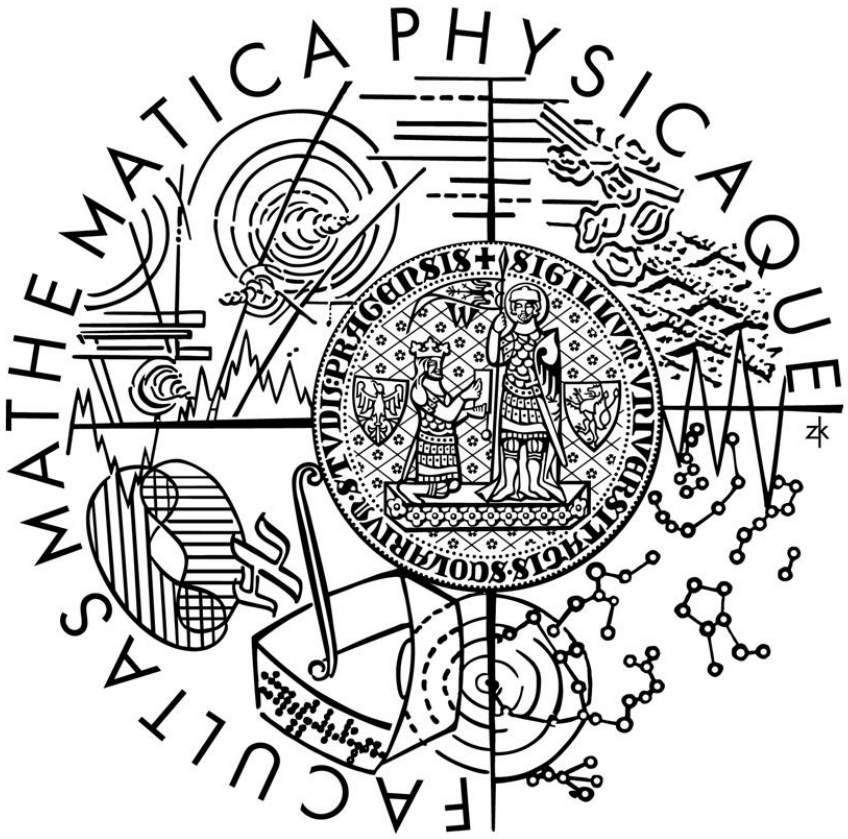
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Bc. Michal Marko

Zhluková analýza dynamických dát

Kabinet software a výuky informatiky

Vedoucí diplomové práce: *RNDr. František Mráz, CSc.*

Studijní program: *Informatika, softwarové systémy*

Poďakovanie:

Na tomto mieste by som rád poďakoval vedúcemu mojej práce, RNDr. Františkovi Mrázovi, CSc., za jeho prístup, podnetné pripomienky, cenné rady a pomoc v celom priebehu písania tejto diplomovej práce. Zároveň by som rád poďakoval celej svojej rodine, kamarátom a v neposlednom rade svojej partnerke za podporu a pochopenie.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 10.12.2010

Bc. Michal Marko

Obsah

Název práce: *Zhluková analýza dynamických dát*
Autor: *Bc. Michal Marko*
Katedra: *Kabinet software a výuky informatiky*
Vedoucí diplomové práce: *RNDr. František Mráz, CSc.*
e-mail vedoucího: *Frantisek.Mraz@mff.cuni.cz*

Abstrakt: Cieľom práce je zvoliť, prípadne navrhnúť vlastné modifikácie metód zhlukovej analýzy tak, aby bolo možné sledovať vývoj dynamických dát a ich zhlukov v čase. Vybrané metódy sú následne aplikované na reálne dáta. Dynamické dáta sú informácie tvorené sériou, v čase po sebe idúcich dátových množín, popisujúcich rovnaké vlastnosti danej množiny objektov. Problém aplikácie klasických zhlukovacích algoritmov na takéto dáta je v absencii zachovania súvislostí medzi zhlukovaniami jednotlivých dátových množín zo série, čo je možné ilustrovať ich nevhodným priebehom na umelých dátach. Rozoberáme princíp navrhnutých modifikácií a porovnávame ich výstupy na spomenutých dátach. Aby sme nami navrhnuté metódy mohli aplikovať na reálne dáta, je nutné otestovať vhodnosť ich použitia na viacrozmerných dátach, a z toho dôvodu si postupne vytvoríme až desaťrozmerné dáta. K porovnaniu efektívnosti týchto modifikácii na dátach o viacerých dimenziách slúži vlastný spôsob klasifikácie. Nami overené metódy následne aplikujeme na reálne dáta, ilustrujeme a zhodnotíme výsledky.

Klíčová slova: *zhluková analýza, dynamické dáta, vizualizácia*

Title: *Cluster analysis of dynamic data*
Author: *Bc. Michal Marko*
Department: *Department of Software and Computer Science Education*
Supervisor: *RNDr. František Mráz, CSc.*
Supervisor's e-mail address: *Frantisek.Mraz@mff.cuni.cz*

Abstract: The mail goal of this thesis is to choose or eventually to propose own modifications to some of the cluster analysis methods in order to observe the progress of dynamic data and its clusters. The chosen ones are applied to the real data. The dynamic data denotes series of information that is created periodically over the time describing the same characteristics of the given set of data objects. When applied to such data, the problem of classic clustering algorithm is the lack of coherence between the results of particular data set from the series which can be illustrated via application to our artificial data. We discuss the idea of proposed modifications and compare the progress of the methods based on them. In order to be able to use our modified methods on the real data, we examine their applicability to the multidimensional artificial data. Due to the complications caused by multidimensional space we develop our own validation criterion. Once the methods are approved for use in such space, we apply our modified methods on the real data, followed by the visualization and evaluation of the results.

Keywords: *cluster analysis, dynamic data, visualization*

1. Úvod

Témou tejto práce je zhluková analýza dynamických dát. Zhluková analýza tvorí jeden veľký celok z oblasti dátového dobývania, alebo taktiež nazývaného dobývanie znalostí. Tento pojem sa dostal do širšieho povedomia v polovici 90tych rokov, keď si nárast objemu dát uchovávaných v rôznych organizáciách spolu s potrebou tieto dáta analyzovať vynútil prepojenie poznatkov z troch oblastí, a to: štatistika, databáze a strojové učenie. Termín dátové dobývanie sa dá definovať ako proces netriviálnej extrakcie implicitných, spočiatku neznámych a potenciálne užitočných informácií z dát [1].

Zhluková analýza je súbor metód, ktoré v empirických údajoch umožňujú hľadať zoskupenia podobných objektov. Takýmto zoskupeniam sa hovorí zhluky. Dôvodom aplikácie metód zhlukovej analýzy môže byť jednak snaha o odhalenie skrytej štruktúry dát, alebo je súčasťou riešenia inej úlohy. Tak, či onak, zhluková analýza hraje dôležitú úlohu v rôznych odvetviach: psychológia a iné spoločenské vedy, biológia, štatistika, rozpoznávanie vzorov, extrahovanie informácií, strojové učenie a iné. Existuje mnoho aplikácií zhlukovej analýzy na praktické problémy.

V tejto práci ju aplikujeme na nový problém. Dynamické dáta sú dáta tvorené sériou, v čase po sebe idúcich dátových množín, popisujúcich rovnaké vlastnosti danej množiny objektov. Požadovaným výstupom nasadenia metódy zhlukovacieho algoritmu na takéto dynamické dáta je zhlukovanie a jeho vizualizácia, v ktorej je možné sledovať vývoj nájdených zhlukov v čase. Pri aplikácii klasických metód zhlukovej analýzy zistíme veľmi jednoducho, že neposkytujú uspokojivé výsledky, pretože zhlukovací algoritmus ako taký, funguje na každej množine dát zo série oddelene. Postupne, bez akejkoľvek heuristiky, hľadá nové zhlukovanie na každej dátovej množine, čo vedie k úplnej absencii istej známky súvislosti vo výsledkoch medzi týmito previazanými dátovými množinami. Tejto problematike sa venujem v kapitole 4.3.

Cieľom tejto práce je navrhnúť metódy, ktoré umožnia sledovať na takýchto dynamických dátach vývoj zhlukov v čase. Navrhnutý postup preskúmam a ilustrujem na sérii umelých dvojrozmerných dynamických dát v 5.sekcii tejto práce. Dôležitým krokom k posúdeniu, či navrhnuté postupy sú schopné podávať prijateľné výsledky aj na reálnych dátach, je prieskum ich priebehu na viacrozmerných umelých dátach. Práve tejto problematike je venovaná kapitola 5.4.

Poslednou fázou výskumu a taktiež predmetom 6. kapitoly je aplikácia modifikovaných postupov na reálne dáta a interpretácia výsledku. Tie majú oproti umelým dátam mnoho nevýhod. Keďže môže ísť o dáta z úplne rôznych oblastí, tak sa v nich užívateľ nemusí tak dokonalo vyznať, ako v prípade umelých dát, ktoré si užívateľ vytvorí sám. Logicky, v reálnom prostredí môže dôjsť k situáciám, ktoré sa v príprave umelých dát ťažko predvídajú. Ďalším podstatným rozdielom je fakt, že reálne dáta môžu, ale aj nemusia medzi sebou istým spôsobom závisieť, ale hlavne, ani skúsený užívateľ nemusí vedieť, v akom stave sú dáta, na ktorých bude prebiehať zhluková analýza.

V neposlednom rade je nutné spomenúť, že viacrozmerné dáta predstavujú veľký problém aj čo sa týka vizualizácie, keďže bežný človek stráca predstavivosť už pri štvrtej alebo piatej dimenzii.

1.1 Členenie práce

V tejto časti stručne predstavím jednotlivé kapitoly. Na začiatku každej hlavnej časti v texte sa čitateľ dozvie, čo sa v danej sekcii textu práce nachádza. V závere každej takejto časti sa nachádza malé zhrnutie, ktoré sumarizuje obsah danej kapitoly.

Úvod, teda prvá kapitola definuje cieľ práce, ktorý je navrhnúť metódy zhlukovej analýzy, umožňujúce sledovať vývoj zhlukov v čase pri aplikácii na dynamické dáta.

V druhej kapitole 2 – **Zhluková analýza** sa bližšie venujem pojmu zhlukovej analýzy. Formálne definujeme základné pojmy a termíny, ktoré budeme v práci používať. Rozoberieme v nej rôzne typy atribútov, pri ktorých popíšeme najznámejšie metódy posudzovania podobnosti, resp. vzdialenosti dvoch dátových objektov.

V nasledujúcej kapitole 3 – **Metódy zhlukovej analýzy** klasifikujem a detailne rozoberiem najznámejšie a najdôležitejšie zhlukovacie algoritmy. Keďže väčšina týchto metód vyžaduje na vstupe informáciu o počte zhlukov, druhú časť tejto kapitoly venujem štúdiu validačných ukazovateľov, ktoré slúžia práve na určenie optimálneho počtu zhlukov.

V štvrtej časti textu , **Porovnanie metód zhlukovania na dynamických dátach,** uvediem zdroje, ktoré slúžili ako základ mojich experimentov. Preskúmam a ilustrujem priebeh vybraných metód zhlukovania a validačných ukazovateľov na umelých dynamických dátach, ktorých štruktúru taktiež nevynechám.

V nadväzujúcej kapitole 5 – **Modifikované metódy** navrhujem postupy, ktoré poskytujú z hľadiska cieľa tejto práce prijateľné výsledky. Popíšem princíp úprav, ktoré sú základom týchto modifikácií. Následne ilustrujem ich priebeh na rôznych modelových situáciách v rámci umelých dvojrozmerných dát. V druhej časti sa pokúsim klasifikovať ich priebeh na viac, než dvojrozmerných dátach.

Šiesta kapitola, **Modifikácie a reálne dáta,** je venovaná výskumu priebehu nami navrhnutých metód na reálnych dátach. Uvedieme zdroj a význam týchto dát, a taktiež spomeniem, ako bolo nutné tieto dáta predspracovať. Pred samotnou analýzou priebehu metód určíme optimálny počet zhlukov a poskytneme riešenie vizualizácie zhlukovania viacrozmerných dát. Následne ilustrujeme výsledky, ktoré sme dosiahli použitím na reálne dáta.

V siedmej kapitole práce zhodnotíme výsledky tejto práce, a taktiež načrtnem smery, ktorými by sa výskum v rámci tejto problematiky mohol vydať.

Na konci práce sa nachádza zoznam použitej literatúry zoradenej podľa výskytu, ako aj obsah priloženého CD.

2. Zhluková analýza

V tejto kapitole bližšie načrtnem problematiku dátového dobývania a hlavne zhlukovej analýzy. Definujem základné pojmy ako aj nároky, ktoré sú kladené na metódy klasteringu. Neskôr sa venujem detailnému popisu dátových typov a možných prístupov k posudzovaniu podobnosti dvoch objektov v rámci dátových typov.

Konkrétne, v nasledujúcej časti, teda 2.1, zavádzam základné pojmy, ktoré budem používať v rámci celej práce. V sekcii 2.2 sa detailne venujem problematike zhlukovej analýzy z hľadiska nárokov užívateľov a aplikácií. Taktiež v nej definujem niektoré dôležité pojmy. Kapitola 2.3 je úvodom do rozdelenia dátových typov. V podkapitole 2.3.1 opisujem intervalové atribúty spolu s definíciami a popisom najznámejších metrík, ktoré slúžia na posudzovanie vzdialeností dvoch dátových objektov. V časti 2.3.2 analogicky rozoberiem binárny typ atribútov. V závere v sekcii 2.3.3 popíšem kategorické, ordinálne a pomerové premenné.

2.1 Základné pojmy

Dnešný život si bez počítača už ťažko dokážeme predstaviť. Počítačové systémy sú takmer všade. S takýmto vysokým stupňom informatizácie a technologickej vyspelosti rastú mimo iného aj nároky na priestor – dáta. Dáta môžu pre niekoho znamenať informácie ako také, informácie spracovávané počítačmi, alebo jednoducho rôzne údaje, čísla, tabuľky, grafy a iné.

Existuje mnoho známych metód na spracovávanie dát, odborne nazývané ako metódy dátového dobývania, no všetky majú spoločnú jednu vec, a to cieľ – extrahovať užitočné informácie z veľkého množstva dát. V tejto práci sa budeme zaujímať výhradne jednou skupinou týchto metód, resp. jej časťou – zhluková analýza alebo tiež klastrovanie. Formálnejším definíciám týchto pojmov sa venujem v ďalšej podkapitole.

S dostupnosťou internetu je možné sa dopátrať k množstvu zaujímavých dát z rôznych oblastí, či už štatistické údaje, zdravotnícke, dáta z rôznych výskumov, ekonomické, demografické a mnoho ďalších od výmyslu sveta. V dátach nebýva nezvyčajné, že obsahujú aj historické údaje, ktoré boli vytvárané s určitou periódou (deň, týždeň, mesiac, kvartál, rok) z dôvodu zaznamenania vývoja. Inými slovami sa dá povedať, že takéto dáta obsahujú časovo oddelené, ale pritom dynamicky meniace sa informácie. A práve tu vzniká nový problém – ako vhodne extrahovať informácie z dynamicky meniacich sa dát. V tejto práci rozoberám niektoré dostupné metódy zhlukovej analýzy, prípadne ich modifikácie a ilustrujem ich priebeh na umelo vytvorených dynamických dátach. Metódy, ktoré sa ukážu ako vhodné, následne aplikujem na reálne dáta.

Pre naše účely, pod pojmom dáta budeme myslieť súbor hodnôt atribútov rôznych entít. Pod entitou, občas taktiež objektom, budeme myslieť jeden záznam v súbore (v databáze). Atribúty, resp. dimenzie sú vlastnosti, ktoré charakterizujú objekt, entitu a môžu nadobúdať rôzne hodnoty z nejakej množiny. Takúto množinu nazývame doména atribútu. Zoskupenie objektov podľa určitej logiky budeme nazývať zhlukom, klastrom alebo taktiež triedou.

2.2 Clustering

V nasledujúcom texte definujem pojem klastering a vyzdvihnem nároky aplikácií na vlastnosti metód zhlukovej analýzy [2].

Predstavte si, že dostanete obrovskú množinu dát a údaj o príslušnosti k nejakej triede nie je známy. Vo veľkých databázach to je celkom bežné, pretože určiť triedu takému množstvu entít je veľmi drahý proces. Klastering je proces zhlukovania dát do tried – klastrov tak, že objekty v jednom klastri si sú navzájom veľmi podobné, ale zároveň sú veľmi rôzne od objektov z iných klastrov. Podobnosť, resp. rôznosť je stanovená na základe hodnôt atribútov popisujúcich objekt, entitu. Spôsob vyhodnocovanie tejto podobnosti, alebo rôznosti je možné ovplyvniť voľbou rôznych postupov, spravidla sa jedná o funkciu. Klastering má korene v mnohých oblastiach, zahŕňajúc dátové dobývanie, štatistika, biológia a strojové učenie.

Klastering je v niektorých aplikáciách taktiež nazývaný dátová segmentácia, pretože rozdeľuje veľké množiny dát do skupín podľa ich podobnosti. Vhodnejší a zároveň presnejší je už spomínaný pojem zhluková analýza. Taktiež sa môže použiť na detekciu krajných hodnôt, alebo skôr hodnôt, ktoré sú ďaleko od ktoréhokoľvek klastru. To môže byť zaujímavejšie, než sa zdá, patrí sem napríklad detekcia podvodov s kreditnými kartami, alebo monitorovanie kriminálnych aktivít v elektronickom obchode. Sledovanie rozmiestnenia dát, pozorovanie vlastností každého klastru, alebo zameranie sa na určitú skupinu klastrov pre hlbšiu analýzu; aj to všetko patrí medzi využitie tohto procesu. Alternatívne, môže poslúžiť ako predprocesný krok pre iné algoritmy.

Na rozdiel od klasifikácie, klastering a metódy učenia bez učiteľa sa nespoliehajú na preddefinované triedy a tréningové príklady. Z tohto dôvodu je zhluková analýza skôr formou metódy učenia pozorovaním, než učenia na príkladoch. Klastering je nepochybne miesto výskumu, v ktorom naň potenciálne aplikácie kladú špecifické požiadavky. Typické sú:

- **Škálovateľnosť:** mnoho algoritmov na malých dátach pracuje uspokojivo; napriek tomu, veľké databázy môžu obsahovať milióny objektov. Zhluková analýza na vzorke z veľkých dát môže viesť k skresleným výsledkom, práve preto sú potrebné rozšíriteľné algoritmy.
- **Schopnosť pracovať s rôznymi typmi atribútov:** mnohé algoritmy sú navrhnuté na kategorizáciu intervalových (numerických) dát. Niektoré aplikácie, avšak, vyžadujú delenie iných typov dát, ako napríklad binárne, kategorické, výčtové, alebo zmes viacerých typov.
- **Objavenie klastrov s ľubovoľným tvarom:** Veľa zhlukovacích algoritmov je založených na Euklidovskej alebo Manhattanskej metrike. Tie zvyknú hľadať guľovité klastre s rovnakou veľkosťou a hustotou. Naproti tomu, zhluk môže byť akéhokoľvek tvaru. Je dôležité vyvinúť algoritmy, ktoré dokážu zistiť klastre ktoréhokoľvek tvaru.
- **Minimálne požiadavky na znalosť domény pre určenie vstupných parametrov:** mnoho algoritmov vyžaduje od užívateľov zadať na vstupe parametre (napríklad počet požadovaných klastrov). Ak sa nad tým človek hlbšie zamyslí, ľahko príde na fakt, že to môže veľmi ľahko ovplyvniť celý výsledok, a to nie len pozitívne. Navyše sa tieto vstupné parametre väčšinou pomerné ťažko určujú, obzvlášť pri viac-dimenzionálnych objektoch. Nezaťažuje to len užívateľa, ale i kontrolu kvality výsledku klastrovania.
- **Schopnosť pracovať s nekvalitnými dátami:** väčšina skutočných databáz v prevádzke obsahuje hraničné alebo chýbajúce hodnoty, neznáme alebo chybné záznamy. Niektoré algoritmy sú citlivé na takéto dáta a môže to viesť k nízkej kvalite klastrov.
- **Inkrementálne zhlukovanie a necitlivosť na poradie vstupných záznamov:** niektoré klastrovacie algoritmy nedokážu spracovať novo vložené dáta (databázové updaty) do existujúcich štruktúr a namiesto toho musia vytvoriť nanovo celé klastrovanie. Niektoré algoritmy sú zas citlivé voči poradiu vstupných dát. To znamená, pri danom vstupe takýto algoritmus môže vrátiť dramaticky rozdielne výsledky v závislosti na poradí vstupných objektov.
- **Mnohorozmernosť:** databázy alebo dátové sklady môžu obsahovať niekoľko dimenzií, či atribútov. Mnoho klastrovacích algoritmov je dobrých pri manipulácii nad dátami s malým počtom dimenzií, napríklad dve, alebo tri. Hľadať zhluky v dátach pri vyššom počte dimenzií je náročné, obzvlášť ak tieto dáta môžu byť riedke, vysoko nevyvážené.
- **Klastering na obmedzenom základe:** aplikácie v reálnom živote môžu vyžadovať vykonanie klasteringu pod rôznymi obmedzeniami. Predstavte si, že Vašou úlohou je vybrať si lokalitu v meste pre umiestnenie určitého počtu nových bankomatov. Pre takéto rozhodnutie môžete zhlukovať domácnosti berúc v úvahu obmedzenia ako vysokorýchlostné komunikácie, typ a počet zákazníkov na klaster. Problém je nájsť skupiny dát s dobrým správaním, ktoré spĺňajú špecifikované obmedzenia.

- **Interpretácia a použiteľnosť:** užívatelia očakávajú, že výsledky klastrovaní sú interpretovateľné, zrozumiteľné, pochopiteľné a použiteľné. Klastering by mal byť zviazaný s určitou sémantikou interpretácií a aplikácií. Je dôležité študovať, ako môže zmysel aplikácie ovplyvniť výber vlastností a metód zhlukovej analýzy.

Skôr než sa dostaneme k rozdeleniu klastrovacích metód a popisu niektorých z nich, je dobré si podrobnejšie rozobrať problematiku dátových typov (typy atribútov), ako ich prípadne predspracovávať pred samotnou analýzou a aké sú možnosti v oblasti určovania podobnosti, či odlišnosti dvoch entít. O problematike dátových typov sa dá dočítať v rôznej literatúre, napríklad [10] alebo [9]. Pre moje potreby som ale použil [2].

Pre potreby nasledujúceho textu si formálnejšie zadefinujeme niektoré premenné. Ako som už spomínal, entitou myslíme jeden objekt, resp. záznam z dátového súboru. V tejto časti textu budeme pod pojmom dátový súbor myslieť množinu X, $X = \{\vec{x}_1, \dots, \vec{x}_N\}$. Entitou, prvkom alebo taktiež objektom myslíme vektor $\vec{x}_i$: $\vec{x}_i = (x_{i1}, \dots, x_{if}), \forall i \in 1 \dots N$, kde f je počet atribútov. Inými slovami je to vektor hodnôt atribútov objektu. Doménu atribútu j, ktorú som už tiež spomínal, budeme označovať D_j a obsahuje hodnoty konkrétneho atribútu, presnejšie $x_{1j}, x_{2j} \dots, x_{Nj} \in D_j, \forall j \in 1 \dots f$. Je nutné poznamenať, že množinu všetkých hodnôt atribútu (doménou) nevieme bližšie definovať, pretože v danej chvíli o jej type nič nevieme. Táto množina môže byť konečne veľká, spočetne nekonečne veľká alebo aj dokonca nespočetne nekonečne veľká. Tieto definované premenné budú v nasledujúcej časti textu prestavovať vstupy. Na základe týchto vstupov rôznymi prístupmi ukážem, ako sa počíta vzdialenosť, resp. odlišnosť medzi dvoma prvkami, formálnejšie $d(\vec{x}, \vec{y}), \vec{x}, \vec{y} \in X$. Analogická je podobnosť, $s(\vec{x}, \vec{y}), \vec{x}, \vec{y} \in X$.

2.3 Typy atribútov

V tejto časti textu vysvetlím, s akými najčastejšími typmi atribútov sa môžeme pri zhlukovej analýze stretnúť, najmä binárne a intervalové. Podrobnejšie sa pozriem na problematiku rôznych prístupov k posudzovaniu podobnosti entít – metriky, korelačné koeficienty a koeficienty podobnosti, resp. odlišnosti. Pri písaní tejto kapitoly som sa inšpiroval najmä [2] a [9].

2.3.1 Intervalové atribúty

Typické intervalové hodnoty sú napríklad váha a výška, teplota a dátum. Majú výhodu, že sú usporiadané, dajú sa medzi nimi počítať rozdiely a určovať vzdialenosti. Dôležité je uvedomiť si, že zmena jednotky atribútu (kilogram → libra) môže ovplyvniť celú analýzu. Vo všeobecnosti, vyjadrenie hodnoty v menších jednotkách vedie k väčšiemu rozsahu pre daný atribút, a tým pádom väčší dopad na klastering. Preto je vhodné dáta nejakým spôsobom normalizovať – snaha, aby všetky jednotky mali rovnakú váhu.

Je to užitočné najmä, keď nemáme žiadnu bližšiu vedomosť o dátach. Naopak, niekedy potrebujeme priradiť niektorým z atribútov vyššiu váhu oproti ostaným. Napríklad pri klastrovaní kandidátov na basketbal by sme pravdepodobne chceli uprednostňovať výšku. Jedna z možností pre normalizáciu – pre dané hodnoty atribútu f sa to dá urobiť nasledovne.

