zhluku vzhľadom na nejakú oddeľujúcu nadrovinu. Využitie nadrovín je hlavne pri určovaní outliers dát. Pomocou oddeľujúcej nadroviny ich používateľ môže veľmi jednoducho odrezať a nezahrnúť do výpočtu. Eliminovaním outliers dát získame presnejšie výsledky. Viac informácií o tejto téme nájdeme v [11].

Oddeľujúce nadroviny sa môžu použiť aj na vytvorenie pevnej hranice medzi dvoma zhlukmi. Môže sa stať, že po výpočte K-Means algoritmu sa nejaké dva zhluky významne prekrývajú v určitej scatterplot vizualizácií (Obr. 8 vľavo). Analytik však z praxe vie, že tieto dva zhluky by mali byť oddelené. Stačí, aby pridal jednu oddeľujúcu nadrovinu, ktorá bude ležať medzi týmito dvoma zhlukmi. Po výpočte algoritmu sú oba zhluky oddelené (Obr. 8 vpravo).

Oddeľujúce nadroviny sa v aplikácii definujú nakreslením úsečky v scatterplot vizualizácii. Táto úsečka je smerový vektor oddeľujúcej nadroviny.

4.3 Analytické sedenie

Počas práce používateľa s aplikáciou sa vytvára postupnosť krokov, ktorými sa používateľ približuje k požadovanému cieľu. Vytvorená aplikácia dokáže tieto kroky zachytiť do analytického sedania. Získali sme tak záznam ľudského uvažovania nad dátami, ktorý budeme môcť opätovne použiť na kompatibilné vstupné dáta. Dva dátové súbory sú kompatibilné ak majú rovnakú štruktúru a usporiadanie jednotlivých rozmerov. Líšiť sa môžu len v počte záznamov. Význam opätovného použitia analytického sedenia je potvrdenie alebo vyvrátenie hypotéz vytvorených na jedných dátach a aplikovaných na iné dáta. Napríklad môžme uvažovať opakované merania z rovnakého zdroja.



Obr. 8: Vľavo prekrývajúce sa zhluky. Vpravo oddelenie prekrývajúcich zhlukov pomocou oddeľujúcej nadroviny.

Používateľ spraví analýzu dát prvého merania. Odhalí niekoľko zhlukov, ktoré hovoria o nejakej významnej vlastnosti. Neskôr používateľ získa dáta z ďalšieho merania z toho istého zdroja. Tentokrát už nemusí spraviť opätovnú analýzu, ale jednoducho použije analytické sedenie z prvého merania. Následne uvidí, či aj nové dáta obsahujú podobné zhluky ako dáta z prvého merania a či si zachovali dôležité vlastnosti. V prípade, že to tak nebude, je zrejmé, že v meranom objekte nastala zmena.

Zachytené analytické sedenie sme sa pokúsili vylepšiť jeho vizualizáciou. Sprehľadnili sme tak jednotlivé kroky, ktoré používateľ vykonal a umožnili sme, aby sa toto sedenie dalo vizuálne prezentovať ďalším osobám. Vizualizácia sedenia nám zároveň zjednodušila interakciu používateľa s jednotlivými objektmi v tomto sedení. Ich popisu a analýze sa venuje ďalšia časť práce.

4.3.1 Štruktúra analytického sedenia

Analytické sedenia sa vytvára postupným pridávaním analytických krokov používateľom. Analýzou jeho štruktúry sa ukázalo, že v skutočnosti je analytické sedenie strom, ktorého uzly sú rôzne akcie používateľa a výsledky týchto akcií. Uzly, ktoré predstavujú akcie používateľa, ako napríklad selekcie alebo spustenie algoritmu sme nazvali **filtrovacie uzly**. Filtrovacie preto, lebo z rodičovských dát vytvárajú použitím nejakého algoritmu (filtra) dcérske dáta. Druhý typ uzlov sú **dátové uzly**, ktoré predstavujú fyzické alebo

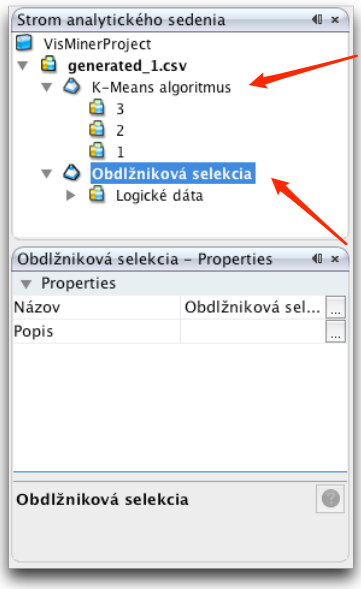
logické dáta, ktoré vznikli nejakou akciou v analytickom sedení. Každý filtrovací uzol spolu so svojimi dcérskymi dátovými uzlami predstavuje logický celok: akcia a jej výsledky. Tieto dva typy uzlov sa v strome vždy nachádzajú spolu. Aplikácia umožní odstrániť len filtrovacie uzly, spolu s ktorými sa automaticky odstránia aj ich dátové uzly. Celý strom sedenia sme zakotvili do špeciálneho uzla, ktorý predstavuje aktuálny analytický projekt. V ďalšej časti textu popíšeme jednotlivé konkrétne druhy uzlov, ktoré sa môžu nachádzať v analytickom sedení našej aplikácie.

Filtrovací uzol predstavuje analytickú operáciu nad dátami. Naša aplikácia umožňuje dva druhy operácie nad dátami, a preto sa môžu v sedení nachádzať dva druhy filtrovacích uzlov a to obdĺžniková selekcia a K-Means algoritmus. Príklady obidvoch nájdeme na obrázku 9.