1. Vypočítať priemernú absolútna odchýlku, s_f:

$$s_f = \frac{1}{n}\left(\left|x_{1_f} - \overline{m_f}\right| + \left|x_{2_f} - \overline{m_f}\right| + \ldots + \left|x_{n_f} - \overline{m_f}\right|\right),$$

   kde $x_{1_f}, \ldots, x_{n_f}$ je n hodnôt atribútu f, a $\overline{m_f}$ je stredná hodnota atribútu f, čiže

$$\overline{m_f} = \frac{1}{n}\left(x_{1_f} + x_{2_f} + \ldots + x_{n_f}\right).$$

2. Vypočítať normalizované hodnoty, alebo tzv. z-skóre:

$$z_{if} = \frac{x_{if} - \overline{m_f}}{s_f}$$

   Priemerná absolútna odchýlka, s_f, je viac odolná voči predsunutým (hraničným, extrémnym) hodnotám než smerodajná odchýlka. Pri výpočte priemernej absolútnej odchýlky, odchýlky od priemeru (napr.: $\left|x_{1_f} - \overline{m_f}\right|$) nie sú umocnené a z toho dôvodu je vo výsledku redukovaný dopad týchto predsunutých hodnôt. Existujú robustnejšie metódy merania rozptylu, ako napríklad mediánová absolútna odchýlka, no napriek tomu, výhoda v použití priemernej absolútnej odchýlky spočíva v tom, že z-skóre extrémnych hodnôt nebude príliš malé, a preto zostanú detekovateľné.

Normalizácia môže, ale nemusí byť v niektorých prípadoch užitočná, preto by mala voľba metódy, a či ju vôbec vykonať, zostať na užívateľovi. Ako som už spomínal, metód existuje celé množstvo a zaslúžili by si aj samostatnú kapitolu „Normalizačné techniky“, ale z dôvodu, že to nie je cieľom tejto práce, to nebudem ďalej rozvádzať.

S normalizáciu, alebo bez nej, dôležitú úlohu v klastrovaní hraje podobnosť objektov, keďže klastrovanie sa, ako som už spomínal, snaží nájsť skupiny objektov, vo vnútri ktorých sú si objekty podobné, ale medzi skupinami si nie sú podobné. Väčšinou sa stanoví nejaká funkcia, ktorá určí, ako sú si dva objekty navzájom podobné. Na tento fakt sa dá pozerať z dvoch strán.

Prvý sa dá opísať ako skutočné posudzovanie podobnosti, teda funkcia, ktorá vracia číslo vyjadrujúce, v akej miere sa porovnávané objekty podobajú. Iste by malo byť stanovené nejaké maximum, ktoré by hovorilo, kedy sú objekty totožné (lepšie povedané ich atribúty).

Druhá možnosť je pracovať tak trochu opačným spôsobom, s nepodobnosťou, teda odlišnosťou objektov. Logicky vezmeme v úvahu vzdialenosť. Minimálna vzdialenosť, teda nulová, značí identitu objektov (entít), alebo presnejšie ich vlastností (atribútov) a akákoľvek kladná vzdialenosť značí mieru ich odlišnosti. Existuje viacero prístupov určovania podobnosti, no dá sa povedať, že medzi hlavné z nich patria: koeficienty asociácie, korelačné koeficienty a metriky. Práve posledné spomenuté, metriky, sú bežné pri posudzovaní odlišnosti (alebo podobnosti) entít popísaných intervalovými atribútmi.

Skôr, než popíšem známe typy metrík, formálne definujem pojem metrika.

Funkcia (zobrazenie) $d: M \times M \rightarrow R_{0+}$, kde M je množina, nazývame **metrika** a má nasledujúce vlastnosti:[3]

1. $d(i,j) \geq 0$ pre každú dvojicu $i,j \in M$ (axiom nezápornosti),
2. $d(i,j) = 0$ práve vtedy, keď $i = j$ pre každú dvojicu $i,j \in M$ (axiom identity),
3. $d(i,j) = d(j,i)$ pre každú dvojicu $i,j \in M$ (axiom symetrie) a
4. $d(i,j) \leq d(i,h) + d(h,j)$ pre každú trojicu $i,j,h \in M$ (trojuholníková nerovnosť).

Ďalej popíšem niekoľko najčastejšie využívaných metrík na $\mathbb{R}^n$.

Euklidovská metrika

Najznámejšia metrika – Euklidovská, je definovaná ako

$$d(i,j) = \sqrt{\left(x_{i1} - x_{j1}\right)^2 + \left(x_{i2} - x_{j2}\right)^2 + \cdots + \left(x_{in} - x_{jn}\right)^2}$$

kde $i = (x_{i1}, x_{i2}, \cdots, x_{in})$ a $j = \left(x_{j1}, x_{j2}, \cdots, x_{jn}\right)$ sú dva n-dimenzionálne dátové objekty. Ak vezmeme i ako vektor $\vec{i}$ a j ako $\vec{j}$, Euklidovská vzdialenosť medzi nimi je

$$d(\vec{i},\vec{j}) = \sqrt{\sum_{k=1}^{n} (i_k - j_k)^2}$$

Táto metrika je bežná v reálnom svete, meriame ňou vzdialenosť, či už v rovine, alebo v priestore.

Manhattanská metrika

Asi druhou najznámejšou metrikou je Manhattanská, taktiež v niektorých literatúrach známa, ako City-Block metrika, ktorú sa dá opísať ako cesta prejdená človekom z jedného bodu do druhého, ak by nasledoval mriežkovitú cestu. Formálne, manhattanská vzdialenosť dvoch objektov, je súčet rozdielov ich korešpondujúcich komponent, a teda

$$d(i,j) = \left|x_{i1} - x_{j1}\right| + \left|x_{i2} - x_{j2}\right| + \cdots + \left|x_{in} - x_{jn}\right|$$

alebo vektorovo

$$d(\vec{i},\vec{j}) = \sum_{k=1}^{n} |i_k - j_k|$$

Vezmime si nasledujúci príklad:

Nech $x_1 = (1,2)$ a $x_2 = (3,5)$ reprezentujú dva objekty. Euklidovská vzdialenosť medzi nimi je $\sqrt{(2^2 + 3^2)} = 3.61$. Manhattanská vzdialenosť medzi nimi je $2 + 3 = 5$.

Minkowského metrika

Zovšeobecnením Euklidovskej a Manhattanskej metriky vzniká Minkovského metrika. Je definovaná ako

$$d(i,j) = \left(\left|x_{i1} - x_{j1}\right|^p + \left|x_{i2} - x_{j2}\right|^p + \cdots + \left|x_{in} - x_{jn}\right|^p\right)^{1/p},$$

kde je p je kladné celé číslo. Takáto vzdialenosť sa taktiež v niektorej literatúre nazýva L_p norma. Reprezentuje Manhattanskú vzdialenosť, ak $p = 1$ (L_1 norma) a Euklidovskú vzdialenosť, ak $p = 2$ (L_2 norma).

Ak by mal každý atribút priradenú váhu podľa zdanlivej dôležitosti, vážená Euklidovská vzdialenosť by bola (váženie sa dá aplikovať aj na Manhattanskú aj Minkowského vzdialenosť)

$$d(i,j) = \sqrt{w_1\left|x_{i1} - x_{j1}\right|^2 + w_2\left|x_{i2} - x_{j2}\right|^2 + \cdots + w_m\left|x_{in} - x_{jn}\right|^2}$$

kde $w_1, \cdots w_m$ sú váhy jednotlivých atribútov.

Čebyševova metrika

Táto vzdialenosť je najľahšie definovateľná ako minimálny počet krokov, ktorý potrebuje kráľ na šachovnici na pohyb medzi dvoma miestami. Tiež sa jej hovorí šachová vzdialenosť [4].

Matematicky sa to dá zapísať ako

$$d(\vec{i}, \vec{j}) = \max_k(|i_k - j_k|)$$

Pearsonov korelačný koeficient

Vzdialenosť sa taktiež dá odvodiť z korelačných koeficientov. Tieto koeficienty sa používajú na určovanie závislosti dvoch premenných. Jeden z takýchto koeficientov je práve Pearsonov korelačný koeficient (miera lineárnej závislosti dvoch premenných) [5], definovaný ako:

$$r_{ij} = \frac{\sum_{l=1}^{d}(x_{il} - \overline{x_i})\left(x_{jl} - \overline{x_j}\right)}{\sqrt{\sum_{l=1}^{d}(x_{il} - \overline{x_i})^2 \sum_{l=1}^{d}\left(x_{jl} - \overline{x_j}\right)^2}}$$

kde $\overline{x_i} = \frac{1}{d}\sum_{l=1}^{d} x_{il}$. Je vhodné poznamenať, že korelačný koeficient je v rozsahu $\langle -1,1\rangle$, kde 1 značí najsilnejšiu pozitívnu koreláciu a -1 značí najsilnejšiu negatívnu koreláciu. Vzdialenosť nasledovne môžeme definovať ako:

$$d\left(x_i, x_j\right) = \left(1 - r_{ij}\right)/2$$

čo spadá do intervalu $\langle 0,1\rangle$ [9].

Spearmanov korelačný koeficient

Je to neparametrická metóda založená na poradí premenných usporiadaných podľa veľkosti vzhľadom ku dvom sledovaným veličinám. Skúma akúkoľvek monotónnu závislosť a nie len lineárnu, ako Pearsonov korelačný koeficient. Každej premennej sa priradí dvojica poradia Q (poradie podľa prvej veličiny X) a R (poradie podľa druhej veličiny Y). Ak by s rastúcimi hodnotami X vzrastali aj hodnoty Y, bolo by poradie oboch veličín zhodné, t.j. $Q = R$ pre každú premennú. Ak s rastúcimi hodnotami X hodnoty Y klesajú, sú práve poradia oboch veličín opačné. Pri nezávislosti sú poradia poprehadzované úplne náhodne. Pre n pozorovaných dvojíc vo výberu sa Spearmanov korelačný koeficient počíta pomocou diferencií poradí $d_i = Q_i - R_i$ ako

$$r_s = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}$$

Pri zhodnom poradí dosahuje koeficient r_s maximálnu hodnotu 1, pri opačnom poradí minimálnu hodnotu -1. V ostatných prípadoch je $-1 < r_s < 1$. Hodnoty korelačného koeficientu blízke nule naznačujú, že poradia sú náhodne prehádzané a medzi sledovanými veličinami nie je závislosť.

V literatúre z oblasti štatistiky by sme našli mnoho ďalších korelačných koeficientov [6][7], no pri ich použití je dôležité mať na pamäti, že majú tendenciu skôr odhaľovať rozdiel v tvaroch, než zisťovať hodnoty rozdielov medzi dvoma objektmi.

V ďalšej časti rozoberiem problematiku binárnych atribútov.

2.3.2 Binárne atribúty

Binárny atribút nadobúda len dve hodnoty – 0 alebo 1; 0 – premenná nie je prítomná, 1 – premenná je prítomná. Napríklad atribút *študent* u entity *zamestnanec,* 1 značí študenta – zamestnanec na pracovnú dohodu, 0 značí opak – zamestnanec na plný úväzok.

Ak by sme binárne hodnoty brali ako intervalové, mohlo by to viesť k zavádzajúcim výsledkom. Preto sú aj metódy odhadu podobnosti, či odlišnosti iné – špecifické pre binárne údaje.

Pri binárnych doménach posudzujeme, či majú dané objekty v atribúte zhodu. Pri tom musíme posúdiť, akú informáciu nám daná zhoda prináša. Môžu nastať dva druhy zhôd. Pozitívna zhoda (1-1) hovorí, že oba objekty vlastnosť majú, negatívna zhoda (0-0) značí, že oba objekty vlastnosť naopak nemajú. Tieto situácie ale pritom nemusia poskytovať rovnako cennú informáciu.

Situácie, ktoré u binárnych atribútov môžu nastať sú popísané v tabuľke.

		Objekt j		
		1	0	∑
Objekt i	1	*a*	*b*	***a+b***
	0	*c*	*d*	***c+d***
	∑	***a+c***	***b+d***	***t***

Kontingenčná tabuľka - binárne atribúty

V tejto tabuľke:

- a je počet premenných rovnajúcich sa 1 pre oba objekty i a j,
- b je počet premenných rovnajúcich sa 1 pri objekte i a 0 pri objekte j,
- c je počet premenných rovnajúcich sa 0 pri objekte i a 1 pri objekte j,
- d je počet premenných rovnajúcich sa 0 pre oba objekty i a j,
- t je počet všetkých premenných, $t = a + b + c + d$.

Posledným rozdielom pri posudzovaní podobnosti binárnych atribútov je, či sa na pozitívne a negatívne zhody pozeráme symetricky alebo asymetricky. Symetrická premenná je taká, ak obidve jej hodnoty majú ekvivalentnú váhu – napríklad pohlavie. Asymetrické premenné majú rozdielne váhy hodnôt – vyjadrenie výsledku testu na nejakú chorobu. Nesmieme ale zabúdať, že pri voľbe ktorejkoľvek metódy z nich sa môžeme navyše rozhodnúť, či budeme počítať podobnosť ($s(\vec{i},\vec{j})$) alebo odlišnosť ($d(\vec{i},\vec{j})$).

Niektoré koeficienty výpočtu podobnosti dvoch binárnych entít popisuje nasledujúca tabuľka.

Meno koeficientu	$s(\vec{i},\vec{j})$
Jaccard	$\frac{a}{a+b+c}$
Simple Matching	$\frac{a+d}{a+b+c+d}$
Yule	$\frac{ad-bc}{ad+bc}$
Hamann	$\frac{(a+d)-(b+c)}{(a+d)+(b+c)}$
Sorenson / Dice	$\frac{2a}{2a+b+c}$
Rogers and Tanimoto	$\frac{a+d}{a+2(b+c)+d}$
Sokal and Sneath	$\frac{2(a+d)}{2(a+d)+b+c}$
Russel and Rao	$\frac{a}{a+b+c+d}$

Koeficienty podobnosti binárnych entít [8]

Záver tejto časti textu venujem krátkemu popisu kategorických, ordinálnych a pomerových premenných.

2.3.3 Kategorické, ordinálne a pomerové premenné

Kategorická (nominálna) premenná je zovšeobecnenie binárnej premennej, ktorá môže obsahovať viac, než dva stavy. Napríklad, *map_color* je kategorická premenná, ktorá môže mať, povedzme päť stavov: červená, žltá, zelená, ružová a modrá.

Ordinálna premenná pripomína kategorickú, s tým rozdielom, že stavy ordinálneho atribútu sú v istom zmysle zoradené. Tie sa dajú rozlíšiť na dva typy: diskrétne a spojité. Príkladom diskrétneho ordinálneho atribútu môže byť záznam subjektívneho hodnotenia kvalít, ktoré sa nedajú zmerať objektívne; hodnotenie zamestnanca v rámci firemnej pyramídy - asistent, konzultant, manažér a partner.

Spojitá ordinálna premenná vyzerá ako množina spojitých hodnôt neznámej mierky; je dôležité relatívne poradie hodnôt, ale ich skutočná hodnota nie. Napríklad, relatívne poradie v športe (zlato, striebro, bronz) je dôležitejšie, než ich skutočné hodnoty.

Pomerové atribúty majú takisto dané usporiadanie, ale určovanie podobnosti dvoch entít nemusí posudzovať len ich rozdiel, ale taktiež ich vzájomný pomer. Taktiež je stanovená nula ako najnižšia hodnota (ako napríklad pri Kelvinovej stupnici). Medzi príklady patria: rast populácie baktérií v čase, rozpad rádioaktívneho prvku, alebo viac predstaviteľné – poradia v športe a ich časy, prípadne pulz pacienta (je logické, ak povieme, že pulz pri 120 je dvakrát viac, než pulz pri 60).

ID_entity	kategoricka_premenna	ordinalna_premenna	pomerova_premenna
1	Cervena	Vyborny	523
2	Zlta	Velmi dobry	56
3	Modra	Chvalitebny	248
4	Ruzova	Nedostatocny	0,609

Tabuľka príkladov premenných

V tejto časti textu som pre potreby tejto diplomovej práce zadefinoval základné pojmy z oblasti zhlukovej analýzy. V stručnom úvode som taktiež načrtol vlastnosti, ktoré sú od zhlukovacích algoritmov a aplikácií vyžadované. Popísal som rôzne typy atribútov, ako aj viaceré metriky, či koeficienty určovania vzdialenosti, resp. podobnosti dvoch entít.

V nasledujúcej kapitole sa venujem metódam zhlukovej analýzy, jednak ich deleniu a jednak detailnejšiemu popisu niektorých najobľúbenejších, resp. najznámejších z nich.

3. Metódy zhlukovej analýzy

Predmetom tejto časti práce je klasifikácia metód zhlukovej analýzy. V príslušných podkapitolách sa venujem detailnému popisu princípu a vlastností najznámejších a najobľúbenejších algoritmov z každej kategórie metód. Niektoré metódy v ďalších častiach tejto práce aplikujem na vytvorené umelé dáta a porovnám ich výsledky. Ako zdroj informácií som použil viacero publikácií [2][9][10].

V literatúre existuje mnoho klastrovacích metód, ale je veľmi ťažké ich rozdeliť do kategórií, pretože hranice medzi nimi sú veľmi tenké, ba dokonca môžu presahovať medzi sebou, a to z jediného dôvodu: metóda môže mať vlastnosti z viacerých kategórií. Každopádne, oplatí sa mať nejakú predstavu o tom, ako sa metódy zhlukovej analýzy približne členia.

Rovnako ako u väčšiny iných problémov, rôzne metódy sú vhodné pre rôzne zadania úloh a častokrát sa nedá dopredu povedať, ktorá metóda poskytne lepšie výsledky. Voľba algoritmu závisí na oboch: type dostupných dát a konkrétnom účele aplikácie. Často je vhodné na dáta aplikovať niekoľko rôznych metód.

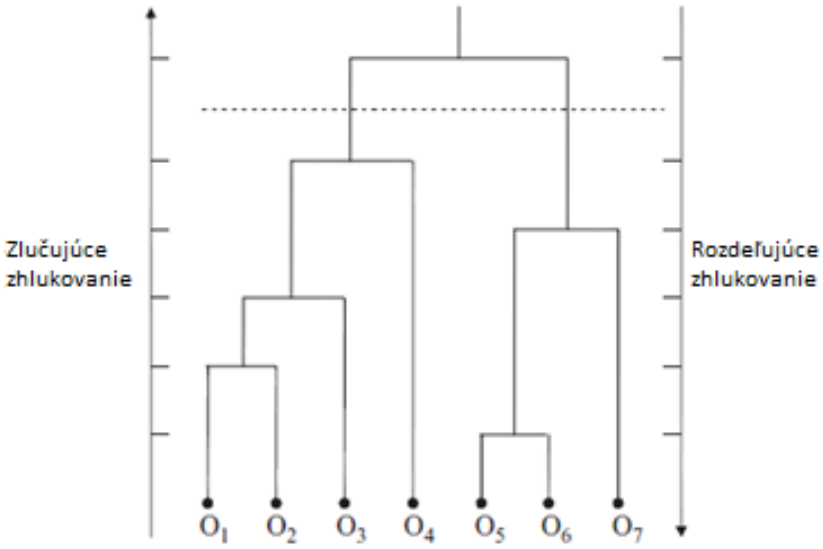
Vo všeobecnosti, klastrovacie techniky sú klasifikované ako hierarchické klastrovanie a segmentujúce (nehierarchické) klastrovanie, na základe vlastností generovaných zhlukov. Techniky segmentujúceho klastrovania priamo rozdeľujú dáta na určitý počet zhlukov bez hierarchickej štruktúry, kým hierarchické metódy zoskupujú dáta postupne do vnorených segmentov, či už zo zhlukov jedincov do zhluku obsahujúceho všetkých alebo naopak. To prvé spomenuté sa volá zlučujúce (hromadenie, aglomerácia) hierarchické zhlukovanie a jeho opakom je rozdeľujúce hierarchické zhlukovanie.

Práve hierarchickými metódami sa budem zaoberať v nasledujúcej podkapitole, teda 3.1. V jej podkapitole 3.1.1 sa pozriem podrobnejšie na zlučujúce hierarchické zhlukovanie spolu s najpoužívanejšími funkciami výpočtu vzdialeností medzi dvoma zhlukmi. V 3.1.2 popíšem rozdeľujúce hierarchické zhlukovanie a jeden z najznámejších optimalizovaných algoritmov z tejto kategórie. V sekcii 3.2 sa zaoberám popisom nehierarchických metód. Podrobne sa venujem popisu metódy *k*Means v 3.2.1, metóde *k*Medoids v časti 3.2.2. V kapitole 3.2.3 rozoberám metódu Fuzzy *c*Means a v sekcii 3.2.4 algoritmus Expectation-Maximization. Zástupcu neurónových sietí – Kohonenove mapy popíšem v časti 3.2.5. Neskôr, v kapitole 3.3 rozoberám problematiku validačných kritérií. V jej podkapitolách postupne rozoberám konkrétne validačné ukazovatele, a teda: časť 3.3.1 je venovaná Dunnovmu indexu, v 3.3.2 je popis Davies–Bouldin indexu, 3.3.3 definuje RMSSTD a RS ukazovateľ, sekcia 3.3.4 Xie–Beni index. Partition ukazovateľ a index separácie je popísaný v kapitole 3.3.5. V časti 3.3.6 preberiem Partition koeficient a na záver v 3.3.7 sa pozriem na koeficient entrópie. Zhrnutie celej tretej kapitoly sa nachádza v sekcii 3.4.

3.1 Hierarchické metódy

Či už zlučovacie alebo rozdeľujúce hierarchické metódy, oba prístupy organizujú dáta do hierarchických štruktúr na základe príbuznosti. Výsledok hierarchických algoritmov je zvyčajne zobrazený pomocou binárneho stromu alebo dendogramu, viď obr.1.

Koreň dendogramu reprezentuje celý dátový súbor a každý list je jeden objekt dát. Uzly medzi nimi opisujú rozsah, v ktorom sú si objekty príbuzné a výška dendogramu vyjadruje vzdialenosť každého páru objektov, či klastrov alebo objektu a klastru. Konečné výsledky zhlukovej analýzy sú získavané prierezom dendogramu na rôznych úrovniach (prerušovaná čiara na obrázku obr. 1).



Obr. 1: Príklad dendogramu z hierarchického klastrovania. Na základe prierezu dendogramu prerušovanou čiarou v danej úrovni získavame dva zhluky.

Z dôvodu vysokej výpočtovej náročnosti pri porovnávaní podmnožín klastrov pri použití rozdeľujúcich metód sú používanejšie práve zlučujúce hierarchické metódy. Každopádne, hierarchické metódy ako také sú kritizované kvôli ich výpočtovej náročnosti, ktorá je aspoň $O(N^2)$, čo je veľmi nevhodné pri použití na obrovských dátach [9].

3.1.1 Zlučujúce hierarchické zhlukovanie

Pre predstavu, ako fungujú zlučujúce zhlukovacie metódy si uvedieme jednoduchý príklad, algoritmus, ktorý využívajú všetky metódy. Konkrétne metódy sa medzi sebou líšia v prístupe, ako určujú vzdialenosti medzi zhlukmi. Rôzne prístupy k posudzovaniu vzdialeností prinášajú skutočne zásadné rozdiely vo výsledkoch zhlukovania.

Vstup algoritmu je N objektov, každý z nich predstavuje práve jeden zhluk. Nasleduje postupnosť operácií zlievania dvojíc najbližších klastrov, až kým sa všetky dátové objekty neocitnú v jednom zhluku. Postup sa dá znázorniť nasledovne:

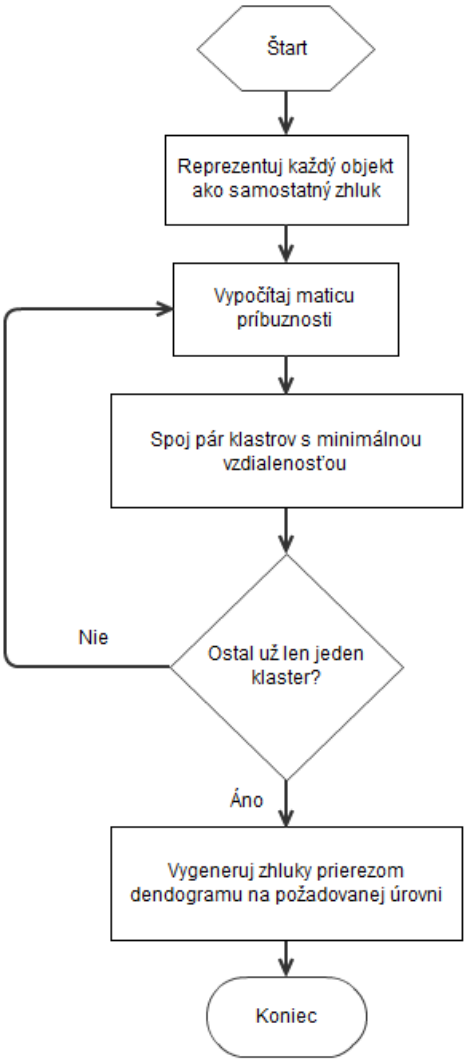


Diagram aglomeračného hierarchického algoritmu. V každom kroku sú spájané dva zhluky, až kým nie sú všetky spojené do jedného veľkého zhluku.

Obr. 2

Očividne, zlievanie dvoch zhlukov alebo formovanie nového zhluku je závislé na definícii funkcie pre výpočet vzdialenosti dvoch zhlukov, a preto v ďalšej časti textu niekoľko z nich bližšie popíšem [10].

Jednoduché spájanie (metóda najbližšieho suseda)

Najstaršia a najjednoduchšia aglomeračná metóda [11][12]. Je definovaná ako:

$$d(R,Q) = \min_{\substack{i \in R \\ j \in Q}} d(i,j)$$

kde R a Q sú zhluky, ktorých vzdialenosť sa snažíme posúdiť. Z definície plynie jej hlavná slabosť a nevýhoda. Ak dva zhluky k sebe priblížia, čo by len v jednom bode, aj keď sú veľké a zreteľne rozdielne, táto metóda ich nebude schopná držať od seba. To vedie k takzvanému „reťazovému efektu“, kde sú nekvalitne separované zhluky zreťazené. Výsledné klastre potom nevyzerajú klasicky, kružnicovo, ale pretiahnuto – reťazec.

Úplné spájanie (metóda najvzdialenejšieho suseda)

Úplným opakom vyššie uvedenej metódy je metóda najvzdialenejšieho suseda [13][14][15]. Rozdielnosť dvoch zhlukov je definovaná ako najväčšia odlišnosť objektu z jedného zhluku a objektu z druhého zhluku. Presnejšie:

$$d(R,Q) = \max_{\substack{i \in R \\ j \in Q}} d(i,j)$$

Zatiaľ čo metóda najbližšieho suseda smeruje k príliš malému počtu zhlukov, ktoré sú navyše natiahnuté, úplné spájanie často vedie k opačnému efektu – mnoho zhlukov, v rámci ktorých sú až príliš malé rozdiely, čo sa javí ako príliš rovnomerné. Výsledok je, že relatívne podobné objekty zostanú v rozdielnych zhlukoch dlhý čas.

Centroidná metóda

Centroidná metóda je založená na myšlienke určovania vzdialenosti dvoch zhlukov na základe vzdialenosti ich centroidov – ťažísk. Ťažisko klastru je vektor, ktorého hodnota v každej zložke – atribúte je priemerná hodnota tohto atribútu vzhľadom k všetkým prvkom v danom zhluku, formálne:

$$\bar{x}(R) = (\overline{x_1}(R), \overline{x_2}(R), \cdots, \overline{x_n}(R))$$

kde R je zhluk a n je počet atribútov (dimenzií). f-tá súradnica je:

$$\overline{x_f}(R) = \frac{1}{|R|}\sum_{i \in R} x_{if}$$

pre $f = 1, \cdots, n$. Čiže, ak máme dva zhluky R, Q, ich vzdialenosť je rovná druhej mocnine ich Euklidovskej vzdialenosti, čiže:

$$d(R, Q) = \|\bar{x}(R) - \bar{x}(Q)\|^2$$

Wardova metóda

Princípom Wardovej metódy nie je optimalizácia vzdialeností medzi hlukmi, ale minimalizácia heterogenity zhlukov podľa kritéria minima prírastku vnútroskupinového súčtu štvorcov odchýlok objektov od centroidu zhluku. V každom kroku sa pre všetky dvojice odchýlok spočíta prírastok súčtu štvorcov odchýlok vzniknutých ich zlúčením. Potom sa spoja tie zhluky, ktorým odpovedá minimálna hodnota tohto prírastku. Formálne je prírastok definovaný nasledovne

$$\Delta(R, Q) = \sum_{i \in R \cup Q} \left\|\vec{x}_i - \vec{m}_{R \cup Q}\right\|^2 - \sum_{i \in R} \|\vec{x}_i - \vec{m}_R\|^2 - \sum_{i \in Q} \left\|\vec{x}_i - \vec{m}_Q\right\|^2 = \frac{n_R n_Q}{n_R + n_Q} \left\|\vec{m}_R - \vec{m}_Q\right\|^2,$$

kde $\vec{m}_j$ je stred zhluku j, n_j je počet bodov v ňom. Δ je cena zlievania zhlukov R a Q.

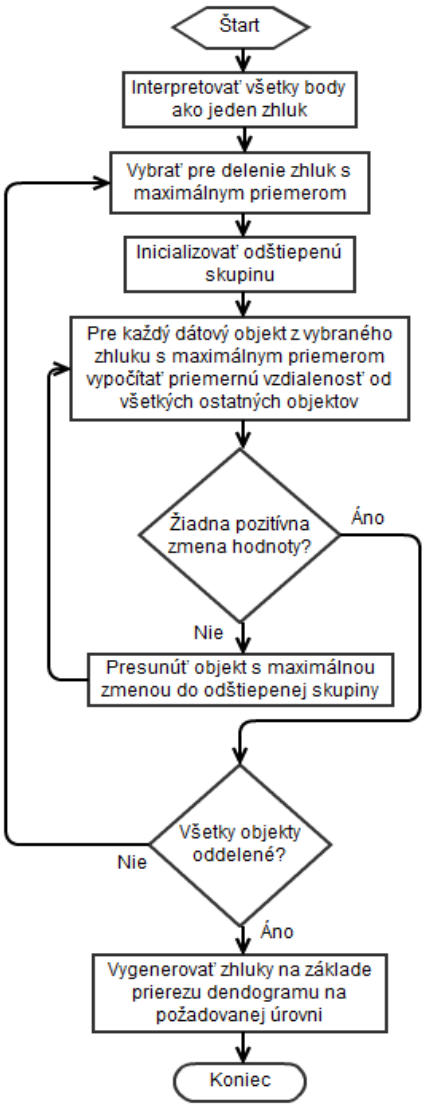
Prírastok sme teda vyjadrili ako súčet štvorcov odchýlok od stredu v oboch zanikajúcich zhlukoch. Úpravami sme ho zjednodušili a vyjadrili ako súčin euklidovskej vzdialenosti medzi ťažiskami uvažovaných zhlukov pre spojenie a koeficientom závisiacim od veľkosti zhlukov. Hodnota tohto koeficientu rastie s rastúcou veľkosťou zhluku a pre pevné $n_R + n_Q$ je maximálna pri zhlukoch zhodnej veľkosti $n_R = n_Q$. Wardova metóda ma tendenciu odstraňovať malé zhluky, teda tvorí zhluky približne zhodnej veľkosti, čo môže byť príjemná vlastnosť.

Detailný popis uvedených metód, ako aj ďalších, sa dá nájsť v odbornej literatúre, napríklad [5] a [10].

3.1.2 Rozdeľujúce hierarchické zhlukovanie

V porovnaní so zlučujúcimi hierarchickými metódami zhlukovej analýzy, rozdeľujúce algoritmy pri výpočte postupujú opačne. Na začiatku sú všetky objekty v jednom zhluku a procedúra ho krok po kroku rozdeľuje, až kým každý zhluk neobsahuje práve jeden objekt. Pre vstup s N objektmi musí takýto algoritmus len na začiatku zvážiť $2^{N-1}-1$ možných rozdelení objektov do dvoch neprázdnych podmnožín, čo je výpočtovo náročné dokonca aj pri malých dátach. Práve z tohto dôvodu nie sú v praxi rozdeľujúce hierarchické metódy zhlukovania častou voľbou. Napriek tomu podávajú lepší náhľad do hlavnej štruktúry dát, pretože väčšie zhluky sú tvorené v prvotných krokoch celého klastrovacieho procesu, a tým pádom budú s menšou pravdepodobnosťou trpieť nahromadenými chybami pri rozhodovaní, ktoré sa už v neskorších krokoch napraviť nedajú.

Z dôvodu vysokej náročnosti týchto metód vznikli rôzne modifikácie, resp. heuristické metódy. Jednou z nich je napríklad algoritmus DIANA (Divisive ANAlysis), ktorého základy položil [15].



Obr. 3

Podľa [5], v každej fáze, DIANA pozostáva zo série iteratívnych krokov s cieľom presunúť podobné objekty do tzv. odštiepenej skupiny. V každej fáze teda:

1. Nájsť zhluk s maximálnym priemerom.

2. V danom zhluku vybrať objekt, ktorý má najväčšiu mieru odlišnosti vzhľadom k ostatným objektom daného zhluku; tento objekt je reprezentant nového zhluku

3. Odštiepiť objekty pôvodného zhluku, ktoré sú bližšie k novému reprezentantovi a vytvoriť nový zhluk

Diagram priebehu algoritmu DIANA. Na rozdiel od zlučujúcich hierarchických algoritmov, DIANA je rozdeľujúci hierarchický algoritmus, ktorý začína so zhlukom pozostávajúceho zo všetkých dátových objektov a postupne ich rozdelí do zhlukov o veľkosti práve jedného dátového objektu.

Kritika klasických hierarchických algoritmov spočíva v nedostatočnej robustnosti a citlivosti voči šumu (irelevantné alebo nezmyselné dáta) a hraničným hodnotám. Ak je objekt už raz priradený nejakému zhluku, nebude sa znovu posudzovať, čo znamená, že algoritmus nie je schopný opraviť možné zlé rozhodnutie. Ako som už spomínal, ďalšia veľká nevýhoda je vysoká výpočtová náročnosť, ktorá je minimálne $O(N^2)$, čo je nepoužiteľné v databázach, úlohách dátového dobývania a iných úlohách dnešného rozmeru. Tieto nedostatky podnietili výskum nových metód práve v tejto oblasti a vznikli tak nové hierarchické zhlukovacie algoritmy s výrazne lepším výkonom. Sú to napríklad BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [16], CURE (Clustering Using REpresentatives) [17], ROCK (RObust Clusetring using linKs) [18] a Chameleon [19]. Pre lepšiu predstavu uvediem algoritmus BIRCH [2].

BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies

BIRCH je navrhnutý pre zhlukovú analýzu veľkého množstva numerických dát integráciou hierarchického klastrovania (v počiatočných fázach) a iných zhlukovacích metód, ako napríklad iteratívne rozdeľovanie (v neskorších fázach). Táto metóda prekonáva dva problémy: škálovateľnosť a neschopnosť vrátiť zmeny, ktoré boli vykonané v predošlom kroku.

BIRCH prináša dva koncepty, tzv. clustering feature (charakteristika zhluku, CF) a clustering feature tree (CF strom). Tieto štruktúry pomáhajú klastrovacej metóde dosahovať vyššiu rýchlosť a škálovateľnosť vo veľkých databázach, ako aj efektívnosť pri inkrementálnom a dynamickom zhlukovaní objektov.

Nech máme v zhluku n d-rozmerných dátových objektov. Ťažisko $\overline{x_0}$, polomer R a priemer D takého zhluku definujeme nasledovne:

$$\overline{x_0} = \frac{\sum_{i=1}^{n} \vec{x}_i}{n}$$

$$R = \sqrt{\frac{\sum_{i=1}^{n} (\vec{x}_i - \overline{x_0})^2}{n}}$$

$$D = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} (\vec{x}_i - \vec{x}_j)^2}{n(n-1)}}$$

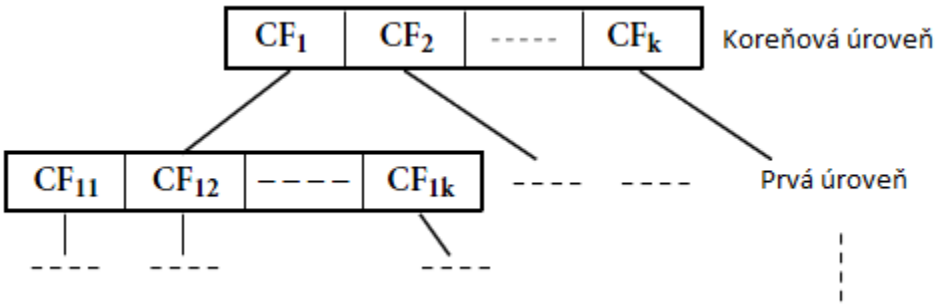
kde R je priemerná vzdialenosť od členských objektov k ťažisku a D je priemerná vzdialenosť dvojíc v rámci zhluku. R a D odrážajú tesnosť zhluku okolo jeho ťažiska. CF je trojrozmerný vektor uchovávajúci informáciu o zhlukoch. Pri daných n d-rozmerných objektoch, $\{\vec{x}_i\}$, v zhlukuje potom CF tohto zhluku definovaný ako

$$CF = < n, LS, SS >,$$

kde n je počet objektov v zhluku, LS je lineárny súčet n objektov zhluku ($\sum_{i=1}^{n} \vec{x}_i$) a SS je štvorcový súčet objektov zhluku ($\sum_{i=1}^{n} \vec{x}_i^{\,2}$).

CF je v zásade súhrn informácií, resp. štatistík pre daný zhluk a je vhodné poznamenať, že CF je aditívna vlastnosť, t.j. predpokladajme, že máme dva disjunktné zhluky, C_1 a C_2, s príslušnými CF_1 a CF_2. Charakteristika zhluku, ktorý vznikne zlúčením C_1 a C_2 je jednoducho $CF_1 + CF_2$. Charakteristiky zhlukov sú výpočty, ktoré sú potrebné pre rozhodovanie počas behu algoritmu BIRCH.

CF strom je výškovo vyvážený strom, ktorý uchováva charakteristiky zhlukov. Podľa definície, uzol (nie listový) má potomkov – deti. Takýto uzol obsahuje súčty charakteristík jeho detí a teda zhŕňa zhlukovacie informácie o ich deťoch. CF strom má dva parametre: vetviaci faktor B a prah T. Vetviaci faktor určuje maximum detí pre uzol. Parameter prah určuje maximálny priemer podzhlukov uložených v listoch CF stromu. Tieto dva parametre ovplyvňujú veľkosť výsledného stromu.



Obr. 4: Obrázok znázorňujúci štruktúru CF stromu

Pri obmedzenom množstve pamäte je dôležité minimalizovať čas potrebný na I/O operácie. BIRCH aplikuje techniku viacfázového zhlukovania: jeden prechod dátovou množinou dáva základný, dobrý výsledok a druhý prechod môže (voliteľne) vylepšiť kvalitu výsledného zhlukovania. Tieto fázy sú:

- Fáza 1: BIRCH prejde databázu, aby v pamäti postavil CF strom.
- Fáza 2: BIRCH aplikuje vybranú metódu zhlukovej analýzy na listy CF stromu.

Vo fáze 1 sa CF strom postaví inkrementálne, tak ako sa vkladajú objekty. Objekt je vložený do najbližšieho listu. Ak je priemer zhluku po vložení objektu do zhluku uloženého v liste väčší než prahový parameter, tento zhluk sa rozdelí (možno aj ďalšie).

Po samotnom vložení nového objektu je informácia predávaná hore smerom ku koreňu stromu. Veľkosť CF stromu je ovplyvniteľná veľkosťou prahovej hodnoty. Ak veľkosť pamäte, ktorá je potrebná pre uloženie CF stromu, je väčšia než dostupná pamäť, hodnota prahu musí byť znížená a CF strom sa prestavia. Proces rekonštrukcie stromu je postavenie nového stromu z listov starého stromu, takže to prebehne bez potreby opätovného načítania všetkých objektov, resp. bodov. Je to podobné vkladaniu a štiepeniu uzlov pri konštrukcii B+- stromov. To je dôvod, prečo na postavenie stromu stačí prečítať dáta len raz. Vzniklo niekoľko heuristík a metód, ktoré riešia problematiku hraničných hodnôt objektov a zlepšujú tak kvalitu CF stromov pomocou ďalších prístupov do dát. V čase, keď je CF strom kompletne postavený, čiže vo fáze 2, BIRCH na tento CF strom aplikuje metódu zhlukovej analýzy, napríklad nejaký nehierarchický algoritmus.

Výpočtová náročnosť BIRCH algoritmu je $O(n)$, kde n je počet objektov pre zhlukovanie, čo je rádovo nižšia náročnosť, než pri algoritme DIANA. Napriek tomu, aj tento algoritmus má svoje nedostatky. Keďže každý uzol kvôli jeho veľkosti v CF strome môže obsahovať len obmedzený počet záznamov, nie vždy je tento uzol schopný odpovedať stavu, ktorý užívateľ považuje za prirodzený zhluk.

Navyše, ak nemajú zhluky guľovitý tvar, BIRCH neposkytne dobrý výsledok, pretože posudzovanie hraníc zhluku funguje na myšlienke polomeru, resp. priemeru.

3.2 Nehierarchické zhlukovacie metódy

V tejto kapitole predstavím najznámejšie a najobľúbenejšie metódy z oblasti nehierarchických, iným slovom segmentujúcich zhlukovacích metód, ako sú napríklad: kMeans, kMedoids, fuzzyCmeans, EM Clustering. Popíšem princíp ich fungovania, a taktiež ich prednosti a slabosti. Tieto metódy vyžadujú na vstupe informáciu o požadovanom počte zhlukov vo výsledku, preto sa v kapitole venujem aj problematike určenia tohto počtu – označovaného k.

Na vstupe týchto metód je pochopiteľne dátová množina pozostávajúca z n objektov a číslo k, ktoré určuje počet výsledných zhlukov. Nehierarchické metódy organizujú objekty z dátovej množiny do k partícií – zhlukov. Zhluky sú sformované na základe kritéria, ako napríklad koeficienty podobnosti založené na metrikách.

Medzi najznámejšie a najčastejšie používané metódy patria kMeans [20] a kMedoids. Práve nimi sa budem zaoberať ako prvými v nasledujúcom texte.

3.2.1 *k*-Means clustering

Metóda *k*-Means [2] hľadá optimálny zhluk z dát na základe minimalizácie kritéria (typicky kritérium súčtu štvorcov chýb) pomocou iteratívneho optimalizačného postupu, čo mimochodom spadá do kategórie tzv. hill-climbing, resp. gradientných algoritmov (hladové algoritmy, v každom kroku sa rozhodne pre lokálne optimum aj za cenu, že sa cez neho nemusí dostať ku globálnemu optimu – ako horolezec, ktorý zdoláva najbližšiu horu).

Na vstupe sa nachádza dátová množina a parameter k. Metóda rozdelí dáta o n objektoch do k zhlukov tak, že výsledná vnútrozhluková podobnosť je vysoká a medzihluková podobnosť naopak, nízka. Zhluková podobnosť je počítaná s ohľadom na priemernú hodnotu objektov v zhluku, čo sa dá nazvať ťažiskom zhluku.

Funguje nasledovným spôsobom. Najprv náhodne vyberie k objektov, každý z nich na začiatku reprezentuje priemer zhluku, ťažisko. Ostatné objekty sú priradené týmto zhlukom, podľa toho, ktorému sú najviac podobné, čo sa posudzuje na základe vzdialenosti objektu a ťažiska zhluku. Potom vypočíta nové ťažisko pre každý zhluk. Tento proces sa opakuje kým funkcia opisujúca kritérium neskonverguje, t.j. výpočet beží do bodu, keď v už zhlukoch nenastávajú žiadne zmeny. Použité kritérium súčtu štvorcov chýb je definované ako:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - \bar{c}_i|^2$$

kde E je súčet štvorcov chýb všetkých objektov z dátovej množiny, p je bod v priestore reprezentujúci príslušný objekt a $\bar{c}_i$ je ťažisko zhluku C_i (p a $\bar{c}_i$ sú viacrozmerné). Inými slovami, pre každý objekt v každom zhluku, vzdialenosť objektu od ťažiska zhluku je umocnená a tieto vzdialenosti sčítané. Kritérium sa snaží vytvoriť zhluky, ktoré sú čo najviac kompaktné a separované.

Algoritmus: *k*-Means. Segmentujúca metóda *k*-Means, kde ťažisko každého zhluku je reprezentované priemerom hodnôt objektov v zhluku.

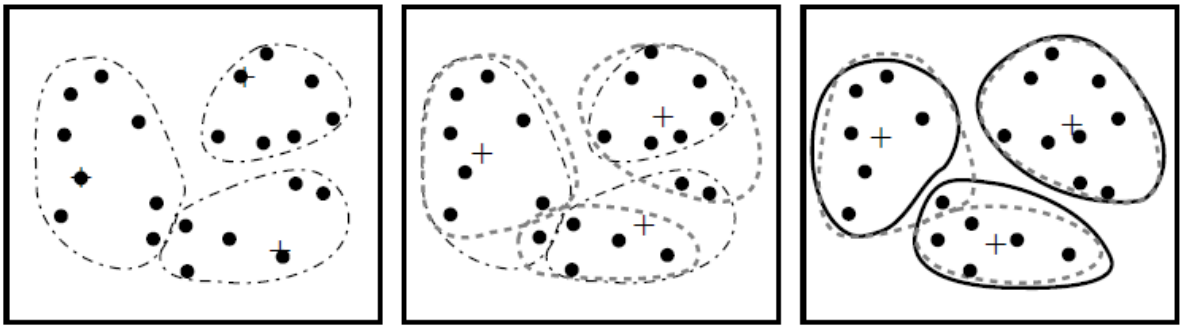
Vstup:

- k: počet zhlukov,
- D: dátová množina obsahujúca n objektov.

Výstup: k zhlukov

Postup:

1. ľubovoľne vybrať k objektov z D ako počiatočne ťažiská zhlukov,
2. **opakovať**,
    a. znovu priradiť objekt zhluku, ktorému je objekt najpodobnejší, na základe priemernej hodnoty objektov v zhluku,
    b. aktualizovať ťažiská zhlukov – vypočítať priemernú hodnotu objektov každého zhluku,
3. **kým** žiadna zmena.



Obr. 5

Na obrázku 5 je príklad priebehu algoritmu k-means. Nech $k = 3$; prajeme si, aby vo výsledku boli tri zhluky.

a) náhodne vyberieme tri objekty ako počiatočné ťažiska zhlukov, označíme ich $+$. Každý objekt z dátovej množiny je priradený do zhlukov na základe vzdialenosti od týchto ťažísk. Tieto zhluky sú ohraničené bodkočiarkovanou líniou.
b) zaktualizujeme ťažiska zhlukov. Znovu prepočítame priemer každého zhluku na základe aktuálnych objektov v zhlukoch. Objekty sú následne rozdelené do zhlukov podľa najbližších nových ťažísk. Hranice týchto nových zhlukov sú vyznačené prerušovanou čiarou.
c) objekty postupne iteratívne prerozdeľujeme do zhlukov, aby sme vylepšili výsledok. Tomu sa hovorí iteratívne premiestňovanie. Nakoniec, až nenastane žiadne premiestňovanie, proces sa ukončí a metóda vráti výsledne zhluky.

Táto metóda je považovaná za základ zhlukovacích metód, najmä vďaka ľahkej implementácii. Pracuje správne s mnohými praktickými problémami, obzvlášť ak výsledné zhluky sú kompaktné a hypersférického tvaru. Algoritmus je relatívne efektívny v spracovávaní veľkého množstva dát z dôvodu výpočtovej zložitosti $O(nkt)$, kde n je počet dátových objektov, k je počet zhlukov a t je počet iterácií. Keďže k, t sú zvyčajne výrazne menšie než n, časová náročnosť algoritmu je približne lineárna.

Ako každá metóda, aj k-means má svoje nevýhody, konkrétne vrodené obmedzenia gradientných hladových algoritmov pri použití v optimalizačných úlohách.

Nehodí sa na odhaľovanie zhlukov s nekonvexným tvarom, alebo zhlukov s príliš rozdielnymi tvarmi. Navyše, ak by sa nachádzali v dátach nejaké hraničné alebo nezmyselné hodnoty, mohli by podstatne ovplyvniť ťažiská zhlukov.

Tieto nevýhody priniesli mnoho záujmu a snahy zo strán rôznych komunít a výsledkom sú jej mnohé modifikácie, ktoré tieto prekážky obchádzajú.

3.2.2 Metóda *k*-Medoids

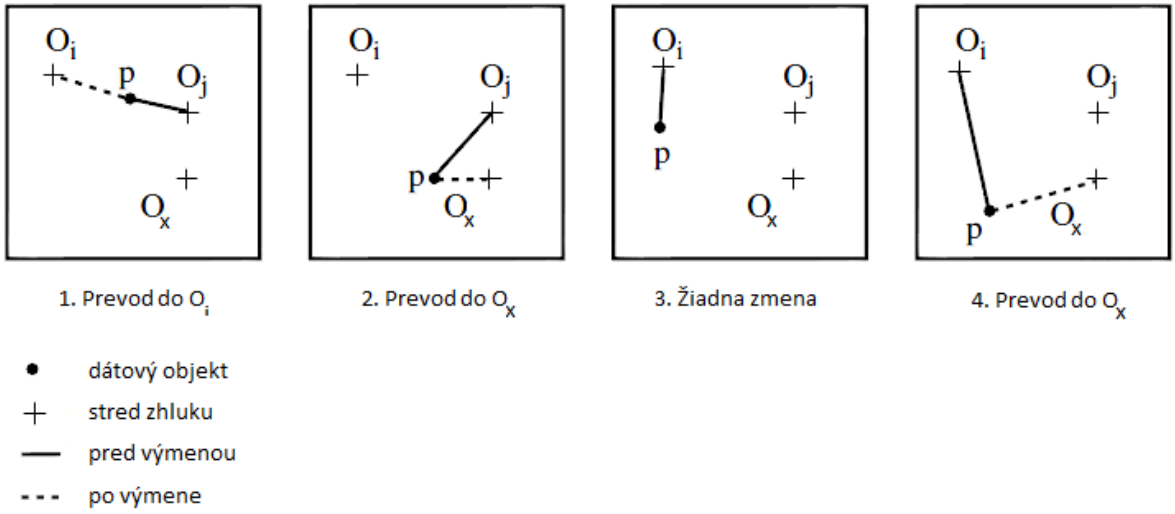
Spomínané slabosti algoritmu *k*-Means voči hraničným alebo nezmyselným hodnotám sú najmä dôsledkom použitia kritéria štvorcov chýb. Metóda *k*-Medoids [2] je modifikáciou algoritmu *k*-Means a snaží sa o redukciu tejto slabosti. Namiesto zisťovania priemernej hodnoty objektu v zhluku vyberá konkrétny exitujúci objekt, ktorý následne reprezentuje jeho zhluk. Každý zhluk má jedného reprezentanta spomedzi objektov v ňom. Každý zvyšný objekt je klasifikovaný podľa najväčšej podobnosti s reprezentantom. Zhlukovacia metóda potom postupuje na princípe minimalizácie rozdielov medzi každým objektom a jeho príslušným referenčným bodom. Tomu sa hovorí kritérium absolútnej chyby a je definované ako:

$$E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - o_j|$$

kde E je súčet absolútnych chýb všetkých objektov z dátovej množiny, p je bod v priestore reprezentujúci príslušný objekt v zhluku C_j a o_j je reprezentant zhluku C_j. Vo všeobecnosti táto metóda iteruje, kým každý reprezentatívny objekt nie je v skutočnosti medoid – ku stredu zhluku najviac umiestnený objekt. Metóda *k*-Medoids takto zhlukuje n objektov do k zhlukov.

Podrobnejšie. Počiatoční reprezentanti sú vybraní náhodne. Iteratívny proces zámeny reprezentantov za nové objekty pokračuje, kým sa kvalita zhlukov zlepšuje. Kvalita je stanovená použitím hodnotiacej funkcie, ktorá porovnáva priemernú rozdielnosť medzi objektom a reprezentantom príslušného zhluku.

Zistenie, či nereprezentatívny objekt o_x je dobrá náhrada za aktuálneho reprezentanta, objekt o_j, sa dá rozdeliť do nasledujúcich štyroch prípadov pre každý nereprezentatívny objekt p, ako je znázornené v obrázku 6.