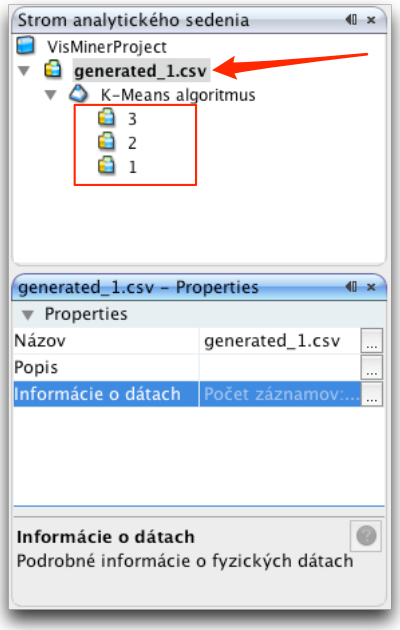
- **Obdĺžniková selekcia** je uzol, ktorý vznikne ak používateľ vytvorí logické dáta zo selekcie. Tento uzol môže mať len jedného syna a to sú logické dáta, ktoré daná selekcia definuje. Obdĺžniková selekcia predstavuje kritérium vo forme dvojrozmerného intervalu v dvojrozmernom podpriestore pôvodného priestore. Záznamy, ktoré toto kritériu spĺňajú patria do dcérskych logických dát.
- **K-Means algoritmus** je uzol, ktorý vznikne aplikovaním algoritmu na dáta. Tento uzol má práve toľko synov, koľko vypočítal K-Means algoritmus zhlukov, teda každý syn tohto uzla reprezentuje jeden vypočítaný zhluk.

Dátový uzol predstavuje skupinu dátových záznamov. Tieto uzly umožňujú pokračovať v sedení buď pomocou nejakého algoritmu alebo vytvorením ďalších logických dát selekciou. V sedení sa môžu nachádzať dva druhy dátových uzlov, a to fyzické alebo logické dáta.

- **Fyzické dáta** predstavujú dáta, ktoré sú naimportované z fyzického súboru. V sedení sa môžu vyskytovať len raz a tvoria spolu jediné dcérske dáta uzla pre projekt. Aplikácia štandardne pomenuje tento uzol podľa súboru, z ktorého boli dáta importované. Ná obrázku 10 ukazuje šípka na fyzické dáta umiestnené v strome analytického sedenia.
- **Logické dáta** predstavujú dáta, ktoré vznikli aplikovaním analytickej operácie na iné dáta, a to buď aplikovaním selekcie alebo nejakého algoritmu. V prídade K-Means algoritmu sú logické dáta vypočítané zhluky (Obr. 10 v červenom obdĺžniku).



Obr. 9: Ukážka filtrovacích uzlov v strome analytického sedenia. Horná šípka ukazuje na K-Means algoritmus. Dolná šípka ukazuje na obdĺžnikovú selekciu dát.



Obr. 10: Ukážka dátových uzlov v strome analytického sedenia. Šípka ukazuje na fyzické dáta a v obdĺžniku sú zobrazené logické dáta.

Okrem spomenutých uzlov existuje v sedení ešte jeden špeciálny uzol, ktorý predstavuje aktuálny projekt. Tento uzol spolu s fyzickými dátami tvorí koreň a pod nimi sú zavesené kombinácie ostatných dátových a filtrovacích uzlov. Projektový uzol je vytvorený pre interakciu s celým projektom, ktorú popíšeme neskôr.

4.3.2 Vizualizácia sedenia

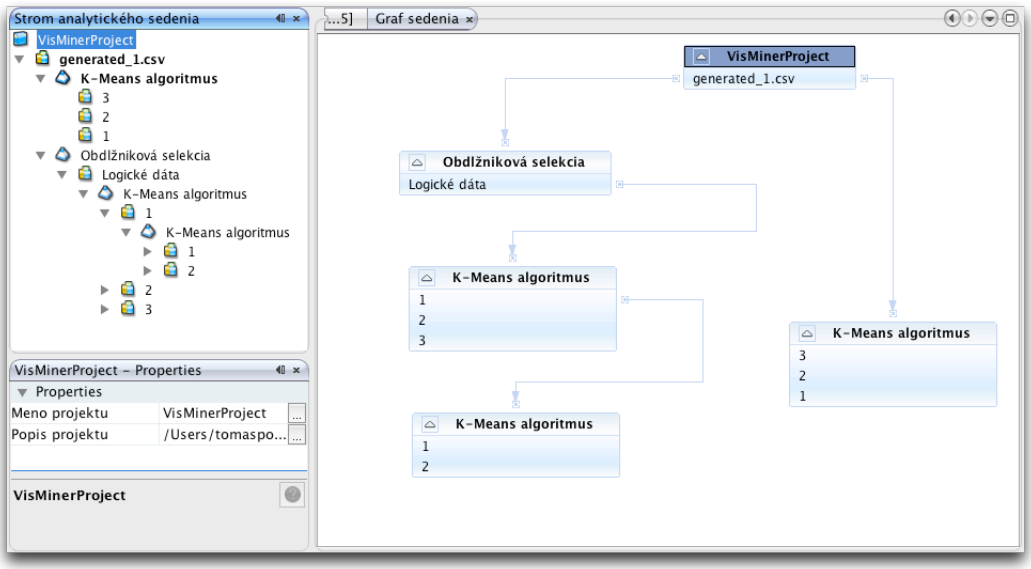
Analytické sedenie má stromovú štruktúru, preto ho bolo vhodné vizualizovať primárne pomocou stromu. Každý uzol v strome, okrem uzla pre projekt, predstavuje dátový alebo filtrovací uzol. Ich význam sme popísali v časti 4.3.1. Pre ľahšie identifikovanie typu uzla má každý vlastnú ikonu (Obr. 11). Jednotlivé uzly sedenia sa dajú rozložiť alebo zložiť a spraviť tak viac miesta pre tie časti sedenia, na ktoré sa používateľ momentálne sústredí.

Obr. 11: Použité ikony pre jednotlivé typy uzlov v strome analytického sedenie. Z ľava: uzol projektu, dátový uzol a filtorvací uzol.

Aplikácia umožňuje ešte jednu vizualizáciu sedenia. Jedná sa o rovinný graf, kde jednotlivé vrcholy predstavujú filtrovacie uzly sedenia spolu s ich logickými dátami, ktoré definujú. Špeciálny prípad tvoria fyzické dáta, tie nemajú žiadny nadriadený filtrovací uzol, preto sú zobrazené spolu s vrcholom pre projekt. Ukážku vizualizácie nájdeme na obrázku 12. Grafová vizualizácia bola do aplikácie pridaná hlavne pre prezentačné účely. Používateľ si môže sám rozmiestniť a upraviť jednotlivé uzly podľa vlastného uváženia a potrieb pre prezentáciu.

4.3.3 Interakcia so sedením

Používateľ môže interagovať s jednotlivými uzlami v strome sedenia cez kontextovú ponuku, ktorú vyvolá kliknutím pravého tlačidla myši na príslušný uzol. Pre každý uzol môže používateľ okrem špecifických vlastností, ktoré popíšeme neskôr, upravovať meno uzla a jeho popis. Tieto dve vlastnosti slúžia pre grafickú reprezentáciu uzlov v strome a grafe sedenia. Všetky vlastnosti



Obr. 12: Ukažka grafového zobrazenia analytického sedenia a prisluchajúceho stromu.