Obr. 6

1. p patrí reprezentatívnemu objektu o_j. Ak je o_j vymenený za o_x ako nového reprezentanta a p je najbližšie k iným reprezentantom, ako napríklad o_i a $i \neq j$, tak p je prevedený do o_i.
2. p patrí reprezentatívnemu objektu o_j. Ak je o_j vymenený za o_x ako nového reprezentanta a p je najbližšie k reprezentantovi o_x, tak p je prevedený do o_x.
3. p patrí reprezentatívnemu objektu o_i, $i \neq j$. Ak je o_j vymenený za o_x ako nového reprezentanta a p je stále najbližšie k o_i, tak sa nič nemení.
4. p patrí reprezentatívnemu objektu o_i, $i \neq j$. Ak je o_j vymenený za o_x ako nového reprezentanta a p je najbližšie k o_x, tak p je preradené do o_x.

Vždy, keď sa vykoná prevod, rozdiel v absolútnej chybe E prispeje do hodnotiacej funkcie. Táto funkcia vypočíta rozdiel v hodnote absolútnej chyby, ak je reprezentatívny objekt vymenený za nový a pôvodne nereprezentatívny objekt. Celková cena výmeny je súčet cien spôsobených všetkými nereprezentatívnymi objektmi. Ak je celková cena negatívna, tak o_j je vymenený s o_x, keďže by aktuálna absolútna chyba bola zredukovaná. Ak je celková cena pozitívna, aktuálny reprezentatívny objekt o_j je považovaný za akceptovateľný a nič sa v iterácii nezmení.

Algoritmus: k-Medoids. Segmentujúca metóda založená na medoidoch alebo stredných objektoch.

Vstup:

- k: počet zhlukov,
- D: dátová množina obsahujúca n objektov.

Výstup: k zhlukov

Postup:

1. ľubovoľne vybrať k objektov z D ako reprezentatívne objekty,
2. **opakovať**,
    a. každý zvyšný objekt priradiť do zhluku s najbližším reprezentatívnym objektom,
    b. náhodne vybrať nereprezentatívny objekt o_x,
    c. vypočítať celkovú cenu S výmeny reprezentatívneho objektu o_j s o_x,
    d. ak $S < 0$, vymeniť o_j s o_x a vytvoriť tak novú množinu k reprezentatívnych objektov,
3. **kým** žiadna zmena.

Tento algoritmus sa po úvodnej náhodnej selekcii k reprezentantov opakovane snaží zlepšiť voľbu týchto reprezentantov. Potom sú zanalyzované všetky páry objektov, kde jeden objekt je braný ako reprezentant a druhý opačne. Kvalita výsledného zhlukovania je vypočítaná z každej takejto dvojice. Objekt o_j je vymenený s objektom, ktorý spôsobuje najväčšiu redukciu v chybe. Množina najvhodnejších objektov pre každý zhluk z jednej iterácie tvorí množinu reprezentatívnych objektov v ďalšej. Posledná množina predstavuje konkrétne medoidy zhlukov. Náročnosť každej iterácie je $O(k(n-k)^2)$. Pri vysokých hodnotách n a k môže byť tento algoritmus veľmi náročný.

Metóda *k*-Medoids je robustnejšia, než *k*-Means, čo sa týka hraničných a nezmyselných dát, pretože medoid je práve takýmito hodnotami menej ovplyvňovaný, než priemer. Na druhú stranu je ale výpočtovo náročnejšia.

3.2.3 Fuzzy *c*-Means clustering

V predošlých metódach sme dáta zhlukovali do ostrých zhlukov s jasne vymedzenými hranicami. Algoritmus Fuzzy *c*-Means (FCM) funguje na báze fuzzy logiky. To znamená, že dátové objekty môžu patriť viacerým zhlukom. Táto vlastnosť je indikovaná stupňom príslušnosti dátového objektu k danému zhluku.

FCM [9][21] je jeden z najpoužívanejších fuzzy algoritmov. Snaží sa rozdeliť konečnú množinu dátových objektov do množiny c fuzzy zhlukov s ohľadom na niektoré dané kritéria.

Nech D=$\{x_1, x_2, \cdots, x_n\}$ je konečná množina dátových objektov a $c < n$ je počet zhlukov. Algoritmus FCM rozdelí jednotlivé dátové objekty do c zhlukov $Z_1, \cdots, Z_c$, pričom každý objekt môže patriť viacerým zhlukom. Stupeň príslušnosti objektov do zhlukov je vyjadrený maticou U $(c \times n)$, kde u_{ij} ($u_{ij} \in \langle 0,1 \rangle$) určuje hodnotu členskej funkcie pre objekt x_i a zhluk Z_j.

Zároveň musí platiť, že súčet hodnôt pre jeden objekt cez všetky zhluky je rovný jednej, t.j. $\sum_{j=1}^{c} u_{ij} = 1, \forall i$. Metóda sa snaží iteratívne minimalizovať hodnotiacu funkciu:

$$\sum_{i=1}^{n}\sum_{j=1}^{c} {u_{ij}}^{m} d\left(x_i, c_j\right)^2$$

kde $m \in \langle 1, \infty)$ je fuzzifikačný parameter (vyššie m znamená rozmazanejšie zhluky vo výsledku) a $d\left(x_i, c_j\right)$ je vzdialenosť medzi objektom x_i a zhlukom c_j. Metóda navyše vyžaduje parameter $\varepsilon > 0$, ktorý určuje požadovanú presnosť, inými slovami, aká malá musí byť posledná aktualizácia stredov zhlukov na to, aby algoritmus mohol ukončiť výpočet.

Algoritmus: Fuzzy *c*-Means. Segmentujúca metóda založená na princípe fuzzy logiky.

Vstup:

- c: počet zhlukov,
- D: dátová množina obsahujúca n objektov,
- m: fuzzifikačný paramater,
- ε: požadujúca presnosť algoritmu.

Výstup: c zhlukov a matica U.

Postup:

1. na základe vstupu inicializovať maticu U^0, nastaviť iteračný index (horný index pri U) na 0,
2. v kroku k: vypočítať stredy zhlukov za použitia U^k

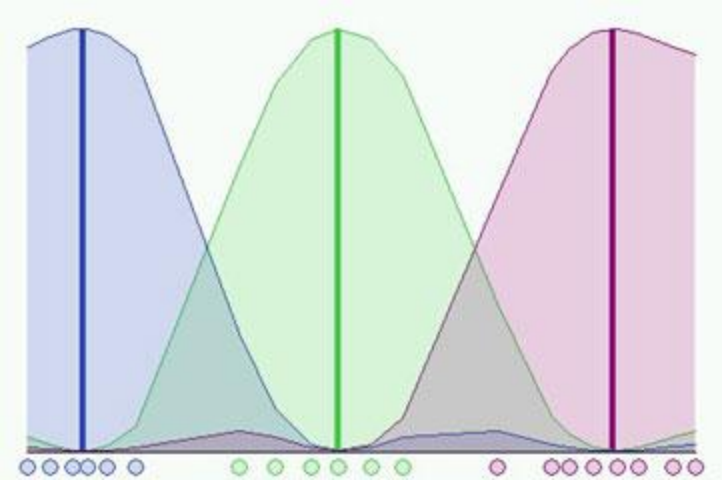
$$c_j = \frac{\sum_{i=1}^{n} {u_{ij}}^{m} x_i}{\sum_{i=1}^{n} {u_{ij}}^{m}}$$

3. aktualizovať U^k a U^{k+1}

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

4. ak $\|U^{k+1} - U^k\| < \varepsilon$, tak ukončiť výpočet, inak sa vráť na krok 2.

Metóda FCM trpí zásadným problémom: nedostatočne vhodná identifikácia počiatočných zhlukov a prítomnosti hraničných a nezmyselných dát. Výpočtová náročnosť FCM je $O(ndc^2i)$, kde n je počet dátových objektov, d je počet rozmerov, c je počet zhlukov a i je počet iterácií.



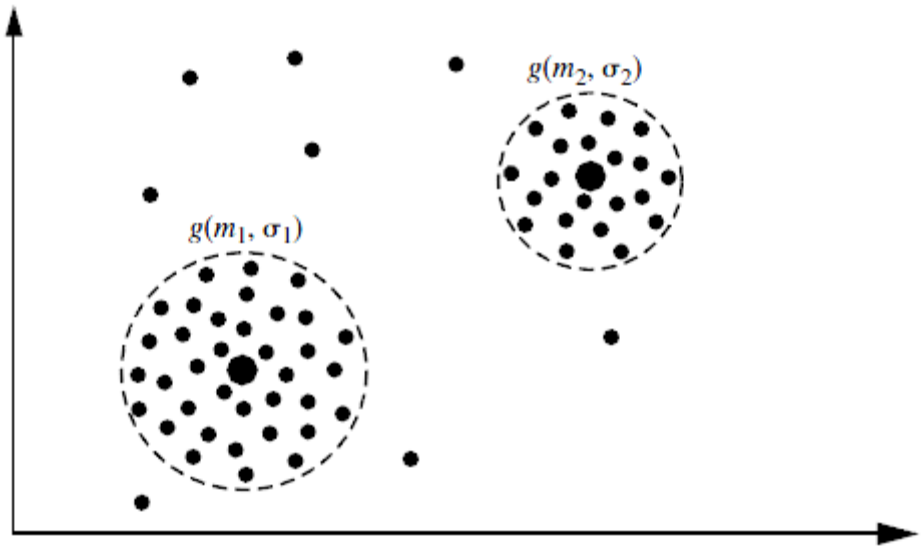
Obr. 7: Výsledok zhlukovania pomocou FCM algoritmu. Vizualizácia v jednorozmernom priestore. Fuzzyfikačný parameter v tomto prípade je veľmi nízky (0.01) – je požadovaná vysoká presnosť, čo je výpočtovo náročnejšie (vyšší počet iterácií).

3.2.4 Expectation-Maximization clustering

Táto metóda sa dá zaradiť medzi modelovo orientované zhlukovacie metódy. Vypĺňa miesto medzi dátami a nejakým matematickým modelom. Takéto typy metód fungujú často na princípe, že dáta sú generované zmesou základných pravdepodobnostných rozdelení.

V praxi je možné každý zhluk matematicky reprezentovať pomocou parametrického pravdepodobnostného rozdelenia. Celé dáta sú zmesou takýchto rozdelení, kde každému individuálnemu rozdeleniu sa typicky hovorí čiastkové (component) rozdelenie. Takto môžeme zhlukovať dáta použitím konečného hustotného modelu pozostávajúceho z k pravdepodobnostných rozdelení.

Problém spočíva tom, ako najlepšie odhadnúť parametre týchto pravdepodobnostných rozdelení, aby čo najlepšie sedeli na dáta. Nasledujúci obrázok je jednoduchý príklad konečného hustotného modelu. Sú tam dva zhluky. Každý sa správa podľa normálneho rozdelenia s jeho vlastným priemerom a smerodajnou odchýlkou.



Obr. 8: Každý zhluk sa dá reprezentovať pravdepodobnostným rozdelením, centrované v priemere a so smerodajnou odchýlkou. Máme tu dva zhluky, korešpondujúce Gausovému rozdeleniu. Prerušované čiary reprezentujú prvé smerodajné odchýlky.

Algoritmus Expectation-Maximization (EM) sa dá považovať za rozšírenie metódy *k*-Means, ale namiesto priraďovania každého objektu do určeného zhluku, EM priradí každý objekt zhluku podľa váhy, ktorá reprezentuje pravdepodobnosť členstva daného objektu v danom zhluku. Inými slovami, medzi zhlukmi neexistujú žiadne striktné hranice – nové priemery sú počítané na základe váh.

EM začína s počiatočným odhadom parametrov modelu (vektor parametrov). Iteratívne prehodnocuje objekty voči zmesi hustôt rozdelení vytvorenej pomocou vektora parametrov. Prehodnotené objekty sú potom použité na aktualizáciu vektora parametrov. Každému objektu je priradená pravdepodobnosť, s ktorou by nadobúdal určitú množinu hodnôt, ak by bol členom daného zhluku.

Algoritmus funguje nasledovne:

1. Odhadnúť počiatočný vektor parametrov. To zahŕňa náhodný výber *k* objektov, ktoré reprezentujú priemery zhlukov (podobne ako v k-Means), ako aj odhady pre ďalšie parametre.

2. Iteratívne zjemňovať parametre (alebo zhluky) podľa nasledujúcich dvoch krokov:

   a. **Očakávanie (expectation):** priradiť každý objekt x_i zhluku C_k s pravdepodobnosťou

   $$P(x_i \in C_k) = p(C_k|x_i) = \frac{p(C_k)p(x_i|C_k)}{p(x_i)}$$

   kde $p(x_i|C_k) = N(m_k, E_k(x_i))$ sa správa podľa normálneho rozdelenia okolo priemeru m_k s očakávaním E_k. Inými slovami, tento krok kalkuluje pravdepodobnosť členstva objektu x_i pre každý zhluk. Tieto pravdepodobnosti sú očakávane členstvá objektu x_i v zhlukoch.

   b. **Maximalizácia (maximization):** použiť pravdepodobnostné odhady z predošlého kroku na znovu-odhadnutie, resp. zjemnenie parametrov modelu. Napríklad:

   $$m_k = \frac{1}{n}\sum_{i=1}^{n}\frac{x_i P(x_i \in C_k)}{\sum_j P(x_i \in C_j)}.$$

   Tento krok je „maximalizácia“ pravdepodobnosti rozdelení.

Metóda EM je jednoducho a ľahko implementovateľná. V praxi rýchlo konverguje, ale nemusí dosiahnuť globálneho optima. Pre niektoré optimalizačné funkcie je konvergencia garantovaná. Výpočtová náročnosť je lineárna – d (počet vstupných vlastností), n (počet objektov) a t (počet iterácií) [2].

3.2.5 Kohonenove mapy

Kohonenove mapy, alebo samoroganizačné mapy, (SOM) navrhol T. Kohonen a patria k špeciálnemu typu neurónových sietí. Pre lepšiu orientáciu tu v jednoduchosti popíšem problematiku neurónových sietí.

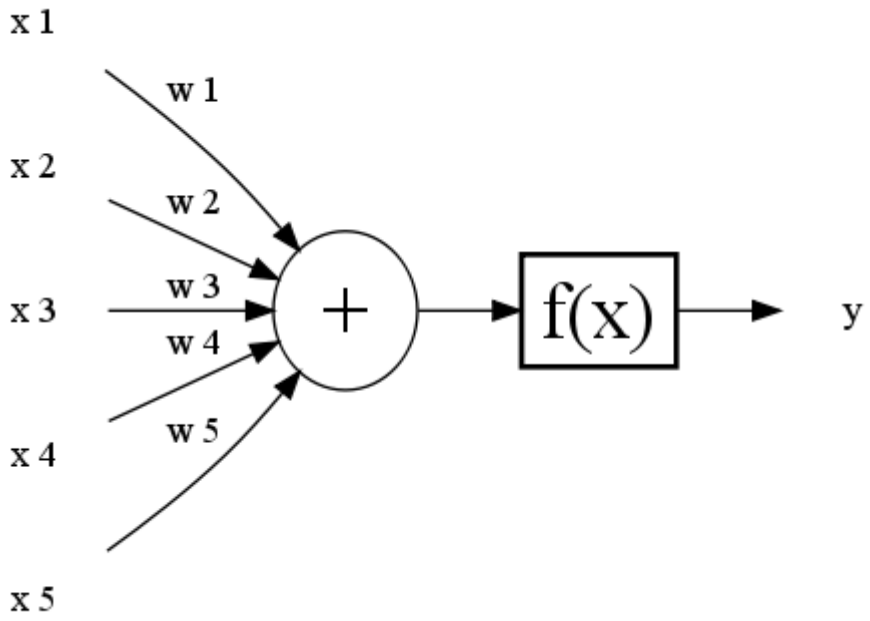
Princíp neurónových sietí je založený na zjednodušenom modely reálnych neurónov. Každý neurón môže mať určitý počet vstupov (x), pričom každý vstup môže mať rôznu váhu (w). Výstup (y) jedného neurónu môže byť vstupom iných neurónov, a tak vzniká neurónová sieť.

Vstupy a ich váhy sú z pravidla reálne čísla z intervalu $\langle 0,1 \rangle$. Samotný zjednodušený neurón potom funguje asi tak, že najprv vynásobí každú hodnotu vstupu príslušnou váhou a potom všetky tieto hodnoty sčíta. To však nie je všetko. Na tento súčet sa ešte aplikuje tzv. aktivačná, resp. prahová funkcia a až jej výsledok je potom výstupom neurónu.

Táto aktivačná funkcia zabezpečí, aby výstup neurónu bol znova v intervale $\langle 0,1 \rangle$, a teda vhodný ako vstup do ďalšieho neurónu. Prahová funkcia však môže slúžiť aj na transformáciu výstupu.

Nasadenie neurónových sietí na daný problém ma väčšinou dve fázy. Fázu učenia a fázu používania. Vo fáze učenie sa pomocou testovacích vstupov rôznymi technikami upravujú váhy vstupov neurónov, až kým nedávajú požadovaný výsledok. Vo fáze používania potom naučená neurónová sieť reaguje na vstupy, pričom dokáže rozpoznávať aj vstupy, na ktoré nebola naučená.

Najjednoduchší model neurónu sčíta všetky vstupy (x) vážené odpovedajúcimi váhami (w) a výsledok vloží do prahovej funkcie (f). Ako prahová funkcia sa napríklad používajú rôzne parametrizovateľné sigmoidy a funkcie signum.



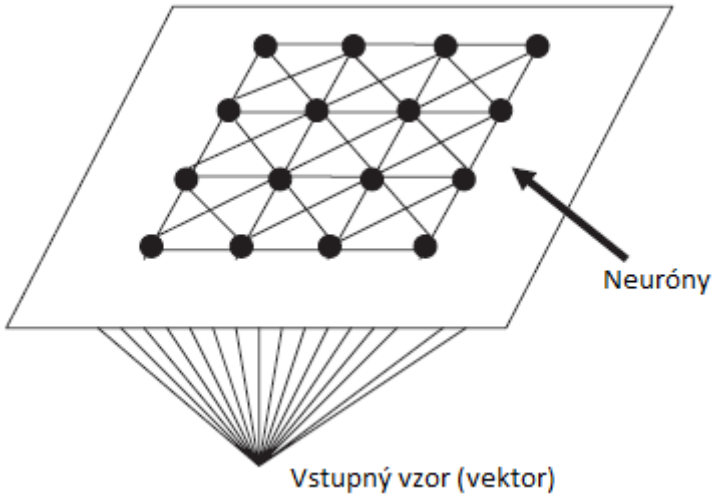
Obr. 9: Matematický model neurónu (perceptron)

$$y = f(\sum_{i=1}^{n} w_i x_i)$$

Prahová funkcia by mala byť nelineárna a aspoň veľmi približne modelovať reálne správanie neurónu.

- Log–Sigmoida (logistická funkcia): $f(x) = \frac{1}{1+e^{-\alpha x}}$
- Tan-Sigmoida: $f(x) = \tanh(\alpha x)$
- Signum: $f(x) = \begin{cases} 1, ak\ x > 0 \\ 0, ak\ x = 0 \\ -1, ak\ x < 0 \end{cases}$

SOM sa od snáď najznámejšieho modelu so spätným šírením dosť líšia. Jednak používajú iný model neurónov, ich aktivácie, ale aj iný spôsob učenia, Kohonenove mapy totiž fungujú na princípe učenia bez učiteľa. To znamená, že algoritmus učenia nemá informáciu o požadovaných aktivitách výstupných neurónov v priebehu trénovania. Adaptácia váh prebieha len na základe trénovacej množiny, ktorá je v sieti reprezentovaná formou vstupných vzorov (vektorov) s binárnymi alebo reálnymi zložkami.



Obr. 10: Na tomto obrázku je znázornená základná architektúra SOM. Vstupný vzor je plne prepojený so všetkými neurónmi uloženými v dvojrozmernej mriežke. Veľkosť siete, v tomto prípade 4 × 4, musí byť určená dopredu.

Špecifickou črtou SOM je to, že po natrénovaní umožňuje realizovať zobrazenie zachovávajúce topológiu trénovacej množiny dát. To znamená, že ľubovoľné dva vzory blízke vo vstupnom priestore evokujú v sieti odozvy na neurónoch, ktoré sú fyzicky tiež blízke (vo výstupnom priestore). Vzdialenosť dvoch neurónov sa rovná ich euklidovskej vzdialenosti na mriežke.

Samoorganizácia alebo taktiež učenie v SOM je založené na tzv. Hebbovom pravidle učenia [22]. Práve T. Kohonen navrhol jednu z možných formalizácií tohto pravidla, ktorá sa používa na trénovanie SOM. Váhy medzi vstupom a neurónmi v mriežke inicializujeme na malé náhodné čísla. Nech $A_{train} = \{(\bar{x}^1, d^1), (\bar{x}^2, d^2), \dots, (\bar{x}^p, d^p), \dots, (\bar{x}^P, d^P)\}$ je množina trénovacích vzorov, teda množina P dvojíc vstupných vektorov a k nim prislúchajúcich požadovaných výstupov. Neuróny v sieti sú lineárne s nulovými prahmi, t.j. výstup i–tého neurónu je:

$$o_i = \sum_{j=1}^{m} w_{ij} x_j = \bar{w}_i \bar{x},$$

kde m je dimenzia vstupu. Nech $i = 1, \dots, n$ sú neuróny v mriežke. Na vstup siete sa postupne náhodne vyberajú jednotlivé vzory a pre každý takýto vzor nájdeme víťazný neurón. Informácia o pozícii víťazného neurónu sa získa procesom zvaným súťaženie. Je to proces hľadania neurónu, ktorý najviac reaguje na daný vstup $\bar{x}$, napríklad hľadať maximum výstupu lineárneho neurónu, čiže:

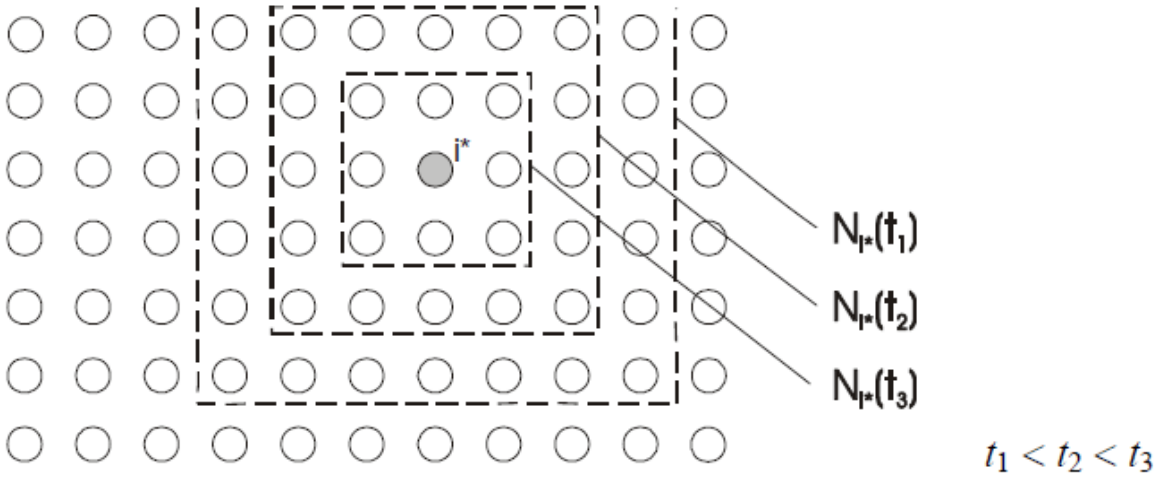
$$i^* = argmax(\bar{w}_i, \bar{x}),$$

kde i^* je pozícia víťazného neurónu. Víťaz súťaženia získava právo reprezentovať predložený vzor.

Po nájdení víťaza nasleduje adaptácia váh – učenie. To zabezpečí, že váhové vektory víťazného neurónu a jeho topologických susedov sa posunú smerom k aktuálnemu vstupu podľa vzťahu

$$\bar{w}_i(t+1) = \bar{w}_i(t) + \alpha(t)N(i^*, i)[\bar{x}(t) - \bar{w}_i(t)],$$

kde $\alpha(t)\epsilon(0,1)$ predstavuje monotónne klesajúcu rýchlosť učenia až k nule, $\bar{x}(t)$ je vstupný vektor. $N(i^*, i)$ je funkcia okolia, ktorá definuje rozsah kooperácie medzi neurónmi, inými slovami, koľko váhových vektorov prislúchajúcich neurónom v okolí víťaza bude adaptovaných, a do akej miery.



Obr. 11: Zmenšujúce sa okolie víťazného neurónu

Medzi najpoužívanejšie funkcie patrí pravouhlé okolie a gaussovské okolie.

Pravouhlé okolie je založené na manhattanskej vzdialenosti medzi neurónmi i^* a i v mriežke mapy (suma absolútnych hodnôt rozdielov ich súradníc). Najlepšie výsledky sa dosiahnu vtedy, ak sa veľkosť okolia s časom diskrétne zmenšuje.

Gaussovské okolie je definované nasledovne:

$$N(i^*, i) = \exp\left(-\frac{d_E^2(i^*, i)}{\lambda^2(t)}\right),$$

kde $d_E(i^*, i)$ je euklidovská vzdialenosť neurónov i^* a i. Parameter $\lambda(t)$ klesá s časom k nule, čím sa zabezpečuje zmenšovanie okolia počas učenia. Používanie takto definovanej funkcie okolia vedie k aktualizácii polohy všetkých neurónov v každej iterácii algoritmu, čo má za následok zvýšenú časovú náročnosť.

Algoritmus: Učenie kohonenových máp

Vstup:

- A_{train}: trénovacia množina
- α_0: parameter učenia
- t_{max}: maximálny počet iterácií
- $N(i^*, i)$: Funkcia okolia
- λ_0: parameter funkcie okolia

Výstup: hodnoty členských funkcií

Postup:

1. Inicializovať počiatočné váhy (napr. náhodné hodnoty z intervalu $(-0.5,0.5)$), nastaviť iterátory $t = 0$, $p = 1$, kde t je čas a p je index vzoru. Inicializovať parameter učenia $\alpha(t)$ a parameter okolia $\lambda(t)$,
2. na vstup dať vzor $\bar{x}^p \in A_{train}$ a nájsť víťazný neurón,
3. upraviť váhy víťaza a jeho topologických susedov,

$$\bar{w}_i(t+1)\begin{cases}\bar{w}_i(t) + \alpha(t)\big(\bar{x}(t) - \bar{w}_i(t)\big), pre\ neuróny\ z\ okolia\ definovanými\ N_{i^*}(t)\\ \bar{w}_i(t), pre\ ostatné\ neuróny\end{cases}$$

4. aktualizovať hodnoty α a λ,
5. ak $p < P$, položiť $p = p + 1$ a prejsť na krok 2, inak prejsť na krok 6,
6. ak $t = t_{max}$, ukončiť učenie, inak položiť $p = 1$, upraviť parameter učenia a prejsť na krok 2. Začína sa nový trénovací cyklus ($t = t + 1$), t.j. nový prechod cez A_{train}

Je dôležité, aby α bola monotónne klesajúca funkcia z nejakej hodnoty blízkej 1, s malými hodnotami počas fázy doladenia (napr. lineárna lomená funkcia, exponenciálna funkcia atď). Nezáleží ani na presnom počte iterácií. Kohonen uvádza empiricky získanú pomôcku, podľa ktorej počet iterácií má byť minimálne 500-násobok počtu neurónov v sieti.

Kohonenove mapy sú z hľadiska implementácie a výpočtovej náročnosti pomerne o dosť zložitejšie, než metódy, ktoré sú uvedené vyššie, no majú však aj svoje výhody. Najväčšia z nich je topologická štruktúra medzi neurónmi. Vďaka nej získavame nielen informáciu o rozdelení trénovacích vzorov do zhlukov, ale určíme konkrétnejšiu štruktúru trénovacej množiny. Podobné vstupy aktivujú neuróny, ktoré sú blízko v topologickej štruktúre, a práve to značí ich podobnosť. Oproti tomu rozdielne vstupy aktivujú neuróny, ktoré sú v topologickej štruktúre od seba ďaleko a tým je určené nielen to, že vstupy sú rozdielne a nepatria do rovnakého zhluku, ale taktiež, že sú podstatne rozdielne. Ďalšou výhodou je mapovanie multidimenzionálneho vstupu do nízkodimenzionálnej mriežky [23].

3.3 Zhluková validácia

Táto kapitola sa zaoberá problematikou posudzovania kvality výsledkov zhlukovacích metód. Najprv rozoberiem všeobecné vlastnosti techník z oblasti zhlukovej validácie a v ďalších častiach potom detailne rozoberiem najčastejšie používané metódy [24][25].

Zhluková analýza predstavuje väčšinou proces učenia bez učiteľa, takže ich hodnotenie je celkom dôležité. V zhlukovacom procese nie sú preddefinované žiadne triedy, takže je dosť zložité nájsť vhodnú metriku na vyhodnotenie výsledkov z algoritmov, teda či sú akceptovateľné, alebo nie.

Existuje niekoľko známych prístupov k zhlukovej validácii. Hlavné nevýhody týchto ukazovateľov sú, že nedokážu vyhodnotiť ľubovoľne tvarované zhluky, keďže zvyčajne vyberú reprezentatívny bod z každého zhluku a počítajú vzdialenosti týchto bodov a iné parametre na nich založené (napr. variácia).

Výsledok jednej zhlukovacej metódy sa pri použití na tých istých dátach môže diametrálne odlišovať od inej. Cieľom týchto validácií je nájsť segmentáciu, ktorá najlepšie opisuje použité dáta. Bežne sa používajú dvojrozmerné vizualizácie, pretože sú ľahko čitateľné, ale v prípade vysokorozmerných dát to nie je vôbec triviálne, a práve z toho dôvodu vznikli tieto metódy.

V procese ohodnocovania výsledkov zhlukovacích algoritmov boli navrhnuté dve kritéria pre výber optimálnej zhlukovacej metódy[26]:

- Kompaktnosť: člen každého zhluku by mal byť čo najbližšie druhému. Bežná miera kompaktnosti je variácia.

- Separácia: zhluky by mali byť navzájom čo najlepšie separované. Tri najbežnejšie prístupy k meraniu vzdialenosti dvoch zhlukov sú: vzdialenosť medzi najbližšími, najvzdialenejšími a strednými objektmi zhlukov.

Taktiež, existujú tri rôzne techniky pre ohodnocovanie výsledkov klastrovacích algoritmov [27]:

- Externé kritéria
- Interné kritéria
- Relatívne kritéria

Oboje, externé a interné sú založené na štatistických metódach a sú výpočtovo veľmi náročné. Externé validačné metódy hodnotia zhlukovanie podľa nejakej predstavy špecifikovanej užívateľom. Interné, naopak, fungujú na základe určitých metrík, ktoré sú odvodené od dátovej množiny a schémy zhlukovania. Ako som už spomínal, ich hlavnou nevýhodou je vysoká výpočtová náročnosť.

Základom relatívnych kritérií je porovnanie rôznych zhlukovacích schém. Jeden alebo viacero klastrovacích algoritmov je na tých istých dátach viackrát spustených, samozrejme s rozdielnymi vstupnými parametrami. Cieľ relatívnych kritérií je vybrať z týchto výsledkov najlepšiu zhlukovaciu metódu. Základom porovnávania je validačný index, ktorých existuje niekoľko [28][29][30].

Tie najpoužívanejšie rozoberiem v nasledujúcej časti textu.

Validačné ukazovatele

Tieto indexy slúžia na určovanie kvality výsledku zhlukovania v porovnaní s výsledkami iných zhlukovacích metód, prípadne rovnakých, ale s inými vstupnými parametrami. Validačné ukazovatele sú vhodné pre posudzovanie kvality ostrých zhlukov (s jasne vymedzenými hranicami, ktoré sa neprekrývajú). V nasledujúcom texte budeme používať nasledovné definície.

n_c	Počet zhlukov
d	Počet dimenzií
$d(x, y)$	Vzdialenosť medzi dvoma dátovými prvkami
$\overline{X}_j$	Očakávaná hodnota v *j*-tej dimenzii
$\|X\|$	Norma vektora $\vec{X}$
$\overline{c_i}$	Ťažisko *i*-tého zhluku
c_i	*i*-tý zhluk
$\|c_i\|$	Počet prvkov v *i*-tom zhluku
n	Počet prvkov (dátových objektov) v celej dátovej množine
$\vec{x}_j$	*j*-tý prvok z dátovej množiny

3.3.1 Dunnov index

Tento ukazovateľ bol uvedený v [31]. Index je definovaný nasledovne:

$$D = \min_{i=1\cdots n_c} \left\{ \min_{j=i+1\cdots n_c} \left(\frac{d(c_i, c_j)}{\max_{k=1\cdots n_c} \left(diam(c_k)\right)} \right) \right\}$$

kde $d(c_i, c_j) = \min_{x\epsilon c_i, y\epsilon c_j}\{d(x,y)\}$ je vzdialenosť dvoch zhlukov a $diam(c_i) = \max_{x,y\epsilon c_i}\{d(x,y)\}$ je priemer zhluku, resp. kompaktnosť zhluku.

Ak dátová množina obsahuje dobre separované zhluky, vzdialenosti medzi nimi sú zvyčajne vysoké a priemery zhlukov malé [25], takže vyššia hodnota znamená lepšiu konfigurácia zhlukov. Hlavné nevýhody Dunnovho indexu sú nasledujúce: výpočet ukazovateľa je časovo náročný a tento index je veľmi citlivý na nezmyselné dáta (v takýchto dátach môže maximálny priemer zhluku byť vysoký).

Existuje viacero indexov, založených práve na Dunnovom indexe, len s tým rozdielom, že pre vzdialenosti a priemery zhlukov používajú iné prístupy [27][32], ako napríklad Alternatívny Dunnov index, ktorý sa pri posudzovaní vzdialeností opiera o trojuholníkovú nerovnosť. Výhodou Dunnovho indexu je naopak to, že jeho hodnota nerastie s rastúcim počtom zhlukov, čiže vyššia hodnota pri vyššom počte zhlukov indikuje, že výsledná konfigurácia je lepšia.

3.3.2 Ukazovateľ Davies Bouldin

Index Davies – Bouldin funguje na princípe miery podobnosti zhlukov (R_{ij}), miery priemernej vzdialenosti od ťažísk jednotlivých zhlukov s_i a miery rozdielnosti zhlukov d_{ij}. Miera podobnosti zhlukov zhlukov (R_{ij}) môže byť definovaná ľubovoľne, ale musí splňovať nasledujúce podmienky:

- $R_{ij} \geq 0$
- $R_{ij} = R_{ji}$
- ak $s_i = 0$ a $s_j = 0$, tak $R_{ij} = 0$
- ak $s_j > s_k$ a $d_{ij} = d_{ik}$, tak $R_{ij} > R_{ik}$
- ak $s_j = s_k$ a $d_{ij} < d_{ik}$, tak $R_{ij} > R_{ik}$

Najčastejšie je R_{ij} definované ako:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}},$$

kde $d_{ij} = d(\bar{c}_i, \bar{c}_j)$ a $s_i = \frac{1}{\|c_i\|}\sum_{x \epsilon c_i} d(\vec{x}, \bar{c}_i)$.

Davies – Bouldinov index je potom definovaný ako:

$$DB = \frac{1}{n_c}\sum_{i=1}^{n_c} R_i,$$

kde $R_i = \max_{j=1\ldots n_c, i \neq j}(R_{ij}), i = 1 \ldots n_c$.

Tento index, z definície, určuje priemernú mieru podobnosti medzi každým zhlukom a jemu najpodobnejším a keďže zhluky musia byť kompaktné a separované, nižšie hodnoty Davies – Bouldinového indexu znamenajú lepšiu konfiguráciu zhlukov.

Inými slovami, tento index je pomerovou funkciou sumy vnútorného rozloženia zhluku k medzizhlukovej distribúcii. Spolu s Dunnovým indexom sú vhodné na určenie vhodného počtu zhlukov, pretože nerastú s rastúcim počtom zhlukov.

3.3.3 RMSSTD a RS ukazovatele

Myšlienka RMSSTD (root-mean-square standard deviation) indexu [33] spočíva v meraní homogenity zhlukov na základe rozptylu a priemeru hodnôt dátových objektov. Nižšia hodnota RMSSTD indexu znamená lepšie zhlukovanie.

Keďže RMSSTD index vyjadruje smerodajnú odchýlku všetkých atribútov použitých pri zhlukovacom procese.

Pre celé rozdelenie je index RMSSTD definovaný nasledovne:

$$RMSSTD = \sqrt{\frac{\sum_{i=1}^{n_c}\sum_{k=1}^{f}\sum_{x\in c_i}\left(x(k)-\bar{c}_i(k)\right)^2}{\sum_{i=1}^{n_c}(f-1)}},$$

kde f je počet atribútov použitých v zhlukovaní a $x(k)$ je hodnota atribútu k objektu x.

Hodnota RMSSTD indexu klesá pri stúpajúcom počte zhlukov, pretože čím viac je zhlukov, tým je v každom zhluku menší rozptyl.

Myšlienkou RS (R-squared) indexu [33] je meranie rozdielnosti zhlukov. Formálne, posudzuje stupeň homogenity medzi zhlukmi. Hodnoty RS sú v rámci intervalu $\langle 0,1\rangle$, kde 0 značí, že medzi zhlukmi nie je žiaden rozdiel a 1 indikuje, že sú medzi nimi významné rozdiely. Je nutné definovať nasledujúci pojem – súčty štvorcov (SS), ktoré sa v tomto ukazovateli používajú. Máme tri typy SS:

1. SS_w: súčet štvorcov v rámci zhluku. Pre zhluk c_i, ktorý pozostáva z $\|c_i\|$ objektov $\vec{x}_{i_1}, \dots, \vec{x}_{i_{\|c_i\|}}$ a ťažisko zhluku $\bar{c}_i$, SS_w definujeme ako

$$SS_w(i) = \sum_{j=1}^{\|c_i\|}\left(\vec{x}_{ij}-\bar{c}_i\right)^2.$$

   Celkový súčet štvorcov SS_w definujeme takto:

$$SS_w = \sum_{i=1}^{n_c}\sum_{j=1}^{\|c_i\|}\left(\vec{x}_{ij}-\bar{c}_i\right)^2.$$

2. SS_b: súčet štvorcov medzi zhlukmi. Na dátovej množine pozostávajúcej z n_c ťažísk zhlukov $c_1, \dots, c_{n_c}$ a celkového stredu zhlukov $\bar{C}$ definujeme SS_b nasledovne:

$$SS_b = \sum_{i=1}^{n_c} (\bar{c}_i - \bar{C})^2.$$

3. SS_t: celkový súčet štvorcov definujeme ako

$$SS_t = SS_w + SS_b,$$

a RS je potom pomer SS_b ku SS_t, čiže:

$$RS = \frac{SS_b}{SS_t}.$$

SS_b je miera rozdielnosti zhlukov, takže čím sú dva zhluky separovanejšie, tým väčšia je veľkosť SS_b. Navyše, SS_w je miera kompaktnosti jedného zhluku. Čím je hodnota SS_w menšia, tým je zhluk kompaktnejší. Nakoniec vieme zapísať RS ako

$$RS = \frac{SS_t - SS_w}{SS_t}.$$

RS je mierou homogenity medzi zhlukmi a jeho hodnota sa stále nachádza v intervale $\langle 0,1 \rangle$.

RMSSTD index popisuje vlastnosť kompaktnosti a RS index naopak, vlastnosť separácie. Práve z toho dôvodu sa tieto ukazovatele často používajú súčasne.

3.3.4 Xie-Beni index

Nasledujúce validačné techniky patria medzi ukazovatele, ktoré sa opierajú o členské hodnoty. Inými slovami, sú to techniky validácie pri fuzzy zhlukovaní. V porovnaní s predošlými metódami, ktoré chápali zhluky ako ostré, navzájom disjunktné množiny objektov, nasledujúce techniky počítajú so zhlukmi, ktoré sa môžu prekrývať, teda miera patričnosti objektu v zhluku je vyjadrená hodnotou členskej funkcie daného zhluku. Xie-Beni index [34] je definovaný ako:

$$XB = \frac{\sum_{i=1}^{c_i} \sum_{j=1}^{n} u_{ij}^2 \left\| \vec{x}_j - \bar{c}_i \right\|^2}{n(\min_{i \neq k} \|\bar{c}_i - \bar{c}_k\|^2)},$$

kde u_{ij} je vyjadrenie stupňa príslušnosti objektu $\vec{x}_i$ k zhluku c_j s ťažiskom $\bar{c}_j$.

Malé hodnoty XB značia kompaktné a dobre separované zhluky. Je nutné spomenúť, že XB je monotónne klesajúco závislé na počte zhlukov, obzvlášť pri vysokom počte (blízko k n). Jeden spôsob eliminácie tejto tendencie je stanoviť si počiatočný bod monotónnosti, teda c_{max}, a minimálnu hodnotu indexu hľadať len na intervale $\langle 2, c_{max} \rangle$.

3.3.5 Partition index a Separation index

Partition ukazovateľ (SC) je vyjadrený ako pomer kompaktnosti a separácie zhlukov [35]. Pre jeho formálny popis je potrebné definovať nasledujúce pojmy:

- Fuzzy odchýlka objektu (vzoru) $\vec{x}_i$ od zhluku c_j, teda:

$$d_{ij} = u_{ij}^m \left(\vec{x}_i - \bar{c}_j\right)^2,$$

  kde $\bar{c}_j$ je ťažisko zhluku c_j, m je fuzzifikačný parameter zhlukovania a u_{ij} je vyjadrenie stupňa príslušnosti objektu $\vec{x}_i$ k zhluku c_j s ťažiskom $\bar{c}_j$.

- Rozptyl fuzzy zhluku c_i je definovaný ako:

$$\sigma_i = \sum_{j=1}^{n} d_{ij}^2$$

- Fuzzy kardinalita zhluku je definovaná ako:

$$k_i = \sum_{j=1}^{n} u_{ij}$$

Kompaktnosť fuzzy zhluku c_i je potom pomer jeho rozptylu a kardinality, teda:

$$\pi_i = \frac{\sigma_i}{k_i}.$$

Separáciu (s_i) fuzzy zhluku c_i môžeme vyjadriť ako súčet vzdialeností od jeho ťažiska $\bar{c}_i$ do ťažísk ostatných $n_c - 1$ zhlukov, teda:

$$s_i = \sum_{k=1}^{n_c} (\bar{c}_k - \bar{c}_i)^2$$

Nakoniec, validačný index zhluku i je vyjadrený ako pomer (π_i/s_i) jeho kompaktnosti k jeho separácii. SC sa získa súčtom takýchto pomerov cez všetky zhluky, presnejšie:

$$SC = \sum_{i=1}^{n_c} \frac{\sum_{j=1}^{n} u_{ij}^m \left(\vec{x}_i - \bar{c}_j\right)^2}{n_i \sum_{k=1}^{n_c} (\bar{c}_k - \bar{c}_i)^2}$$

Separačný index (S) je podobný [34], no sú tu dva rozdiely.

$$S = \sum_{i=1}^{n_c} \frac{\sum_{j=1}^{n} u_{ij}^2 \left(\vec{x}_i - \bar{c}_j\right)^2}{n \left(\min_{i,k} (\bar{c}_k - \bar{c}_i)^2\right)}$$

Prvý, v S je separácia definovaná ako minimálna vzdialenosť medzi ťažiskami zhlukov. Druhý rozdiel je, že SC je súčtom jednotlivých validačných kritérií zhlukov, normalizovaným podielom kardinalít každého zhluku, čo má za následok nezávislosť SC od veľkosti zhlukov. Nízke hodnoty SC a S indikujú lepšiu konfiguráciu zhlukov vo výsledku.

3.3.6 Partition koeficient

Tento koeficient určuje mieru prekrývania medzi zhlukmi pri fuzzy zhlukovaní [36]. Je definovaný nasledovne:

$$PC(c) = \frac{1}{n} \sum_{i=1}^{n_c} \sum_{j=1}^{n} \left(u_{ij}\right)^2$$

kde u_{ij} $(i = 1,2,\dots,n_c; j = 1,2,\dots,n)$ je členstvo dátového objektu j v zhluku i. Čím bližšie je táto hodnota k jednej, tým lepšia je výsledná konfigurácia zhlukovania. Hlavná nevýhoda tohto koeficientu je monotónne klesajúca závislosť na počte zhlukov, a na rozdiel od XB indexu taktiež neexistencia priameho prepojenia s hodnotami atribútov dátových objektov.

3.3.7 Koeficient entrópie

Podobne ako XB, pracuje s hodnotami členských funkcií, ale nepracuje s objektmi ako takými, čo naznačuje, že je veľmi podobný práve PC.

Popisuje mieru neistoty v rozdelení do fuzzy zhlukov. Je definovaný nasledovne:

$$CE(c) = -\frac{1}{n}\sum_{i=1}^{n_c}\sum_{j=1}^{n} u_{ij}\log_a(u_{ij}),$$

kde a je základ logaritmu (ľubovoľný). Nadobúda hodnoty z intervalu $\langle 0, \log_a(n_c)\rangle$, pre $c > 1$ a zároveň platí $0 \leq 1 - PC(c) \leq CE(c)$. Čím je hodnota CE bližšia 0, resp. minimálna, tým ostrejšia je výsledná separácia zhlukov [21].

3.4 Zhrnutie kapitoly

V tejto kapitole som rozobral problematiku metód zhlukovacích algoritmov. Rozdelil som ich podľa vlastností generovaných zhlukov a z každej kategórie som popísal najznámejšie, resp. najrozšírenejšie z nich. Pri zlučujúcich hierarchických metódach som definoval najpoužívanejšie metódy spájania. V ďalšej časti textu v rámci nehierarchických metód som mimo najobľúbenejších metód stručne rozobral aj špeciálny typ neurónových sietí – Kohonenove mapy.

Je vhodné pripomenúť si, že každá metóda má svoje výhody a nevýhody. Pri aplikácii ľubovoľnej z nich na nejaký problém je nutné mať na pamäti, že každý problém je špecifický a vyžaduje mimo znalosti metód, pokiaľ možno, čo najlepšiu znalosť problému. Typicky – akého typu sú dáta, čo vyjadrujú, ich množstvo a samozrejme, čo chceme pomocou zhlukovej analýzy dosiahnuť. Preto je voľba vhodnej metódy pri danom probléme často neľahká úloha aj pre skúseného užívateľa.

Tretiu časť kapitoly som venoval popisu validačných kritérií. Tieto kritéria hrajú svoju úlohu v určovaní optimálneho počtu zhlukov, keďže väčšina metód, ktoré som spomenul, vyžaduje na vstupe aj informáciu o počte zhlukov. Pre tieto kritéria platia rovnaké pravidlá, ako pri voľbe zhlukovacieho algoritmu. Najvhodnejšie je sledovať viacero kritérií súčasne a optimálny počet zhlukov odhadovať ich vzájomným posúdením.

V nasledujúcej kapitole sa vrátim k problematike zhlukovania dynamicky meniacich sa dát. Neskôr sa venujem výskumu priebehu vybraných zhlukovacích a validačných metód na umelo vytvorených dynamických dátach, kde ilustrujem ich správanie.

4. Porovnanie metód zhlukovania na dynamických dátach

V nasledujúcej časti textu sa venujem problematike zhlukovania dynamicky meniacich sa dát z hľadiska konkrétnych výsledkov zhlukovacích metód. Popíšem dáta, ktoré slúžili ako podklad pre porovnanie vybraných zhlukovacích algoritmov a taktiež validačných metód pre určenie optimálneho počtu zhlukov. Priebeh aplikácie týchto metód následne ilustrujem. Je nutné dodať, že porovnávanie týchto metód je podstatné z hľadiska pochopenia a dosiahnutia stanoveného cieľa práce, ktorý taktiež spomeniem.

Presnejšie, v sekcii 4.1 rozoberám problematiku zhlukovej analýzy dynamických dát, stanovím cieľ môjho výskumu a spomeniem software, v ktorom som tieto experimenty uskutočňoval. V jej podkapitole 4.1.1 popisujem umelé dáta, s ktorými budeme pracovať. V časti 4.1.2 spomeniem zdroje, odkiaľ som použil zhlukovacie a validačné metódy.

V kapitole 4.2 určujem optimálny počet zhlukov za použitia validačných ukazovateľov. V sekcii 4.3 porovnám a ilustrujem priebeh vybraných metód a ukážem, prečo sú z hľadiska môjho cieľa nevhodné. Nakoniec zhrniem celú kapitolu v časti 4.4.

4.1 Zhluková analýza dynamických dát

Voľba zhlukovacieho algoritmu na konkrétny problém nie je úplne triviálna úloha. Typicky sa vyskúša viacero metód, a užívateľ si potom zvolí tú najvhodnejšiu. Z vlastností zhlukovacích algoritmov je zrejmé, že viacero behov algoritmu na konkrétnych dátach vráti viacero výsledkov, rôznych výsledkov, ktorých odlišnosť spočíva v rozdielnom rozdelení a umiestnení zhlukov. Výsledok sa následne vyberie na základe validačných kritérií, prípadne na základe subjektívneho hodnotenia odborníka.

Na začiatku druhej kapitoly som načrtol typ dát, ktoré obsahujú dynamicky meniace sa informácie. Zvyčajne sa jedná o sériu dát rovnakého pôvodu, štruktúry, resp. typu. V reálnom svete ich predstavujú rôzne štatistické, zdravotnícke, ekonomické, demografické, údaje o zákazníkoch alebo akékoľvek iné údaje, ktoré boli historicky s nejakou periódou zaznamenávané, a teda uchovávajú vývoj entít, ktoré sú v dátach obsiahnuté. Takéto dáta nazývame dynamicky sa meniace.

Predstavme si, že máme sériu dvadsiatich oddelených dátových množín, ktoré popisujú atribúty tých istých objektov v dvadsiatich časových okamžikoch. Obsahujú rovnaký počet dátových prvkov a rovnaký počet atribútov, resp. dimenzií. Jediný rozdiel medzi nimi je ukrytý v rozdielnych hodnotách jednotlivých atribútov dátových objektov medzi dátovými množinami. Našou úlohou je nájsť zhlukovanie na týchto dátach.

Práve tu vzniká nový problém. Po aplikácii klastrovacieho algoritmu síce nájdeme zhlukovanie na každej dátovej množine z tejto série, ale problém je v tom, že každé zhlukovanie bude rozdielne. Za normálnych okolností táto rozdielnosť vo výsledkoch nie je problém, ale keďže ide o sériu dát, je logické, že tu vzniká požiadavka na zachovanie istej súvislosti medzi nimi. Pod pojmom súvislosť môžeme myslieť napríklad sledovanie vývoja, alebo pohybu konkrétneho zhluku počas dvadsiatich sérií, čo v praxi môže znamenať vývoj určitej skupiny štátov počas dvadsiatich rokov.

V sekcii 4.3 ukážeme, že známe metódy zhlukovania skutočne neumožňujú dobre sledovať vývoj takýchto zhlukov v sérii meraní. Preto je mojím cieľom nájsť metódu, alebo metódy zhlukovania, ktorých výsledok by mi bol schopný ukázať priebeh vývoja jednotlivých zhlukov v takýchto dynamických dátach.