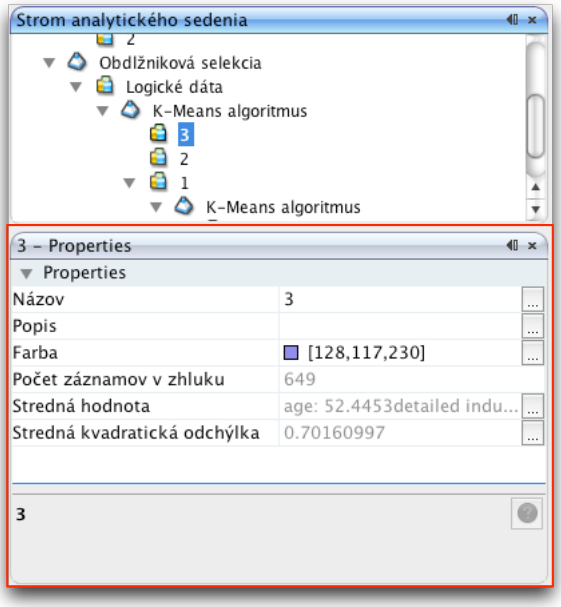
uzla sa dajú prezerať a upravovať cez okno vlastností (Properties). Pre zobrazenie vlastností musí byť najprv uzol selektovaný (Obr. 13). Každý uzol v strome sedenia sa dá z neho vymazať. Pri mazaní sa vymažú aj všetky dcérske uzly. V ďalšom texte popíšeme špecifické vlastnosti pre jednotlivé uzly a operácie, ktoré možno na týchto uzloch vykonávať.

Špecifické vlastnosti a operácie pre jednotlivé typy uzlov:

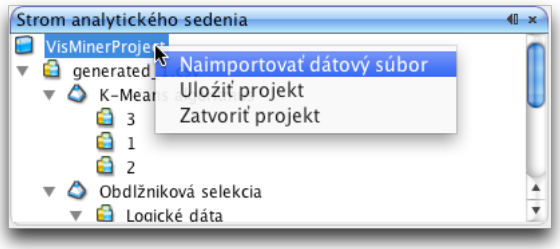
Projekt: Ponuka tohto uzla (Obr. 14) umožňuje uložiť aktuálny projekt do projektového adresára. Po uložení projektu sa zobrazí okno s informáciou o úspešnom uložení. Uzol taktiež umožňuje zavrieť aktuálny projekt a založiť nový projekt. Pri zakladaní projektu treba zvoliť adresár, kam sa budú ukladať všetky súbory projektu. Pomocou uzla projektu sa dajú do sedenia naimportovať fyzické dáta.

Fyzické dáta: Uzol umožňuje aplikovať aktuálne sedenie na iné kompatibilné dáta. Po načítaní kompatibilného dátového súboru sa postupne aplikujú všetky operácie zo sedenia a zobrazia sa tie isté vizualizácie ako pri pôvodných dátach.

Logické dáta: Po označení tohto uzla v strome sedenia aplikácia automaticky zvýrazní vo všetkých scatterplot vizualizáciach len tie dátové zá-

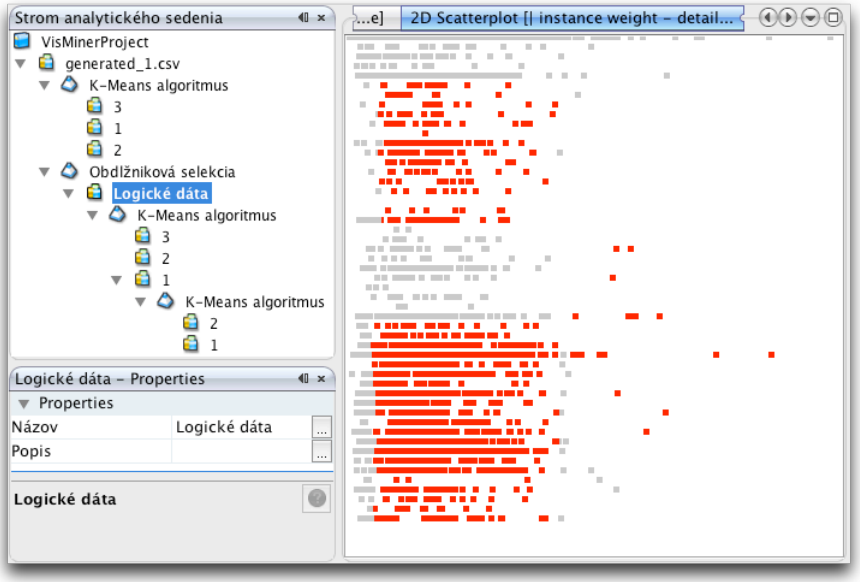


Obr. 13: Zobrazenie zoznamu vlastností pre selektovaný uzol vypočítaného zhluku z K-Means algoritmu.



Obr. 14: Kontextová ponuka uzla pre projekt.

znamy, ktoré patria do týchto logických dát (Obr. 15).



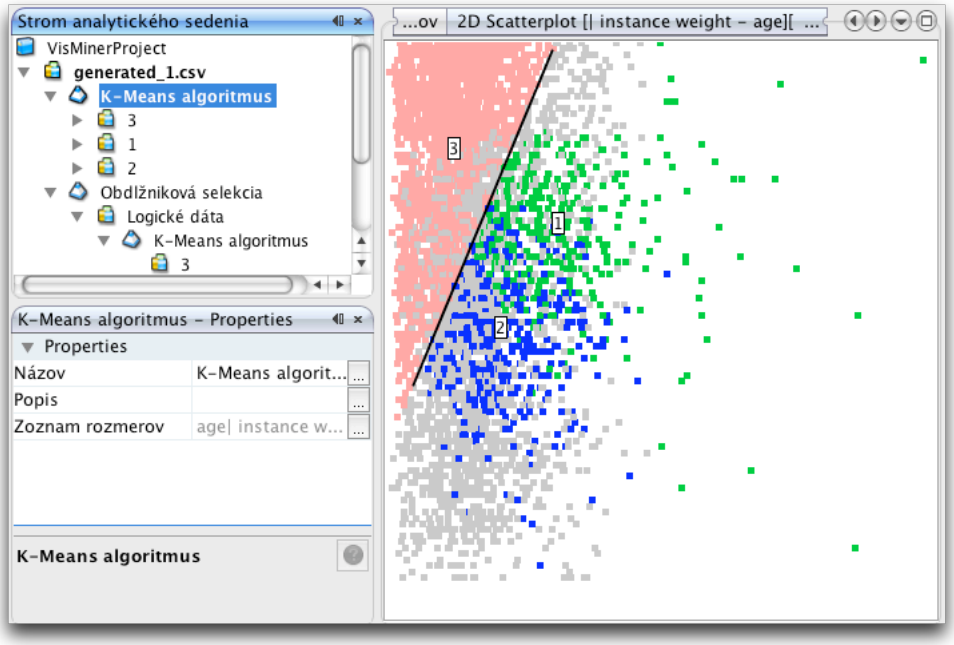
Obr. 15: Po označení logických dát v strome sedenia sa vo všetkých scatterplot vizualizáciach zvýraznia dáta patriace do týchto logických dát.