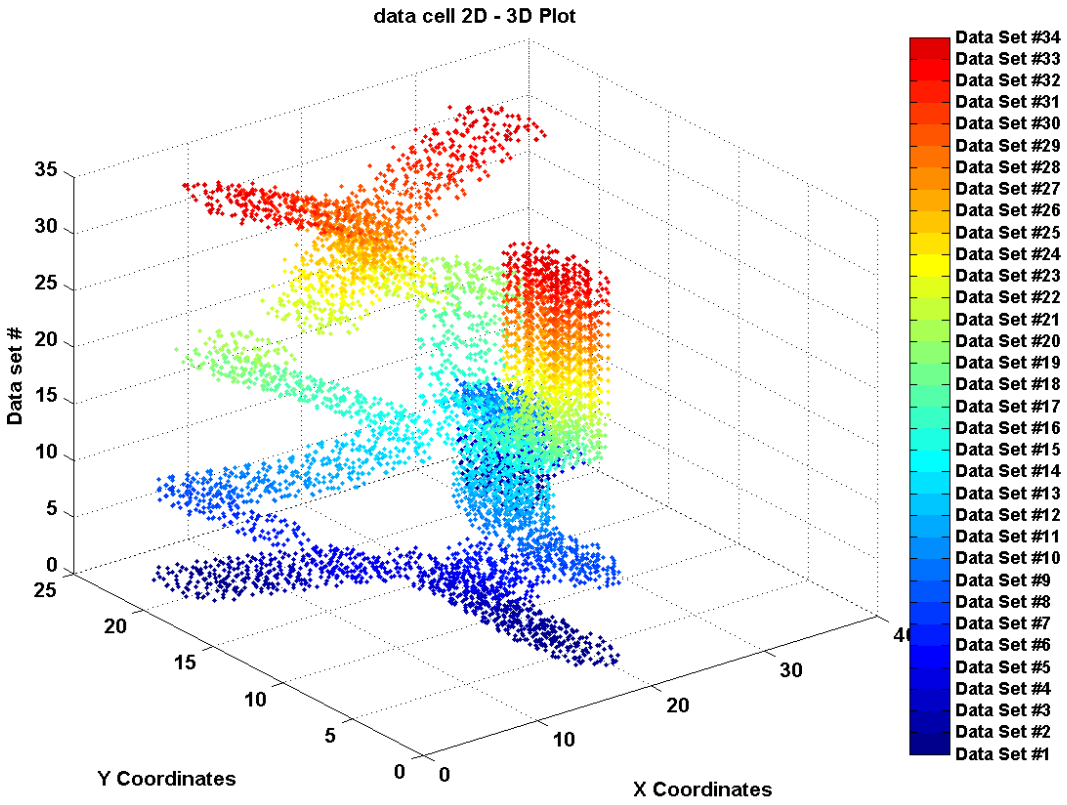
Skôr než začnem so samotným porovnávaním, chcem spomenúť, že pre potreby tejto diplomovej práce som sa rozhodol využiť software *Matlab* vo verzii $R2010b$ od spoločnosti *Mathworks*. Medzi hlavné dôvody môjho rozhodnutia patrí dostupnosť mnohých zhlukovacích a validačných metód, a taktiež široká komunita užívateľov, ktorá prispieva do produktu ďalšími implementáciami, alebo úpravami existujúcich metód, a to nie len z oblasti dátového dobývania. Mimo iného poskytuje priestor pre vhodnú vizualizáciu výsledkov. Posledný, ale rovnako dôležitý dôvod je, že každá z týchto metód je modifikovateľná, nakoľko všetky implementácie sú vo forme skriptov.

4.1.1 Umelé dáta

Pre porovnávanie metód je vhodné použiť dáta, ktorým užívateľ rozumie. Z tohto dôvodu som si vytvoril svoje umelé dáta, pomocou ktorých modelujem niekoľko zaujímavých a hraničných situácií, v ktorých sa vybrané zhlukovacie algoritmy z hľadiska tohto výskumu nesprávajú úplne korektne. Tieto dáta majú nasledujúcu štruktúru:

- 34 do série uložených dátových množín.
- Každá dátová množina obsahuje 383 dátových záznamov o 2 atribútoch (x, y).
- Každý atribút je kladné reálne číslo.

Je vhodné podotknúť, že poradie prvkov sa v rámci dátových množín nemení. Pre potreby programu *Matlab* sú na priloženom CD uložené v súbore „`three_dots_2D.mat`“. V neskoršej časti textu budem používať ešte ďalšie dáta, ale ich popis uvediem, až ich budem používať.



Obr. 12: Trojdimenzionálne zobrazenie umelých dát „`three_dots_2D.mat`“. Farby zodpovedajú príslušným dátovým množinám zo série.

4.1.2 Fuzzy Clustering and Data Analysis Toolbox

Jedným zo spomínaných výhod softwaru *Matlab* je jeho rozšíriteľnosť o nové, prípadne modifikovateľnosť existujúcich metód. Ako základ pre túto prácu som použil prídavný modul Fuzzy Clustering and Data Analysis Toolbox [37], ktorý je voľne dostupný k stiahnutiu na stránkach *Matlab FileExchange Central* [38].

Tento modul obsahuje implementácie nehierarchických zhlukovacích algoritmov *k*Means, *k*Medoids, ktorých výsledkom sú ostré zhlukovania a implementáciu Fuzzy *c*Means (ako aj jej modifikácie Gustafson-Kessel clustering a Guth-Geva clustering), ktorej výsledkom je fuzzy zhlukovanie.

Okrem zhlukovacích algoritmov obsahuje implementácie väčšiny validačných ukazovateľov, ktoré som opísal v kapitole 3.3.

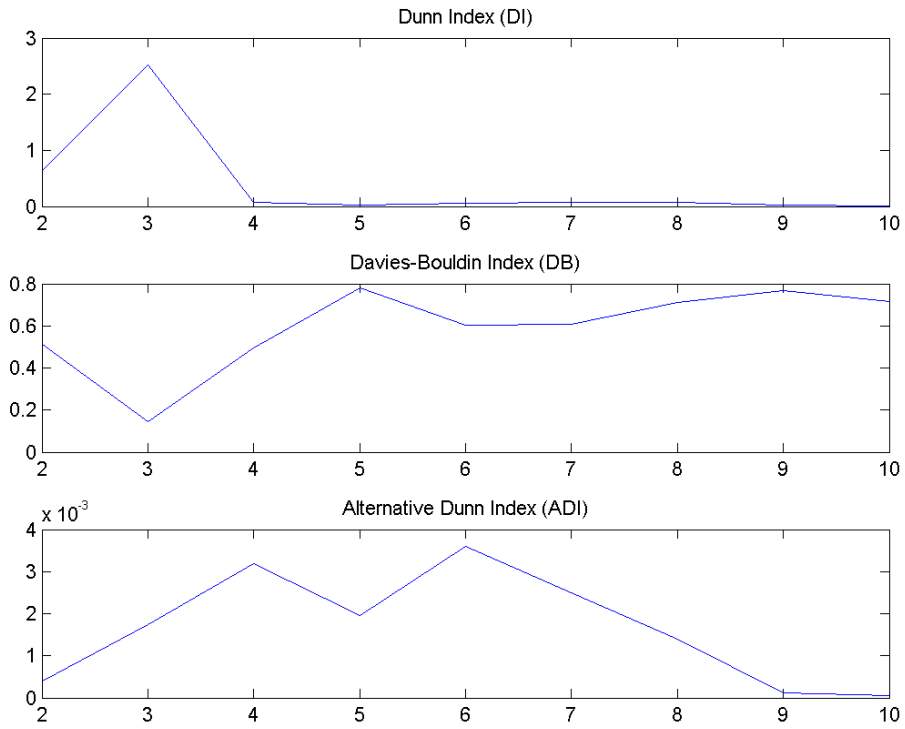
4.2 Optimálny počet zhlukov

Samotnému porovnávaniu zhlukovacích metód predchádza nutné stanovenie optimálneho počtu zhlukov na základe validačných ukazovateľov. Pre porovnanie som si zvolil zástupcu ostrých zhlukovacích metód – *k*Medoids a zástupcu fuzzy zhlukovacích metód – fuzzy *c*Means. Všetky metódy boli samozrejme testované na umelých dátach „`three_dots_2D.mat`“. Nasledujúce porovnania sú uskutočnené na prvej dátovej množine zo série. Keďže sa optimálny počet zhlukov počas celej série môže meniť, v závere tejto sekcie ukážem, ako sa toto optimum menilo a určím priemerný optimálny počet zhlukov v rámci celej série.

V prípade *k*Medoids som sledoval nasledovné validačné indexy:

- Dunnov index (DI), ktorého maximálna hodnota určuje stav pri optimálnom počte zhlukov.
- Davies-Bouldinov index (DB), v ktorom sa pri optimálnom počte zhlukov minimalizuje jeho hodnota.
- Alternatívny Dunnov index (ADI), podobne ako Dunnov Index, nadobúda maximum pri optimálnom počte zhlukov.

Výsledné hodnoty týchto validačných indexov pre počty zhlukov od 2 do 10 na prvej dátovej množine zo série „`three_dots_2D.mat`“ ilustruje obrázok 13.



Obr. 13: Hodnoty validačných ukazovateľov algoritmu *k*Medoids na prvej dátovej množine zo série „`three_dots_2D.mat`“.

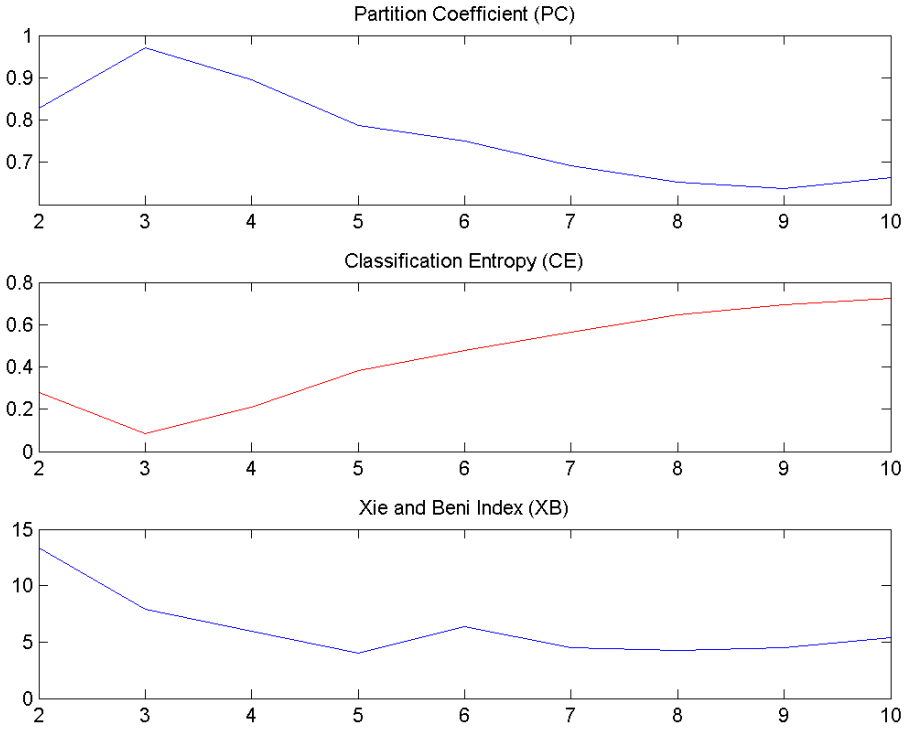
Pri posudzovaní výsledkov je nutné znovu pripomenúť, že žiaden validačný index nie je sám o sebe príliš spoľahlivý, preto je nutné porovnávať viacero ukazovateľov naraz. Z týchto výsledkov je zrejmý optimálny počet zhlukov $k = 3$. Je to vidieť najmä z hodnôt DI a DB (maximálna hodnota DI a minimálna hodnota DB).

V prípade fuzzy *c*Means algoritmu som sledoval nasledujúce validačné ukazovatele:

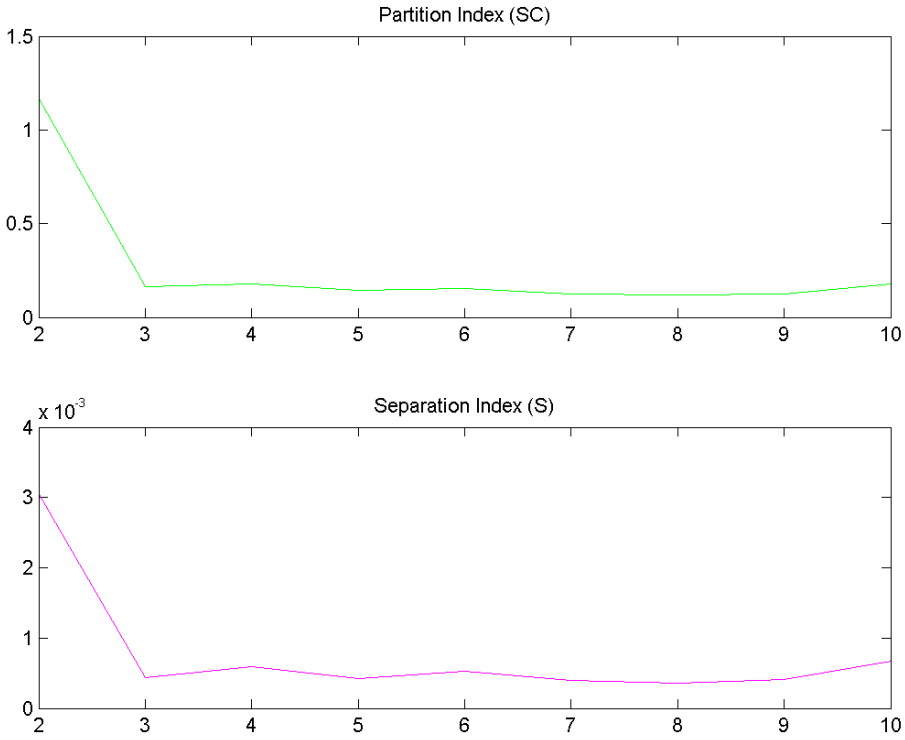
- Partition koeficient (PC). Jeho hodnota je v stave optimálneho počtu zhlukov blízko k 1.
- Koeficient entrópie (CE), ktorý v stave optimálneho počtu zhlukov má hodnotu blízku k nule.
- Partition index (SC) a Separation index (S) sú vhodné na porovnávanie výsledkov dvoch rozdielnych algoritmov. Ich nízke hodnoty značia vhodnejšiu konfiguráciu zhlukov.
- Xie-Beni ukazovateľ (XB). Nízke hodnoty XB indikujú lepšie separovanú konfiguráciu výsledného zhlukovania. Je ale monotónne klesajúci so zvyšujúcim sa počtom zhlukov.

Počas behu algoritmu boli parametre konštantné a to: $m = 2$ (fuzzyfikačný parameter), $\varepsilon = 0.001$ (presnosť algoritmu – parameter ukončenia).

Pre fuzzy *c*Means algoritmus boli výsledné hodnoty spomenutých validačných ukazovateľov nasledovné:



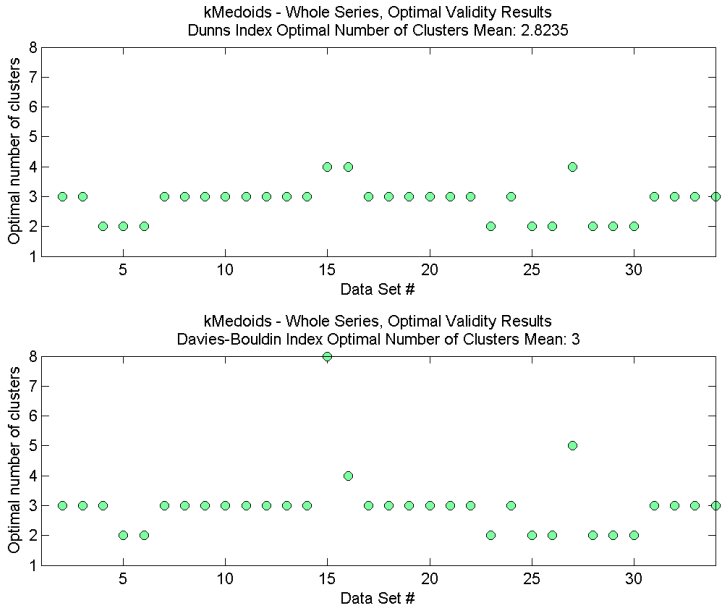
Obr. 14: Výsledné hodnoty validačných ukazovateľov PC, CE a XB pre algoritmus fuzzy *c*Means na prvej dátovej množine zo série „`three_dots_2D.mat`“.



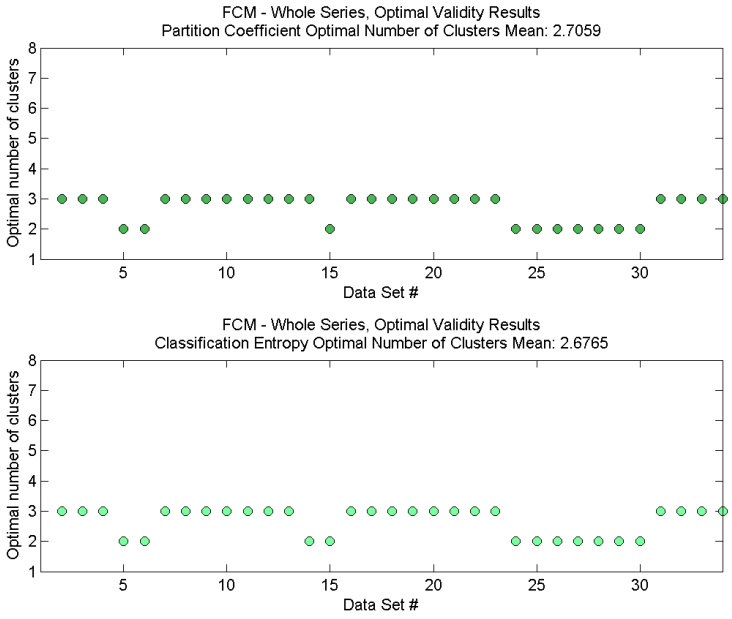
Obr. 15: Výsledné hodnoty validačných ukazovateľov SC a S pre algoritmus fuzzy *c*Means na prvej dátovej množine zo série „`three_dots_2D.mat`“.

PC ukazovateľ klesá s rastúcim počtom zhlukov a CE index stúpa s rastúcim počtom zhlukov, preto sa dajú posudzovať len do počtu, kedy začína táto monotónnosť. V oboch prípadoch je to $k = 3$. Hodnoty SC a S rapídne klesajú v bode $k = 3$. Jediný index, ktorý nepotvrdzuje výsledok, je XB, ktorý nadobúda lokálne minimum v bode $k = 5$.

Ako som spomínal na začiatku tejto časti textu, nájdené optimum nemusí byť ideálne počas celého behu algoritmu v rámci série dát, pretože môžu nastať situácie, kedy budú hodnoty validačných ukazovateľov indikovať úplne inú situáciu. Z toho dôvodu som určil priemernú hodnotu optimálneho počtu zhlukov z celej dátovej série. V obrázkoch 16 a 17 sú ilustrované výsledky, ktoré som dosiahol pri postupnom aplikovaní určovania optimálneho počtu zhlukov na jednotlivé dátové množiny zo série. Ich priemerom som sa opäť dopracoval k hodnote $k = 3$. Dôvodom počítania priemeru je, že chceme sledovať, ako sa zhluky v čase vyvíjajú, resp. zväčšujú, zmenšujú a pohybujú. Ak by sa počas zhlukovania spojili alebo rozdelili, mohli by sme ich natrvalo stratiť, čo nechceme.



Obr. 16: Optimálne počty zhlukov pre jednotlivé dátové množiny zo série „`three_dots_2D.mat`“ – algoritmus *k*Medoids.



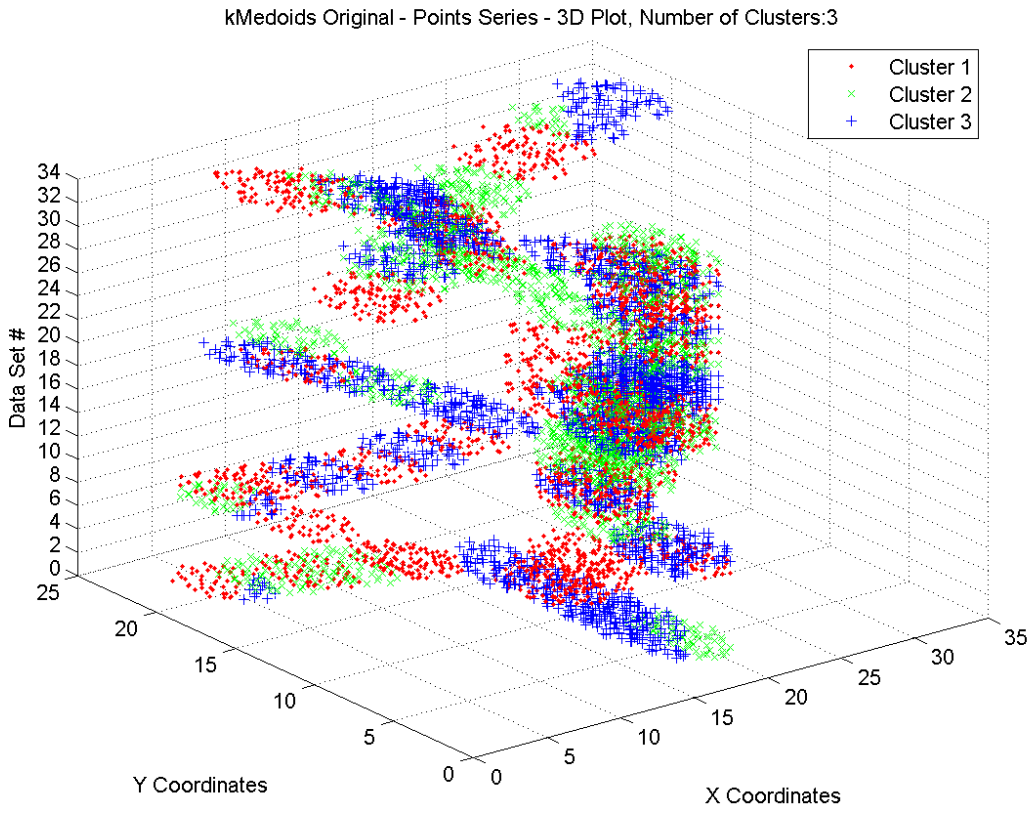
Obr. 17: Optimálne počty zhlukov pre jednotlivé dátové množiny zo série „`three_dots_2D.mat`“ – algoritmus FCM.

Na základe týchto výsledkov som určil optimálny počet zhlukov pri použití algoritmu *k*Medoids a fuzzy *c*Means na umelých dvojrozmerných dátach na $k = 3$, čo je očakávaný výsledok.

4.3 Experimenty s dátami

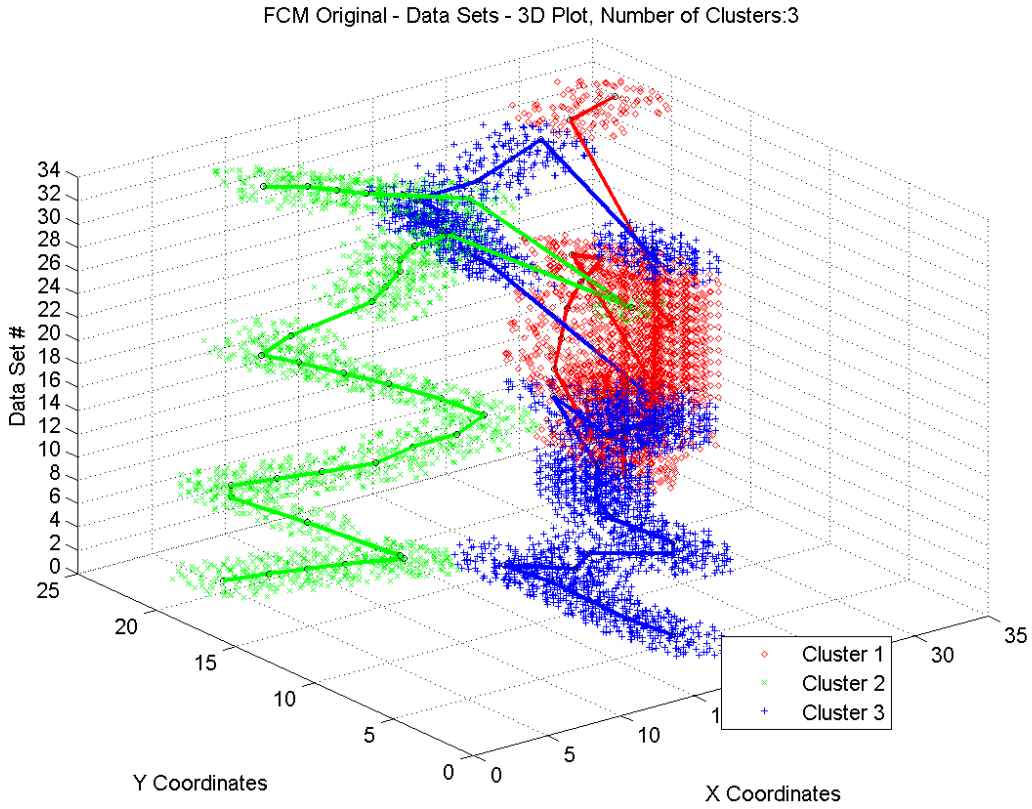
Problém zachovania súvislosti vo výsledkoch zhlukovania na sérii dát, ktorý som načrtol na začiatku tejto kapitoly, sa dá jednoducho ilustrovať.

Nasledujúci graf (obr. 18) je výsledkom behu algoritmu *k*Medoids na umelých dátach „`three_dots_2D.mat`“, postupne od prvej dátovej množiny po poslednú, resp. tridsiatu štvrtú v poradí. Parameter $k = 3$ je konštantný.



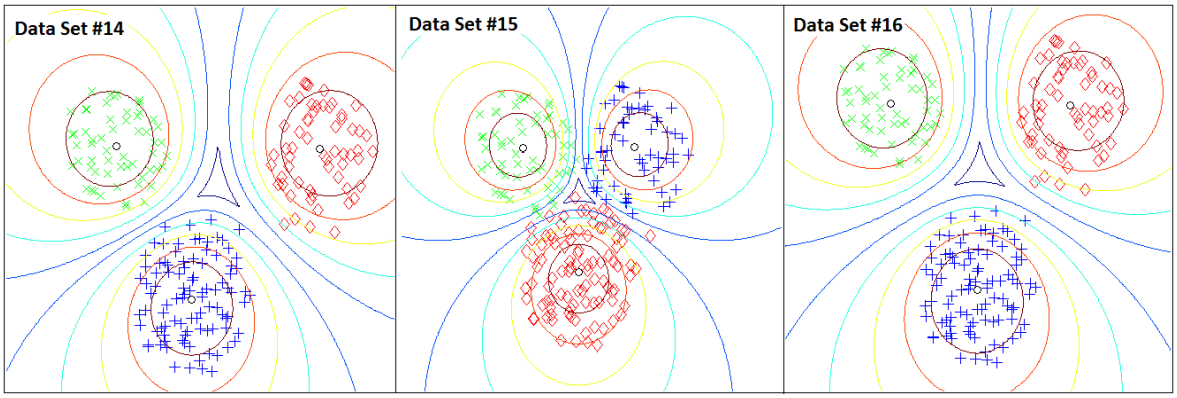
Obr. 18: Zobrazenie výsledkov zhlukovania na dátových množinách tvoriacich sériu dynamických dát. Náhodnosť rozhádzania zhlukov zviditeľňuje neexistenciu previazanosti jednotlivých zhlukovaní.