Obdĺžniková selekcia: Po označení tohto uzla aplikácia zobrazí obdĺžnikové oblasti selekcie (Obr. 16). Používateľ môže týmto oblastiam meniť veľkosť a typ. Po zmene selekcie sa automaticky prepočítajú všetky uzly zavesené pod selekciou a aktualizujú sa všetky vizualizácie. To zahŕňa aj prepočítanie všetkých K-Means algoritmov, ktoré sú v strome sedenia pod týmto uzlom.

K-Means algoritmus: Ponuka uzla umožňuje spustiť výpočet algoritmu. Pokiaľ s vypočítanými používateľ nie je spokojný, môže cez ponuku uzla výsledky algoritmu vymazať a umožniť tak opätovné zadávanie stredov zhlukov a oddeľujúcich nadrovín. Po označení tohto uzla sa zobrazia vo všetkých scatterplot vizualizáciach zadané stredy zhlukov a oddeľujúce nadroviny. V prípade, že je algoritmus už vypočítaný, farebne sa zvýraznia aj všetky vypočítané zhluky (Obr. 17).

Vypočítaný zhluk: Je to špeciálny typ logických dát. Vzniká ako výsledok K-Means algoritmu a má správanie a interakciu ako bežné logické dáta. Okrem toho má niekoľko ďalších vlastností (Obr. 13). Uzol umožňuje používateľovi meniť farbu bodov záznamov patriacich do zhluku.

Obr. 16: Úprava oblastí obdĺžnikovej selekcie. Po zmene selekcie aplikácia automaticky prepočíta K-Means algoritmy zavesené pod selekciou.



Obr. 17: Zobrazenie vypočítané K-Means algortimu s troma zhlukmi.

Zmena farby sa automatický prejaví vo všetkých scatterplot vizualizáciach. Medzi vlastnosťami uzla sa zobrazujú aj niektoré dôležité štatistické údaje o zhluku. Jedná sa o **počet záznamov** patriacich do zhluku, **stredná hodnota** a **stredná kvadratická odchýlka**. Podľa [17] a [9] môžeme strednú kvadratickú odchýlku použiť na určenie kvality zhluku. Čím je menšia, tým je vypočítaný zhluk hustejší a môžeme ho považovať za lepší. V aplikácii sa stredná kvadratická odchýlka počíta z normalizovaných hodnôt.

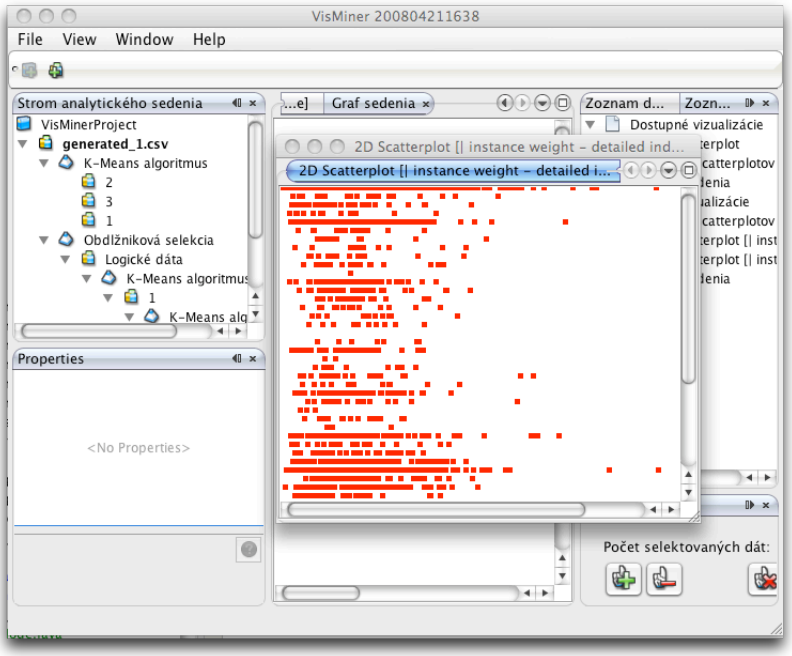
4.4 Ďalšie funkcie aplikácie

Založenie projektu: Pred tým, ako môže analytik vykonať akúkoľvek zmysluplnú činnosť s aplikáciou, musí najprv založiť projekt. Pri zakladaní projektu sa určí adresár, v ktorom sa zhromažďujú všetky konfiguračné a dátové súbory potrebné pri práci s aplikáciou. V tomto adresári sa nachádza aj súbor projektu. Jedná sa o XML súbor, ktorý obsahuje všetky informácie potrebné pre opätovné vybudovanie analytického sedenia. Pri načítaní projektu sa vypočítajú všetky operácie použité v uloženom sedení a zobrazia sa tie vizualizácie, ktoré boli otvorené pri jeho ukladaní. Ak už je v aplikácii nejaký projekt načítaný, tak pred

otvorením iného projektu treba aktuálny projekt zatvoriť. Ukladanie projektu nie je automatické, používateľ si sám zvolí, kedy sa má projekt uložiť.

Import dát: Po vytvorení projektu treba naimportovať súbor s fyzickými dátami. Viac o podporovaných dátach v časti 4.5 Vstupy a výstupy aplikácie. Po naimportovaní dát sa vytvorí uzol s fyzickými dátami, na ktorý sa môžu následne aplikovať filtre a selekcie.

Správa okien: Aplikácia umožňuje širokú manipuláciu s oknami grafického rozhrania. Používateľ si môže okná ľubovoľne rozmiestňovať, pridávať a odoberať. Každé okno sa dá od hlavného okna aplikácie odpojiť a nechať ho plávať (Obr. 18). Vďaka týmto vlastnostiam si môže každý používateľ prispôsobiť grafické rozhranie aplikácie podľa vlastných potrieb a zvýšiť tak efektivitu práce.



Obr. 18: Plávajúce okno vizualizácie.

4.5 Vstupy a výstupy aplikácie

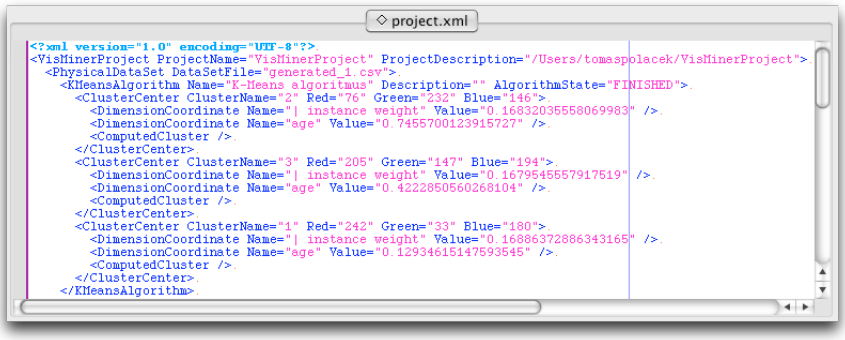
Vstupy aplikácie: Aplikácia podporuje import dát zo známeho a rozšíreného CSV (Comma Separated Values) formátu. Aplikácia vie pracovať len

s číselným rozmermi. Ak vstupné dáta obsahujú aj iné ako číselné rozmery, aplikácia ich bude ignorovať. Rozšírenie aplikácie pre prácu s inými formátmi súborov a dátovými typmi nie je vďaka modulárnej architektúre problém.

Výstupy aplikácie: Aplikácia generuje vizuálne výstupy a súbor projektu.

Vizuálne výstupy sú graf sedenia, strom sedenia a scatterplot vizualizácie. Graf sedenia a scatterplot vizualizácie vie aplikácia vyexportovať do súboru s obrázkom a ten sa môže následne vytlačiť. Graf sedenia sa dá použiť napríklad na prezentáciu výsledkov inej osobe a ozrejmenie postupov ako sa k výsledkom podarilo dostať.

Ďalším výstupom aplikácie je súbor projektu v XML formáte. Vzhľadom na použitý formát sa môže tento súbor ľahko použiť ako vstup do iných aplikácií. Súbor sa dá ľahko priamo upraviť pomocou vhodného editora. Obrázok 19 zobrazuje ukážku projektového súboru.



Obr. 19: Ukážka z projektové súboru v XML formáte.

4.6 Implementačné detaily

Architektúra aplikácie: Aplikácia bola navrhnutá a naprogramovaná tak, aby mala modulárnu architektúru. Jednotlivé vizualizácie a algoritmus K-Means sú naprogramované ako samostatné moduly. Aplikácia môže byť priamočiaro rozšírená o ďalšie vizualizácie a algoritmy. Jednou z možností je napríklad pridanie vizualizácií využívajúcich OpenGL knižnicu a hardvérovú akceleráciu.

Použité technológie: Aplikácia je naprogramovaná v programovacom jazyku Java (www.java.sun.com) a postavená na otvorenej platforme určenej na tvorbu desktopových aplikácií: NetBeans (netbeans.org). Vďaka použitej platforme sme sa mohli plne sústrediť na riešenie problémov aplikácie a nemuseli sme sa zaoberať takými funkciami ako je správa okien, riešenie lokalizácie aplikácie, nízkoúrovňové vykresľovanie grafov atď. Vďaka použitým technológiám je aplikácia multiplatformná a úspešne odskúšaná na platformách:

- Mac OS X (http://www.apple.com/macosx)
- GNU Linux (http://www.gnu.org)
- Microsoft Windows XP(http://www.microsoft.com/windows)

5 Prípad použitia aplikácie

V tejto kapitole si ukážeme použitie aplikácie na reálnych dátach. Popíšeme postup, ako nájdeme vo vstupných dátach dva zhluky, ktoré sa pokúsime vylepšiť odstránením outliers dát. Výsledné analytické sedenie potom aplikujeme na komapitibilné dáta z rovnakého zdroja. Výsledný projekt spolu s dátami sa nachádza na priloženom dátovom nosiči s aplikáciou v adresári "Ukážkový Projekt". Detailný popis ako používať aplikáciu je v používateľskom manuály, ktorý sa nachádza na priloženom dátovom nosiči.

5.0.1 Vstupné dáta

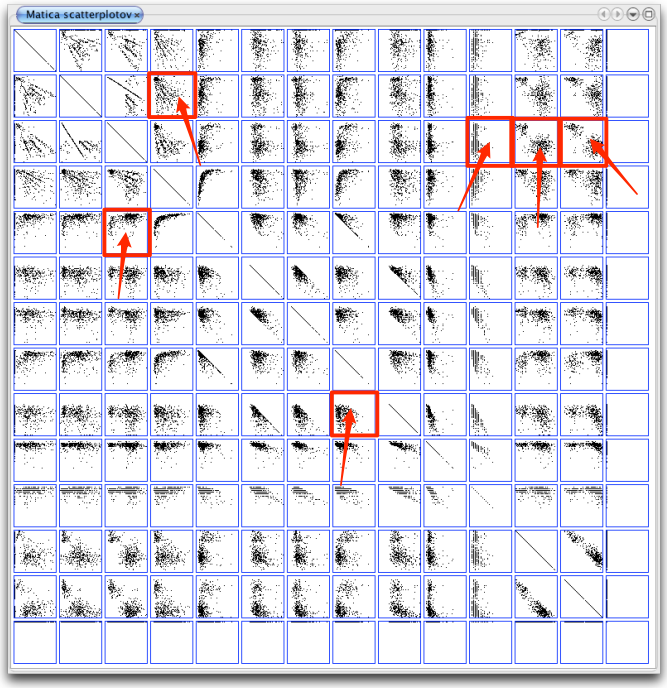
Dáta, ktoré použijeme v ukážke obsahujú hodnoty získané zo snímkov Dopplerovho radaru určeného pre meteorologické merania. Dátový súbor sme získali zo stránky so zadaním školského projektu týkajúceho sa segmentácie dát [4]. Dátový súbor pôvodne obsahoval okolo tisíc záznamov, ale pre naše potreby sme vytvorili dva dátové súbory, do ktorých sme náhodne povyberali päťsto záznamov z pôvodného dátového súboru. Takto chceme simulovať dve kompatibilné množiny dát z rovnakého zdroja. Všetky použité dátové súbory sa nachádzajú na priloženom dátovom nosiči v adresári "Data/mesocyclone".