Na obrázku 18 je pekne vidieť, prečo nie sú klasické postupy, resp. aplikácie zhlukovacích algoritmov vhodné pre riešenie zhlukovej analýzy dynamických dát. Jednotlivé výsledky zhlukovaní dátových množín nie sú vôbec konzistentné, algoritmus tieto dátové množiny zo série prirodzene berie ako nezávislé úlohy, konkrétne tridsať štyri nezávislých zhlukovacích úloh a je logické, že takto neposkytuje viacmenej žiadne možnosti sledovania vývoja dát a ich zhlukov v čase. Pre porovnanie som vyskúšal algoritmus fuzzy *c*Means. Parametre algoritmu boli rovnaké po celú dobu behu, na každej dátovej množine, a to $m = 2$, $\varepsilon = 0.001$ a $c = 3$. Na obrázku 19 je vizualizácia jeho priebehu, ktorý postačuje na to, aby sme potvrdili fakt, že ani metóda na báze fuzzy logiky nepostačuje v takejto podobe na sledovanie vývoja dát, alebo skôr vývoja ich zhlukov.

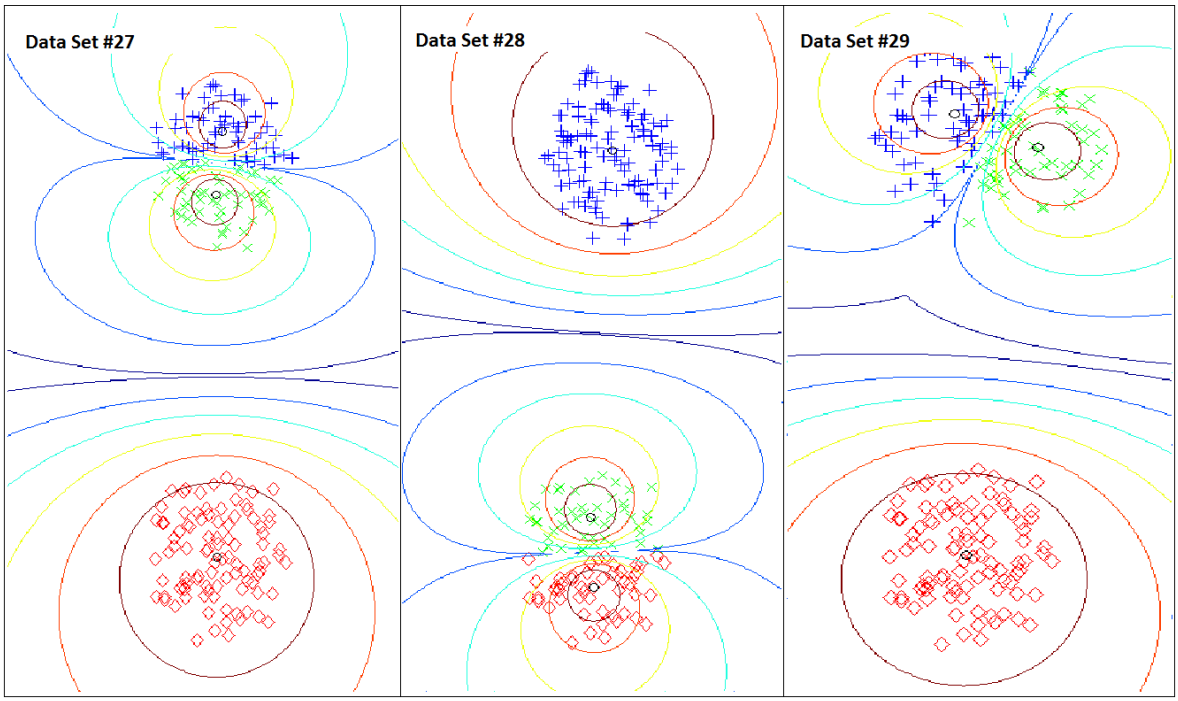


Obr. 19: Graf priebehu FCM metódy na dátových množinách zo série „`three_dots_2D.mat`“. Farebné čiary predstavujú pohyb ťažísk zhlukov.

Pri pohľade na obrázok 19 je vidno, že inicializačná fáza metódy FCM nie je úplne náhodná, na rozdiel od metódy *k*Medoids. Je to vidno najmä na zelenom zhluku. Napriek tomu je táto metóda nevhodná na sledovanie vývoja zhlukov v sérii dát, čo usudzujem podľa nevhodného rozloženia zhlukov zhruba od 16. dátovej množiny, kde sa v dátach zhluky postupne fyzicky približujú ku sebe, až na dotyk a to spôsobuje výmeny červeného a modrého zhluku, a neskôr dokonca aj zeleného zhluku. Pre lepšiu predstavu tu uvádzam aj niekoľko prierezov grafom podľa osy *z* (obr. 20 a obr. 21).



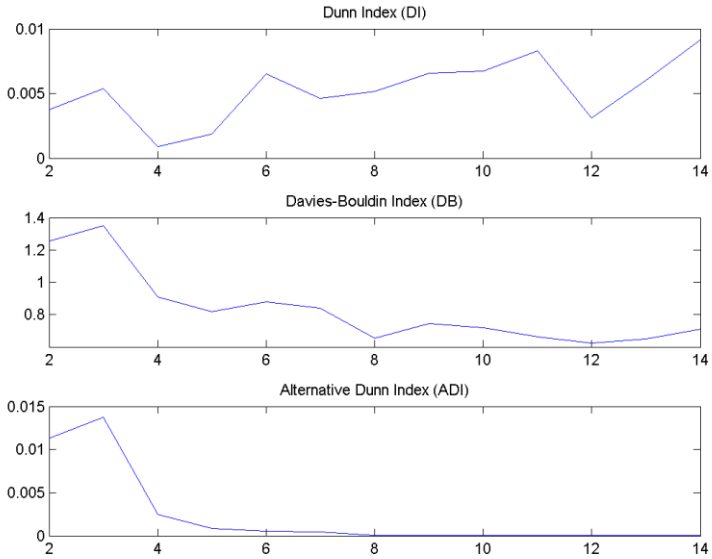
Obr. 20: Príklad nechcenej výmeny priradenia zhlukov počas FCM zhlukovania, keď sa zhluky k sebe priblížia.



Obr. 21: Príklad nechcenej výmeny a umiestnenia zhlukov počas priebehu FCM zhlukovania.

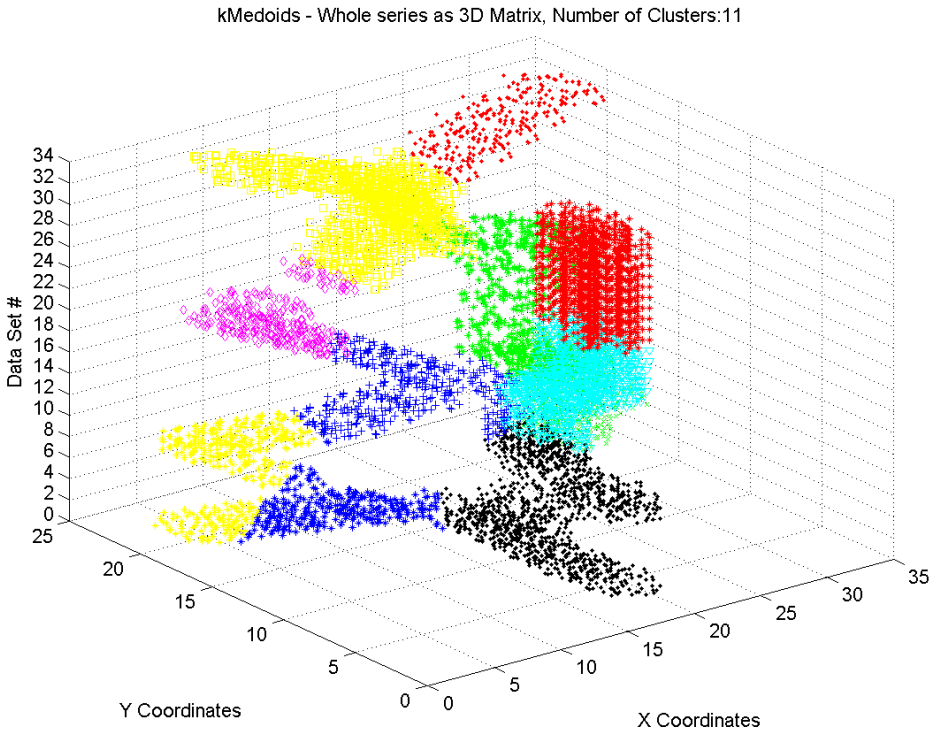
Výsledok experimentu s metódou *k*Medoids bol neuspokojivý. Metóda FCM na mojich dátach preukázala lepšie vlastnosti v inicializácii ťažísk zhlukov, no v závere sa preukázala taktiež ako nevhodná na sledovanie vývoja zhlukov v rámci sériových, dynamických dát.

Pri pohľade na trojdimenzionálnu vizualizáciu mi napadlo uskutočniť zhlukovanie dátovej série ako celku, t.j. dát uložených do trojdimenzionálneho poľa. Zvolil som klastrovací algoritmus kMedoids. Predtým než som spustil samotné zhlukovanie, určil som pomocou validačných ukazovateľov optimálny počet zhlukov $k = 11$.



Obr. 22: Hodnoty validačných indexov pri zhlukovaní *k*Medoids nad trojrozmernou maticou vytvorenej zo série „`three_dots_2D.mat`“.

Zhlukovanie metódou *k*Medoids nad trojdimenzionálnou maticou obsahujúcou celú sériu dát, podáva lepšie výsledky, než jej samostatná aplikácia na oddelené dátové množiny zo série. Každopádne, z hľadiska sledovania vývoja zhlukov od začiatku až do konca série to nie je úplne vhodné, pretože viacero rôznych zhlukov je v skutočnosti jeden zhluk, ktorý menil svoje označenie počas behu, čo je vidno na nižšie uvedenom obrázku (obr. 23).



Obr. 23: Vizualizácia priebehu algoritmu *k*Medoids na trojdimenzionálnej matici vytvorenej zo série „`three_dots_2D.mat`“. Spomenuté nedostatky je vidno napríklad na preznačení zhluku označeného tyrkysovou farbou na zhluk označeným červeným krížikom.

4.4 Zhrnutie kapitoly

V tejto časti textu som sa venoval výskumu priebehu nemodifikovaných, dostupných metód na umelo vytvorených dátach, ktoré som na začiatku kapitoly stručne opísal. Taktiež som uviedol dôvody rozhodnutia pri výbere platformy, v ktorej som všetky výskumy vykonával a zdroje metód, ktoré som používal.

Výsledky experimentov hovoria, že z hľadiska môjho cieľa sú tieto vyskúšané postupy nevhodné, niektoré menej, niektoré viac, ale svojím spôsobom sú pozitívne, pretože som získal cenné informácie, ktoré mi pomohli vytvoriť modifikácie metód, ktorých výsledky sú oveľa vhodnejšie. Práve týmito modifikáciami sa budem zaoberať v ďalšej kapitole.

5. Modifikované metódy

V predošlej kapitole sme sa dozvedeli, že aplikáciou klasických zhlukovacích metód na dynamické dáta nedostaneme výsledky, ktoré by nám pomohli sledovať vývoj ich zhlukov v čase. Po zmene perspektívy sme zistili (séria dát ako trojrozmerná matica), že výsledky sú síce lepšie, ale stále sú nedostatočné. Informácie, ktoré som z týchto experimentov získal, mi pomohli navrhnúť postupy, ktorých výsledky sú z hľadiska stanoveného cieľa omnoho prijateľnejšie. Tieto postupy rozoberám v nasledujúcej časti textu, kde popisujem konkrétne modifikácie a ilustrujem viacero zaujímavých situácií.

Konkrétne, v podkapitole 5.1 vysvetlím princíp navrhnutých modifikácií metód. Tieto modifikované metódy, resp. modifikovanú metódu *k*Medoids a jej priebeh na dvojrozmerných umelých dátach rozoberiem v sekcii 5.2 a podobným spôsobom rozoberiem aj modifikovanú metódu Fuzzy *c*Means v časti 5.3. Podkapitola 5.4 vysvetľuje, akým spôsobom som vytvoril multidimenzionálne umelé dáta, ktoré slúžili ako podklad pre experimenty s modifikovanými metódami na dátach o viacerých rozmerov, než dvoch. V sekcii 5.5 vysvetlím, ako som hodnotil priebeh týchto metód a v podkapitolách 5.6 a 5.7 klasifikujem a porovnám priebeh konkrétnych modifikácií *k*Medoids a FCM na viacrozmerných dátach. Zhrnutie celej kapitoly sa nachádza v 5.8.

.

5.1 Princíp

Ako sme videli v predošlej kapitole, najväčším problémom aplikácie klasického algoritmu na sériu dát je absencia súvislosti medzi jednotlivými výsledkami, alebo nežiaduca výmena priradenia zhlukov, ak sa tieto zhluky k sebe priblížili, prípadne okolo seba prešli. Tomu sa však dá predísť.

Myšlienkou je zviazať samostatné zhlukovania na jednotlivých dátových množinách zo série – šikovne prepojiť inicializáciu nasledujúceho behu algoritmu na ďalšej množine dát s výsledkom predošlého behu algoritmu na predošlej časti dát zo série.

To budem robiť v dvoch krokoch:

1) Inicializácia počiatočných ťažísk zhlukov v rámci dátových objektov zo zhlukov z predošlého výsledku,
2) Úprava metriky, na základe ktorej je posudzovaná vzdialeností objektov.

Podrobnejšie.

Inicializácia prvého behu zhlukovacieho algoritmu prebehne bez modifikácie. V druhom behu, teda na druhej množine dát zo série je ale situácia nasledovná:

- Na vstupe algoritmu mimo dát a parametru k dostáva výsledné priradenia dátových objektov k zhlukom z predošlého behu algoritmu.
- Počiatočné ťažiská zhlukov vyberá náhodne v rámci zhlukov z predošlého zhlukovania, teda pre zhluk $i = 1, \dots, k$ vyberie za ťažisko náhodný dátový objekt zo zhluku i z výsledku predošlého behu algoritmu.

Tým sa zabezpečí konzistencia v rozdelení a umiestnení zhlukov počas behu algoritmu po celej sérii dát.

Druhý krok modifikácie spočíva v úprave posudzovania vzdialeností objektov. V každej iterácii zhlukovacieho algoritmu sa počíta matica vzdialeností $dist$ v tvare $N \times k$. $dist_{ij}$ vyjadruje vzdialenosť dátového objektu i od zhluku j, resp. jeho ťažiska. V tomto kroku modifikujem túto vzdialenostnú maticu podľa nasledujúceho kritéria:

$$dist_{i,j}^{(t)} = \begin{cases} dist_{i,j}^{(t)} * \alpha, & pre\ i \in c_j^{(t-1)} \\ dist_{i,j}^{(t)} * \frac{1}{\alpha}, & pre\ i \notin c_j^{(t-1)} \end{cases}$$

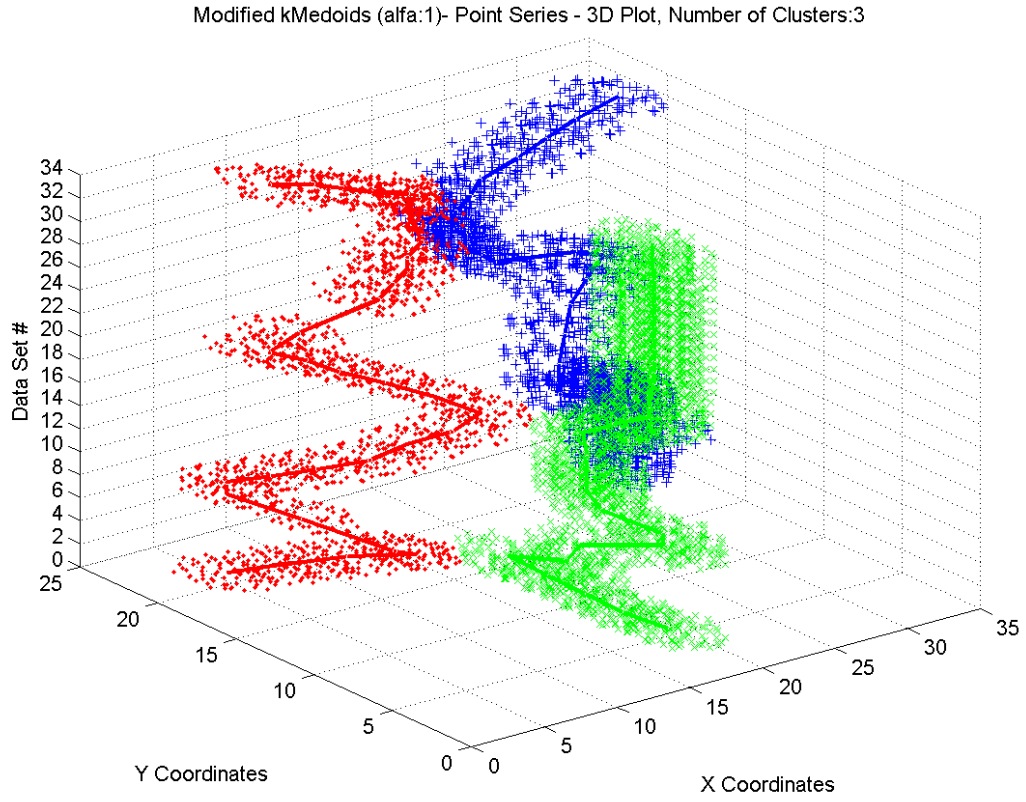
- $\alpha \in \langle 0.1,1 \rangle$ je modifikačný parameter, kde sa pre špeciálny prípad $\alpha = 1$ vzdialenosti nemenia,
- t je označenie časového okamžiku (dátovej množiny) v sérii dát.

Inými slovami, posilňujem väzbu, teda zmenšujem vzdialenosti medzi bodmi a zhlukmi, ktoré k sebe patrili vo výsledku predošlého zhlukovania.

Je potrebné dodať, že táto modifikácia je vhodná pre metódu *k*Means a *k*Medoids. Pre metódu FCM je úprava analogická a k jej popisu sa dostanem neskôr v texte. Hodnota parametru α v skutočnosti nie je nijak obmedzená, je možné vytvoriť aj opačný efekt, napríklad hodnotou väčšou než 1.

5.2 Modifikácia *k*Medoids pre 2D dáta

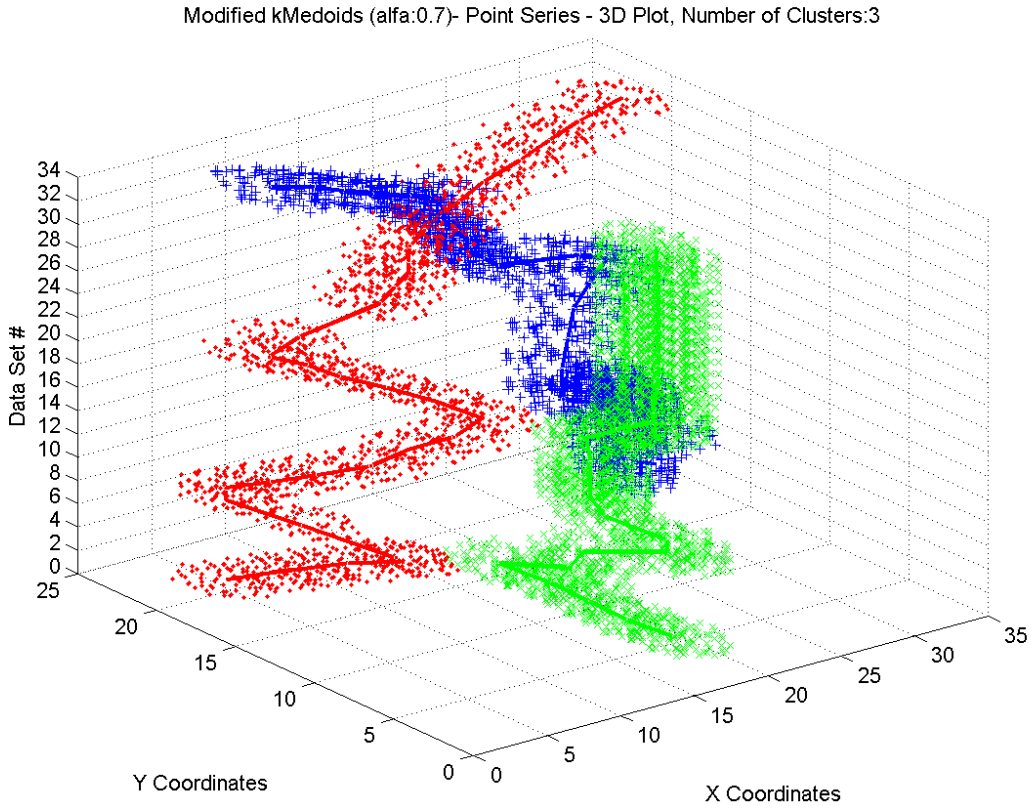
Pre porovnanie vplyvu parametru α na výsledok zhlukovania prezentujem (obr. 24) najprv priebeh algoritmu s hodnotou $\alpha = 1$, čo znamená, že nijako neupravujem maticu vzdialeností, a teda modifikujem len proces výberu počiatočných ťažísk zhlukov.



Obr. 24: Priebeh modifikovanej metódy *k*Medoids s parametrom $\alpha = 1$ na dátových množinách zo série „three_dots_2D.mat“.

Priebeh zhlukovania je očakávaný. Hrubé farebné čiary v rámci zhlukov značia pohyb ich ťažísk. Modifikovaný výber počiatočných zhlukov zabezpečuje súvislosť medzi výsledkami v rámci jednotlivých zhlukovaní dátových množín zo série. Hodnota parametru $\alpha = 1$ nezabezpečí žiadne posilnenie väzieb medzi bodmi a zhlukmi, čo je vidno na pohybe modrého a červeného zhluku. V dátach si v skutočnosti tieto dva zhluky vymenia svoje pozície. To v prípade $\alpha = 1$ nastane, jedine ak náhodný výber počiatočných ťažísk spomedzi zhlukov zvolí v okamžiku ich stretnutia „správne“ objekty (čo najvzdialenejšie medzi červeným a modrým zhlukom).

Postupným znižovaním parametru α sa dostaneme až do stavu, keď naopak, v drvivej väčšine behov algoritmu dosiahneme požadovaný stav, t.j. spomínané zhluky cez seba prejdú. Hodnota toho parametru je $\alpha = 0.7$. Požadovaný stav vyzerá nasledovne (obr. 25).



Obr. 25: Obrázok priebehu modifikovanej metódy *k*Medoids s parametrom $\alpha = 0.7$ na dátových množinách zo série „`three_dots_2D.mat`“.

V skutočnosti veľkosť parametru $\alpha = 0.7$ poskytuje takmer vždy požadované riešenie. Z 10 pokusov bol výsledok pozitívny v 10 prípadoch, ale nie je vylúčené, že pri 100 pokusoch by výsledok nebol aspoň v jednom prípade požadovaný. Parameter $\alpha = 0.8$ skončí približne v 66% z prípadov v nevyhovujúcom riešení (zhluky cez seba neprejdú). Pri veľkosti parametra $\alpha = 0.75$ je už viac ako 75% výsledkov korektných, teda aspoň z hľadiska môjho cieľa. Dodajme, že stav požadovaného riešenia je subjektívna požiadavka, ktorá na inak namodelovaných dátach môže byť iná. Nízke hodnoty parametru α zabezpečujú vysoký stupeň súdržnosti bodov patriacim k rovnakému zhluku, a ako uvidíme neskôr v tejto práci, tento postup funguje aj na iných vstupných dátach.

5.3 Modifikácia fuzzy *c*Means pre 2D dáta

Metódu FCM je možné analogicky, s malými úpravami modifikovať. Pri pokusoch s touto metódou som prišiel k záveru, že ju nie je nutné upravovať v dvoch krokoch. Konkrétnejšie, úprava matice vzdialeností parametrom α na výsledky vplýva minimálne, resp. viditeľne na ne vplýva až vo veľmi nízkych hodnotách α. Keďže postup je v porovnaní s modifikáciou v metóde *k*Medoids kúsok rozdielny, rozoberiem ho podrobnejšie.

Úprava pozostáva opäť z dvoch krokov. Počas úpravy matice vzdialeností postupujeme ako pri upravenej metóde *k*Medoids s tým rozdielom, že posilňujem väzbu medzi bodmi a zhlukmi, ktoré majú z predošlého zhlukovania najvyššiu hodnotu členskej funkcie, a nie hodnotu 1, pretože FCM nie je ostrá zhlukovacia metóda a takúto hodnotu členská funkcia za bežných okolnosti nenadobúda.

Ďalší, ale podstatnejší rozdiel je v prvom kroku, teda vo fáze výberu počiatočných ťažísk. Definujem si tieto dve hodnoty:

- $max\_partnership(j) = max\{u_{jc}\}, \forall c \in 1, \dots, n_c$, kde u_{jc} je hodnota členstva objektu j v zhluku c

- $min\_partnership(j) = min\{u_{jc}\}, \forall c \in 1, \dots, n_c$, kde u_{jc} je hodnota členstva objektu j v zhluku c

Inicializácia prvého behu zhlukovacieho algoritmu prebehne znovu bez modifikácie. V druhom behu, teda na druhej množine dát zo série je ale situácia nasledovná:

- Na vstupe algoritmu mimo dát a počtu zhlukov dostáva výslednú maticu hodnôt členských funkcií pre všetky objekty a zhluky z predošlého behu algoritmu.