5.0.2 Vytvorenie projektu a import dát

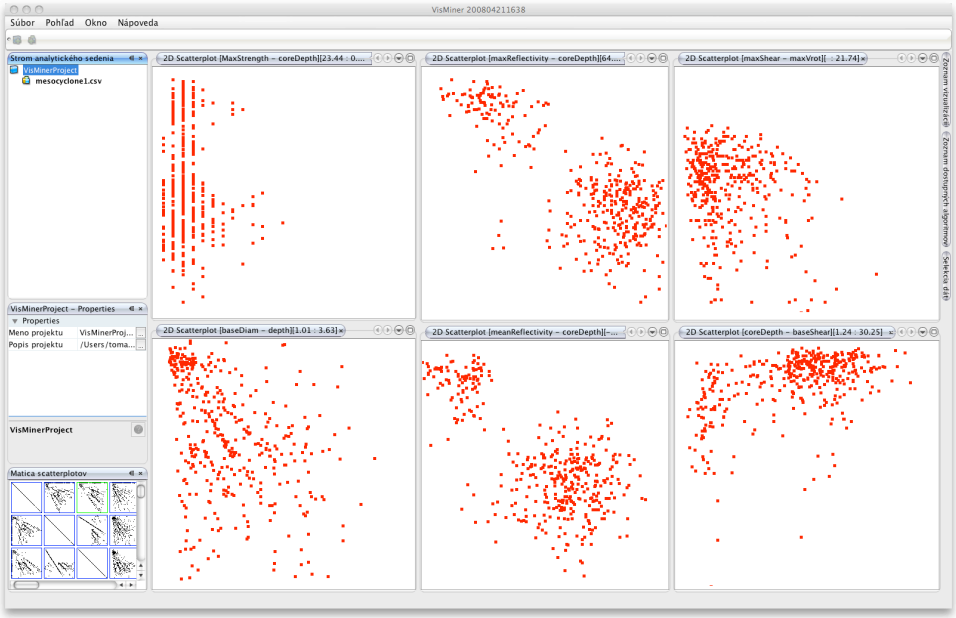
Po spustení aplikácie vytvoríme nový projekt. Ako projektový adresár zvolíme nejaký prázdny adresár na lokálnom disku. Následne naimportujeme dátový súbor "mesocyclone1.csv". Po načítaní dát si otvoríme vizualizáciu "Matica scatterplotov" zo zoznamu dostupných vizualizácií. Mala by vyzerať podobne ako na obrázku 20. Červeným obdĺžnikom sú na obrázku označené tie vizualizácie, ktoré si treba pootvárať. Môžme tak spraviť dvojklikom na príslušný náhľad scatterplotu v matici. Pootvárané vizualizácie si rozmiestnime tak, aby sme mali dobrý prehľad o otvorených oknách (Obr. 21) . Projekt si môžeme priebežne uložiť.

5.0.3 Nájdenie zhlukov

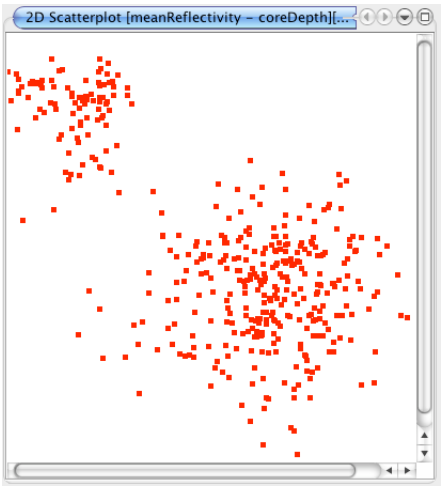
Pokúsime nájsť zhluky. Už od prvého pohľadu na dáta v matici scatterplov (Obr. 20) je zrejme, že dáta obsahujú dva hlavné zhluky. Dobre ich vidieť napríklad v scatterplot vizualizácii s rozmermi "meanReflectivity" a "coreDepth" (Obr. 22). Algoritmus budem konfigurovať tak, aby počítal dva zhluky. Do spomínanej vizualizácie umiestnime dva stredy zhlukov. Postupne zadefinujeme súradnice stredov aj do ďalších rozmerov, ktorých vizualizácie



Obr. 20: Matica sctterplotov pre ukážkové dáta.



Obr. 21: Aplikácia s vizualizáciam ukážkových dát.



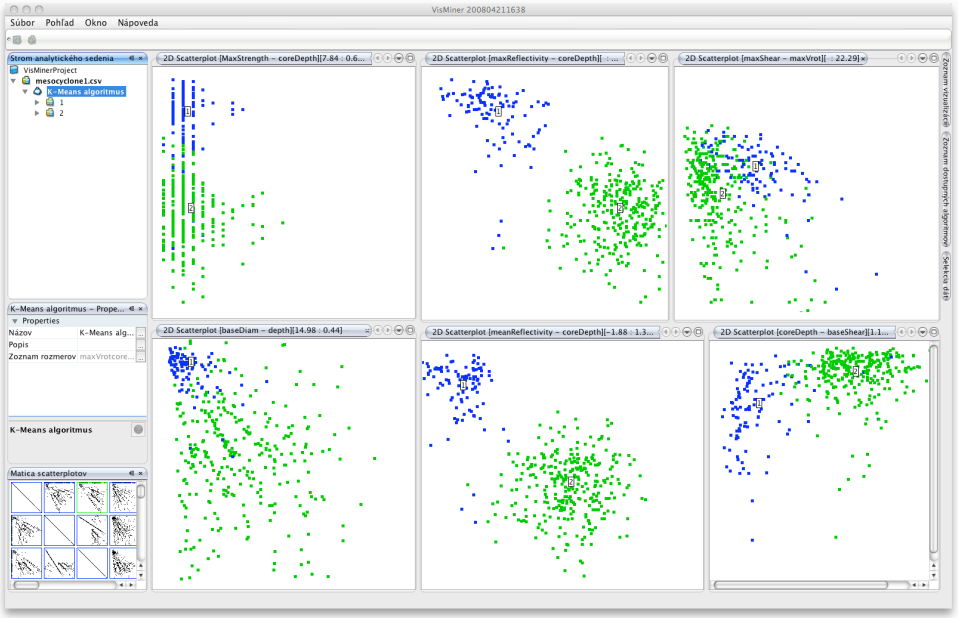
Obr. 22: V skúmaných dátach sú rozpoznateľné dva zhluky.

máme pootvárané. Následne môžeme spustiť výpočet algoritmu. Výsledok bude podobný ako na obrázku 23.

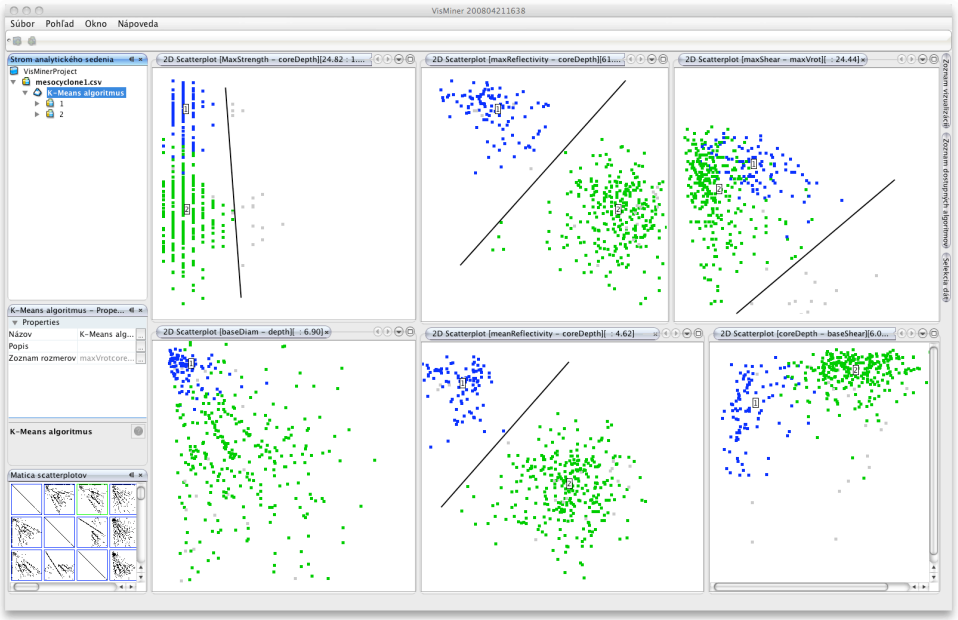
Pozrieme sa na štatistické vlastnosti vypočítaných zhlukov. Pre zhluk 1 má stredná kvadratická odchýlka hodnotu 0.16520947. Pre zhluk 2 má stredná kvadratická odchýlka hodnotu 0.52752472. Pokúsime tieto hodnoty vylepšiť. Do konfigurácie algortimu pridáme niekoľko oddeľujúcich nadrovín tak, aby sme z výpočtu odstránili outliers dáta. Oddeľujúcu nadrovinu môžeme pridať aj na tie miesta, kde sa vypočítané zhluky prekrývali. Vytvoríme tým pevnú hranicu medzi zhlukmi v danej rovine. Necháme opäť zbehnúť výpočet algoritmu (Obr. 24). Pozrieme sa na hodnoty strednej kvadratickej odchýlky. Pre zhluk 1 to je hodnota 0.11643730 a pre zhluk 2 to je hodnota 0.47674487. To znamená, že pridaním oddeľujúcich nadrovín sa nám podarilo vypočítané zhluky vylepšiť vzhľadom na použitú mieru.

5.0.4 Opätovné použitie analytického sedenia

Pokúsime sa aplikovať vytvorené analytické sedenie na kompatibilné dáta zo súboru "mesocyclone2.csv". Nájdeme ho v rovnakom adresári na priloženom dátovom nosiči ako prvé skúmané dáta. Po ich načítaní a aplikovaní sedenia preskúmame vypočítané zhluky. Hodnoty stredných kvadratických odchýlok sú 0.11125163 pre zhluk 1 a 0.47957610 pre zhluk 2. Tieto hodnoty sú veľmi podobné s hodnotami vypočítanými na prvých ukážkových dátach. Na zá-



Obr. 23: Vypočítané zhluky v ukážkových dátach.



Obr. 24: Vypočítané zhluky v ukážkových dátach s pridaním oddeľujúcich nadrovín.

klade tejto podobnosti môžeme usúdiť, že obidva skúmané súbory majú veľmi podobnú štruktúru a charakter dát.

6 Záver

V práci sa nám podarilo ukázať, že kombinácia automatických a vizuálnych metód dolovania dát prináša pozitívne výsledky. Upravenie K-Means algoritmu o možnosť vizuálneho zadávania stredov zhlukov a oddeľujúcich nadrovín viedlo k lepším štatistickým vlastnostiam vypočítaných zhlukov. Vizualizácia procesu konfigurovania algoritmu umožňuje analytikovi využiť svoj trénovaný zrak a skúsenosti v aplikačnej doméne na dosiahnutie presnejších výsledkov. Vďaka náhľadu na dáta cez scatterplot vizualizácie vie používateľ odhadnúť počet zhlukov, ktoré sa v dátach nachádzajú a nemusí ich počet odhadovať experimentovaním. Urýchlil sa tak výpočet algoritmu a tým aj celá analýzu dát.

Podľa [11] by eliminovanie outliers dát malo zlepšiť výsledky K-Means algoritmu. V aplikácii sme umožnili používateľovi vizuálnym spôsobom zadávať oddeľujúce nadroviny a odkrajovať tak outliers dáta z množiny použitej na výpočet algoritmu. Táto metóda viedla k zlepšeniu štatistických vlastností vypočítaných zhlukov. Skvalitnenie výpočtu sme pozorovali na hodnotách strednej kvadratickej odchýlky zhlukov, ktorú sa nám podarilo zredukovaním outliers dát znížiť.

K-Means algoritmus a jeho výsledky sú súčasťou analytického procesu, ktorý prebieha pri skúmaní dát. Jeho analýzou sa nám podarilo odhaliť stromovú štruktúru a niekoľko zaujímavých vlastností. Tento proces predstavuje tok ľudských myšlienok nad dátami. Skladá sa z viacerých menších krokov, ako sú napr. import dát, selekcie, algoritmy dolovania dát, vizualizácie atď. V aplikácii umožňujeme zachytiť jednotlivé kroky používateľa do analytického sedenia a opätovne ho použiť na iné kompatibilné dáta. Analytik tak získal možnosť vytvorenia "vlastnej" automatickej metódy dolovania dát.

Dôležitou súčasťou vytvorenej aplikácie je aj vizualizácia analytického sedenia. Jeho vizuálna podoba umožňuje lepšie pochopenie významu jednotlivých krokov analýzy dát. Jednoduchšia je aj prezentácia výsledkov, popri ktorých vidí analytik aj spôsob, ako boli dosiahnuté a preto sa stávajú dôveryhodnejšie. Vďaka vizualizácii sme umožnili používateľovi pristupovať k jednotlivým krokom v sedení a meniť ich parametre.

Pri programovaní aplikácie sme potrebovali vyriešiť, ako budeme počítať K-Means algoritmus pre neúplne zadané stredy zhlukov. Jedná sa o také stredy, ktoré majú zadané súradnice len v niektorých rozmeroch ale nie vo všetkých. Rozhodli sme sa pre počítanie algoritmu na takej podmnožine roz-

merov, v ktorých majú všetky stredy zhlukov zadané súradnice. Toto riešenie sa ukázalo ako prijateľné. Používateľ aplikácie si má možnosť pred spustením výpočtu K-Means algoritmu overiť, na ktorých rozmeroch bude výpočet prebiehať. Inou alternatívou rišenia tohto problému by bolo nepovoliť výpočet pokiaľ používateľ nezadefinuje súradnice stredov vo všetkých rozmeroch. Tento prístup by však bol príliš obmedzujúci.

Spomínané prínosy práce potvrdzuje aj ukážkový projekt, ktorý sme odprezentovali v 5. kapitole. Na reálnych dátach sme ukázali, ako jednoducho vizuálne zadefinovať stredy zhlukov pre K-Means algoritmus. Vypočítané zhluky sme vylepšili odstránením outliers dát pomocou oddeľujúcich nadrovín. Vytvorené analytické sedenie sme úspešne použili na iné kompatibilné dáta. Ukázali sme tak, že oboje skúmané dáta majú podobnú štruktúru a vlastnosti.

Na záver možno skonštatovať, že ciele, ktoré sme si kládli v časti 3.1 sa nám podarilo splniť. Modulárna architektúra vytvorenej aplikácie umožňuje jej jednoduché rozšírenie. Vhodné by bolo dorobiť podporu pre iné formáty dátových súborov, prípadne možnosť importu dát priamo z relačných databáz. Iná možnosť rozšírenia je pridanie vizualizácií, založených na hardvérovo akcelerovanom vykresľovaní. Podarilo by sa tak eliminovať performančné problémy, ktoré sa prejavujú pri väčšej množine vstupných dát.

7 Prílohy

7.1 CD-ROM nosič

Na priloženom dátovom nosiči nájdete:

1. Návod na inštaláciu a spustenie aplikácie.
2. Používateľsky manuál.
3. Spustiteľnú aplikáciu vo verziách pre platformy: MacOS X, GNU Linux, Microsoft Windows.
4. Zdrojové kódy aplikácie.
5. Ukážkové projekty a dátové súbory.

Literatúra

[1] Http://en.wikipedia.org/wiki/K-means - popis k-means algoritmu.

[2] http://en.wikipedia.org/wiki/scatterplot - popis scatterplot vizualizácie.

[3] Http://mathworld.wolfram.com/K-MeansClusteringAlgorithm.html - popis k-means algoritmu.

[4] http://www.cs.uah.edu/∼jrushing/cs696-summer2004/project2.html - stránka s dátami pre ukážkový projekt.

[5] Http://www.gnu.org/licenses/gpl.html - popis licencie gnu gpl.

[6] http://www.statsoft.com/textbook/glosfra.html?gloss.html&1 - popis scatterplot vizualizácie.

[7] J. Abello and J. Korn. Mgv: A system for visualizing massive multi-digraphs. Transactions on Visualization and Computer Graphics, 2001.

[8] Petr Berka. *Dobývaní znalostí z databází.* ACADEMIA, 2003.

[9] Michael J. A. Berry and Gordon Linoff Author. *Data Mining Techniques: For Marketing, Sales, and Customer Support.* Wiley, 1997.

[10] Milan Schmotzer František Sudzina. Čo by mal vedieť manažér o výbere softvéru na data mining.

[11] Ville Hautamäki, Svetlana Cherednichenko, Ismo Käarkkäainen, Tomi Kinnunen, and Pasi Fränti. Improving k-means by outlier removal. In *SCIA05*, volume 3540, pages 978–987, 2005.

[12] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31:264–323, 1999.

[13] B. Johnson and B. Shneiderman. Treemaps: A space-filling approach to the visualization of hierarchical information. In *Proc. Visualization '91 Conf*, pages 284–291, 1991.

[14] Daniel A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[15] R. Kosara, H. Hauser, and D. Gresh. An interaction view on information visualization. *State-of-the-Art Proceedings of EUROGRAPHICS 2003*, pages 123–137, 2003.

[16] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring n- dimensional databases. In *Visualization '90*, pages 230–239, San Francisco, 1990.

[17] Csaba Legány, Sándor Juhász, and Attila Babos. Cluster validity measurement techniques. In *AIKED'06: Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 388–393, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS).

[18] N. Lopez, M. Kreuseler, and H. Schumann. A scalable framework for information visualization. Transactions on Visualization and Computer Graphics, 2001.

[19] Matej Novotný. Visual abstraction for information visualization of large data. Master's thesis, Fakulta Matematiky Fyziky a Informatiky, Bratislava, 2006.

[20] Thomas Rongitsch. Information visualization and data minig - a comparison and integration. Master's thesis, Technischen Universität, Wien, 2005.

[21] D. Tang, C. Stolte, and P. Hanrahan. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. Transactions on Visualization and Computer Graphics, 2001.

[22] Kurt Thearling. An overview of data mining techniques. http://www.thearling.com/text/dmtechniques/dmtechniques.htm.