- Počiatočné ťažiská zhlukov vyberá náhodne v rámci zhlukov z predošlého zhlukovania, ale len z objektov, ktoré boli relatívne blízko k ťažisku.

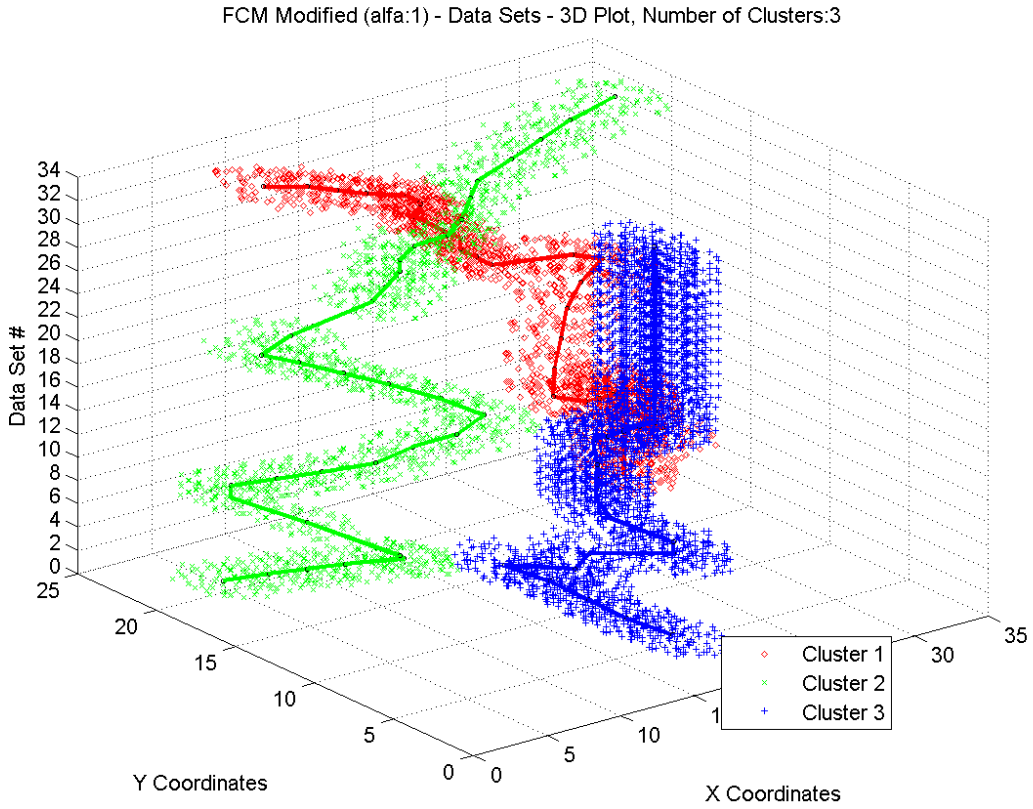
  Objekt j bol blízko k ťažisku i, ak:
  $u_{ji} \geq min\_partnership(j) + \big(max\_partnership(j) - min\_partnership(j)\big) * \beta$

  Objekt j nebol blízko k ťažisku i, ak:
  $u_{ji} < min\_partnership(j) + \big(max\_partnership(j) - min\_partnership(j)\big) * \beta$

  β je parameter, ktorý posúva hranicu, ktorá rozdeľuje body na blízke a vzdialenejšie od ťažiska zhluku. Ak $\beta = 1$, blízke body sú len tie, ktoré majú hodnotu príslušnosti k danému zhluku maximálnu, v najhoršom prípade teda len jeden objekt. Čím je hodnota β bližšia k nule, tým je táto hranica voľnejšia – smerom k bodom s najnižšou príslušnosťou k danému zhluku. Fakt, že pracujem v intervale od minimálnych po maximálne hodnoty príslušnosti k zhluku zaručuje, že vždy vyberiem minimálne jeden objekt z takého zhluku, t.j. zhluk nezanikne.

1. Pre zhluk $i = 1, \dots, n_c$ vyberie za ťažisko dátový objekt z objektov blízkych k ťažisku zhluku i z výsledku predošlého behu algoritmu.

Nasleduje priebeh modifikovaného FCM algoritmu pri $\alpha = 1$ a $\beta = 0.75$.



Obr. 26: Vizualizácia priebehu modifikovaného algoritmu FCM s $\alpha = 1$ a $\beta = 0.75$ na dátových množinách zo série „`three_dots_2D.mat`“. Kreslenie kontúr je z dôvodu zachovania prehľadnosti vynechané.

Na obrázku 26 je pekne vidno, že algoritmus podáva požadovaný výsledok. Červený a zelený zhluk si z hľadiska nášho cieľa korektne vymenia svoje pozície.

Na základe týchto výsledkov je možné povedať, že mnou modifikované metódy poskytujú na týchto dvojrozmerných vytvorených dátach priaznivé výsledky, inými slovami, sú vhodné na sledovanie vývoja zhlukov v rámci nich. Metóda FCM sa javí ako stabilnejšia a vhodnejšia už pri minimálnej, v tomto prípade dokonca žiadnej zmene hodnoty parametra α.

Keďže cieľom tejto práce je nájsť vhodnú metódu, resp. metódy a aplikovať ich na reálne dáta, môžeme predpokladať, že takéto dáta nebudú takéto jednoduché, hlavne čo sa týka počtu dimenzií. Z toho dôvodu v druhej časti tejto kapitoly preskúmam ich priebeh na viacrozmerných dátach.

5.4 Umelé multidimenzionálne dáta

Ako som už spomínal, dvojdimenzionálne dáta „`three_dots_2D.mat`“ nie sú jediné, ktoré som počas výskumu v tejto práci potreboval. Predošlé dáta slúžili ako podklad pre porovnávanie priebehu dostupných zhlukovacích metód a ich modifikácii v dvojrozmernom priestore. Iste, výsledky boli ilustrované v trojrozmernom priestore, ale v skutočnosti išlo o tridsať štyri dvojrozmerných projekcií výsledkov v rámci behu algoritmu uložených v priestore. Takto zobrazené výsledky poskytujú dobrý prehľad o postupnom vývoji zhlukov nájdených v dátach.

S ohľadom na cieľ tejto práce je nutné preskúmať priebeh týchto modifikácií na dátach, ktoré sú viac, než dvojrozmerné. Dôvodom je, že ak mám aplikovať metódu na reálnych dátach, tak môžem predpokladať, že nebudú dvojrozmerného charakteru. Dáta historického charakteru, ktoré uchovávajú vývoj nejakých entít, budú istotne viacrozmerné. Mimo iného, je možné predpokladať, že takéto dáta sú vždy istým spôsobom medzi sebou závislé, napr. hodnota hrubého domáceho produktu ovplyvňuje mnoho ďalších dôležitých ukazovateľov štátu. Hodnoty taktiež závisia aj od svojej histórie. Tieto úvahy viedli k tomu, že som nevytváral len úplne náhodné dáta, ale aj z časti náhodné, teda pseudonáhodné. Z aktuálnych dvojrozmerných dát som vytvoril štvorrozmerné, zo štvorrozmerných šesťrozmerné, a takto som postupoval, až som vytvoril desaťrozmerné dáta.

Pseudonáhodné dáta

V každom kroku som k dátovej množine zo série prilepil dve dimenzie. Postup bol nasledovný pre každú dátovú množinu zo série:

1. Pôvodné dvojdimenzionálne dáta danej dátovej množiny som rozdelil do troch skupín, vizuálne do troch zhlukov bodov.
2. K dátam z každého zhluku som prilepil náhodne dve dimenzie. Tieto pridané dimenzie boli náhodné hodnoty objektov z daných zhlukov.

Takto som pseudonáhodne vytvoril dáta o štyroch, šiestich, ôsmych a desiatich dimenziách. Je jasné, že väčšine ľudí kladie predstavivosť v päť a viac dimenziách hranice, preto sú tieto dáta tvorené práve rozmermi, ktoré sú si podobné. Získavame z toho možnosť vizualizácie práve v rámci iných dimenzií, no stále pri projekcii riešení do trojrozmerného priestoru, čo výrazne udržiava prehľadnosť.

Náhodné dáta

Oproti pseudonáhodným dátam, kde som vyberal hodnoty z predurčených skupín, náhodné dáta vznikli náhodným výberom hodnôt z celej dátovej množiny.

Tieto náhodné dáta som vytvoril z dôvodu, že síce predpokladám, že reálne dáta sú aspoň v nejakej miere závislé, no predošlým spôsobom vytvorené dáta sú svojím charakterom v skutočnosti vysoko závislé. Na takto náhodných dátach je síce ťažké hľadať nejaké zhluky, pretože tam v skutočnosti žiadne neexistujú, ale pre porovnanie je to vhodné vyskúšať.

S pribúdajúcim počtom dimenzií klesá nielen predstavivosť, ale taktiež to má vo všeobecnosti negatívny vplyv na posudzovanie vzdialeností pomocou metrík. To má za následok skresľovanie výsledkov. V algoritmoch, ktoré som modifikoval, je použitá Euklidovská metrika. Táto metrika je spoľahlivá za podmienky, že hodnoty atribútov v každej dimenzii sú rovnakej mierky. Spôsobom, akým som vytvoril tieto viacrozmerné dáta, som bez potreby normalizácie dát túto podmienku dodržal a je teda možné predpokladať, že bude podávať zmysluplné hodnoty aj vo viac, než dvoch dimenziách.

5.5 Klasifikácia priebehu modifikovaných metód na viacrozmerných dátach

Mnohorozmernosť spôsobuje nemalé problémy a prekážky vo vizualizácii výsledkov priebehu zhlukovacích algoritmov. Z toho dôvodu navrhneme spôsob, akým ohodnotíme vhodnosť použitia, resp. priebeh nami modifikovaných metód na rôznych viacrozmerných dátach, ktoré sme si pripravili. Ako som spomínal v kapitole 5.1, najväčším problémom aplikácie klasického algoritmu na sériu dát je absencia súvislosti medzi jednotlivými výsledkami, alebo nežiaduca výmena priradenia zhlukov v krajných situáciách. Z toho dôvodu je myšlienka nami navrhnutej klasifikácie sledovanie zmien v priradení zhlukov.

Presnejšie, hodnota čísla, ktoré indikuje vhodnosť použitej konfigurácie metódy na viacrozmerných dátach, je počet situácií, kedy v rámci zhlukovania zmenil ktorýkoľvek objekt príslušnosť k zhluku. Táto metóda je aplikovateľná na obidve nami modifikované metódy *k*Medoids a FCM, no v prípade FCM je nutné spomenúť, že zhluk, ku ktorému objekt patrí, je určený najvyššou hodnotou členskej funkcie daného objektu.

V ďalšej časti textu analyzujeme výsledky aplikácie modifikovanej metódy *k*Medoids na rôznych viacrozmerných dátach. Neskôr, v podkapitole 5.7 analogicky rozoberieme priebeh modifikovaného algoritmu FCM. V závere zhrnieme a zhodnotíme všetky poznatky získané v rámci piatej kapitoly tejto práce.

5.6 Modifikácia *k*Medoids pre viacrozmerné dáta

Pre zachovanie prehľadnosti sú výsledky priebehu rôznych konfigurácií modifikovanej metódy *k*Medoids a vstupných dát prezentované formou tabuliek.

Dáta / α	1	0.9	0.8
4D pseudonáhodné	498	425	169
6D pseudonáhodné	578	276	162
8D pseudonáhodné	525	263	115
10D pseudonáhodné	579	309	111

Modifikovaný *k*Medoids algoritmus – výsledné ohodnotenia priebehu algoritmu na pseudonáhodných umelých dátach. Hodnoty vyjadrujú počty zmien priradení objektov k zhlukom. Parameter $k = 3$.

Pri pohľade na výsledné hodnoty hlavne v prípade $\alpha = 1$ je vidno, že narastajúci počet dimenzií spôsobuje vyšší počet zmien priradení. Je taktiež vidno, že frekvencia zmien klesá s klesajúcim parametrom α, čo je správne. Čo sa ale na prvý pohľad môže zdať ako nesprávne je, že v prípade $\alpha = 0.9$ a $\alpha = 0.8$ majú počty priradení tendenciu klesať so stúpajúcim počtom dimenzií. Pri hlbšom zamyslení zistíme, že to je očakávaný jav. Tieto dáta v skutočnosti nie sú náhodné, sú náhodné len z časti. Dá sa dokonca povedať, že sú závislé, a to v celkom vysokej miere. Dôvodom je fakt, že v každých dvoch prídavných rozmeroch k objektu pridávam hodnoty z jeho zhluku. To spôsobuje, že dáta k sebe patria ešte bližšie a tie vzdialenejšie sa počas posudzovania vzdialeností vzdialia ešte viac, pretože obsahujú síce náhodné, ale hodnoty práve zo vzdialených objektov. Každými dvoma prídavnými dimenziami sú dáta viac závislé, vzdialenosti sú väčšie, a preto nie je nutný až taký nízky parameter α pre minimalizáciu frekvencie výmen príslušností. Práve preto sme v experimentoch použili aj dáta, ktoré sú menej závislé.

Dáta / α	1	0.9	0.8	0.7
4D náhodné	5608	3817	983	596
6D náhodné	6955	3948	912	523
8D náhodné	7425	4045	747	325
10D náhodné	7460	3782	489	232

Modifikovaný *k*Medoids algoritmus – výsledné ohodnotenia priebehu algoritmu na náhodných umelých dátach. Hodnoty vyjadrujú počty zmien priradení objektov k zhlukom. Parameter $k = 3$.

Je veľmi dobre vidno, že náhodným výberom vytvorené dáta spôsobujú obrovský nárast v počte zmien priradení, takisto aj s rastúcim počtom dimenzií, pokiaľ nie je výpočet vzdialenostnej matice žiadnym spôsobom ovplyvnený, t.j. $\alpha = 1$. To je spôsobené tým, že hľadanie zhlukov v tak náhodných dátach je veľmi ťažké, resp. nemožné. Každopádne aj na takýchto dátach algoritmus určí pomyselné zhluky a posilňovanie vzdialeností prislúchajúcich bodov a zhlukov znižovaním hodnoty α v rámci tej istej dimenzie prináša rapídne nižšiu frekvenciu zmien priradení, čo bolo naším cieľom. Pôvodné zhluky z dvojrozmerných dát sú s každou pridanou dimenziou zamiešané viac a viac, a pohyb spolu s tvarom, ktoré sme v nich namodelovali sa stráca. Keďže vzdialenosti sú vďaka vyššiemu počtu rozmerov väčšie, nižšia hodnota parametru α postačuje na to, aby sa objekty v príslušnom zhluku udržali. To v závere zníži počet výmen.

Je nutné dodať, že hodnoty, ktoré som v prípade *k*Medoids použil, sú priemerné z troch behov algoritmu pri rovnakej konfigurácii, pretože na rozdiel od metódy FCM, inicializačná fáza algoritmu *k*Medoids na prvej dátovej množine zo série funguje na spôsobe náhodného výberu počiatočných ťažísk, takže výsledné hodnoty pri rovnakej konfigurácii vstupu sa niekedy môžu výrazne líšiť.

5.7 Modifikácia FCM pre viacrozmerné dáta

Tak, ako v predošlej sekcii, výsledky prezentujeme pomocou tabuliek. Tentokrát sme na viacerých druhoch dát testovali priebeh rôznych konfigurácii modifikovaného algoritmu FCM.

Dáta / α	**1**	**0.9**	**0.8**
4D pseudonáhodné	516	294	236
6D pseudonáhodné	510	186	84
8D pseudonáhodné	460	44	20
10D pseudonáhodné	462	18	6

Modifikovaný FCM algoritmus – výsledné ohodnotenia priebehu algoritmu na pseudonáhodných umelých dátach. Hodnoty vyjadrujú počty zmien priradení objektov k zhlukom. Parameter $c = 3$.

Rovnako, ako v prípade *k*Medoids z predošlej sekcie, je vidno, že frekvencia zmien klesá s klesajúcim parametrom α, čo je správne. Na rozdiel od spomenutého príkladu, výsledok modifikovanej metódy FCM s parametrom $\alpha = 1$ sa nezhoršuje s narastajúcim počtom dimenzií. Postačuje nám k tomu upravená fáza inicializácie algoritmu FCM podobne, ako v kapitole 5.3, kde sme ukázali, že k požadovanému priebehu modifikovanej metódy FCM na dvojrozmernej verzii našich umelých dát nemusíme znižovať hodnotu parametru α. Metóda FCM sa v dvojrozmernej verzii javila ako stabilnejšia a vhodnejšia a v tomto prípade je to rovnako, čo je vidno na nízkych počtoch zmien príslušností objektov k zhlukom už pri vyšších hodnotách parametru α. Je nutné znovu dodať, že tieto dáta sú v skutočnosti istým spôsobom závislé a výsledky sú touto skutočnosťou do istej miery pozitívne ovplyvnené. Nasleduje prieskum modifikovanej metódy FCM na náhodných viacrozmerných dátach.

Dáta / α	**1**	**0.9**	**0.8**	**0.7**
4D náhodné	6082	3866	3253	2972
6D náhodné	7035	3221	2388	1696
8D náhodné	7645	1510	1217	795
10D náhodné	7847	624	357	62

Modifikovaný FCM algoritmus – výsledné ohodnotenia priebehu algoritmu na náhodných umelých dátach. Hodnoty vyjadrujú počty zmien priradení objektov k zhlukom. Parameter $c = 3$.

Znovu je vidno, že zníženie stupňa závislostí v dátach spôsobuje obrovský nárast frekvencie výmen príslušností, kde samo o sebe nepomáha ani modifikovaná inicializačná fáza metódy FCM. Pri narastajúcom počte rozmerov narastajú aj vzdialenosti, ktorým k tomu, aby sa udržali v príslušnom zhluku postačuje stále nižší a nižší stupeň úpravy ich veľkostí. Inými slovami, pri použití dát nižšieho stupňa závislostí a modifikovanej metódy FCM, k minimalizácii frekvencie zmien príslušností objektov k zhlukom pri narastajúcom počte dimenzií postačuje postupne stále vyššia hodnota parametru α.

5.8 Zhrnutie

V tejto kapitole sme naviazali na problém zhlukovania dynamických dát. Navrhli sme postup, ktorý nám umožní sledovať a ilustrovať vývoj zhlukov v dynamických dátach v čase. Tento postup sme aplikovali na dva rôzne zhlukovacie algoritmy, konkrétne ostrý zhlukovací algoritmus *k*Medoids a fuzzy metódu Fuzzy *c*Means. Experimentmi sme ukázali, že takto modifikované metódy podávajú na umelých dvojrozmerných dátach očakávané výsledky.

Cieľom práce je aplikovať tieto metódy na reálne dáta. Z toho dôvodu bolo nutné preskúmať vhodnosť nami modifikovaných metód na dátach viacrozmerného charakteru. Popísali sme, akým spôsobom sme vytvorili potrebné dáta, na ktorých sme pozorovali priebeh týchto metód. Keďže narastajúci počet dimenzií spôsobuje komplikácie pri vizualizácii, rozhodli sme sa použiť nami navrhnutý klasifikačný mechanizmus, ktorým ohodnotíme každé výsledné zhlukovanie. Princíp spočíva v počítaní zmien príslušností dátových objektov k zhlukom. Takto sme vyhodnotili priebeh modifikovaných metód *k*Medoids a FCM s rôznymi konfiguráciami vstupných parametrov a dát. Z analýzy výsledkov sme došli k záveru, že tieto metódy podávajú aj v rámci viacrozmerného priestoru uspokojujúce výsledky, pričom metóda FCM sa znovu javí ako vhodnejšia.

V nasledujúcej kapitole prestavíme súbor reálnych dát, na ktoré aplikujeme modifikované algoritmy z tejto kapitoly. V závere zhodnotíme a ilustrujeme výsledky ich priebehu.

6. Modifikácie a reálne dáta

Pripomeňme si, že cieľom tejto práce je zvoliť, prípadne navrhnúť vlastné modifikácie metód zhlukovej analýzy tak, aby bolo možné sledovať vývoj dynamických dát a ich zhlukov v čase. Vybrané metódy sú následné aplikované na reálne dáta. Dynamické dáta sú informácie tvorené sériou, v čase po sebe idúcich dátových množín, popisujúcich rovnaké vlastnosti danej množiny objektov.

V predošlej kapitole sme sa z hľadiska cieľa tejto práce zaoberali výskumom správania a vhodnosti aplikácie nami modifikovaných metód najprv na dvojrozmerných dátach a neskôr na ich viacrozmerných verziách. Zo získaných výsledkov sme usúdili, že takéto metódy umožňujú sledovať vývoj zhlukov v čase v rámci dynamických dát.

V rámci splnenia cieľa tejto diplomovej práce, v nasledujúcej kapitole aplikujeme spomínané metódy na skutočné, reálne dáta. Skôr, než tak učiníme, v časti 6.1 vysvetlíme, o aké dáta ide a akým spôsobom ich bolo nutné predspracovať. Neskôr, v sekcii 6.2, mimo samotnej aplikácii a ilustrácii priebehu zhlukovacích metód, určíme optimálny počet zhlukov, podobne ako v kapitole 4.2. V časti 6.3 sa pokúsime interpretovať výsledné zhlukovania. Na záver, v kapitole 6.4 zhodnotíme dosiahnuté výsledky a ich význam.

6.1 World Development Indicators

World Development Indiciators (WDI) je súbor takmer 1200 ukazovateľov o 209 štátoch z celého sveta, ktoré pokrývajú obdobie od roku 1960 do roku 2009 s ročnou periódou. Zdrojom týchto dát je Svetová banka [39]. Tieto dáta obsahujú indikátory z rôznych oblastí, ako napríklad: poľnohospodárstvo, zdravotníctvo, ekonomika, energetika, ťažobný priemysel, finančný sektor, infraštruktúra, chudoba, privátny a verejný sektor a iné.

Vo všeobecnosti sa jedná o ukazovatele popisujúce vývoj štátu z rôznych pohľadov. Pre potreby tejto práce som použil časť týchto dát, pretože majú aj svoje nedostatky v podobe veľkého počtu chýbajúcich hodnôt. Ak by som chcel nájsť ukazovatele, ktoré sú dostupné pre každý z 209 štátov, zúžilo by sa sledované časové obdobie na posledných 10 rokov a aj tak by sa našlo veľa výnimiek. Musel som tak mimo indikátorov redukovať aj počet štátov, na ktoré sa zameriam.

Nájsť indikátory, ktoré sú dostupné aspoň pre nejakých posledných 15 rokov pre aspoň polovicu štátov, zúžilo okno výberu na 97 štátov. Nakoniec som zredukoval počet indikátorov na týchto deväť ukazovateľov počas obdobia od roku 1990 do roku 2008 pri 97 štátoch:

1. **Consumer price index (2005 = 100)**: index spotrebných cien,
2. **Adjusted savings: consumption of fixed capital (% of GNI)**: spotreba fixného kapitalu, vyjadrené ako percentuálny podiel z hrubého domáceho príjmu,
3. **Inflation, GDP deflator (annual %)**: inflácia, deflátor hrubého domáceho produktu, v percentách,
4. **GDP growth (annual %)**: rast hrubého domáceho produktu, v percentách,
5. **Incidence of tuberculosis (per 100,000 people)**: výskyt tuberkulózy v prepočte na 100 000 ľudí,
6. **Labor participation rate, total (% of total population ages 15+)**: podiel pracovnej sily 15+, celkový, v percentách,
7. **Birth rate, crude (per 1,000 people)**: hrubá miera pôrodnosti v prepočte na 1000 ľudí,
8. **Death rate, crude (per 1,000 people)**: hrubá miera úmrtnosti v prepočte na 1000 ľudí,
9. **Merchandise trade (% of GDP)**: obchod s tovarom, percentuálny podiel z hrubého domáceho produktu.

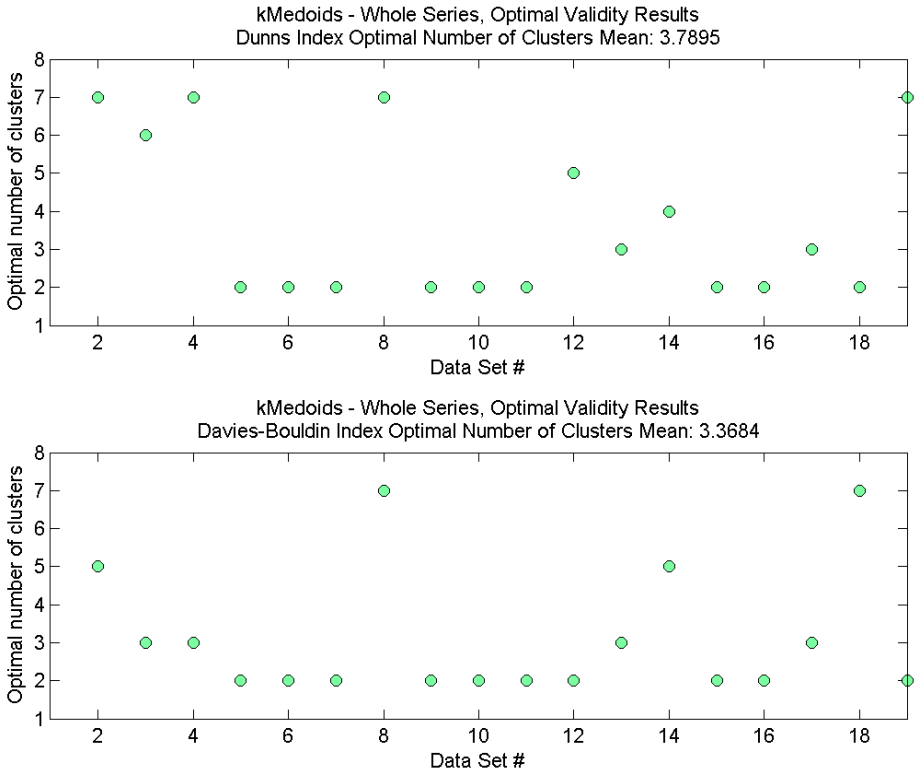
Takto vybrané ukazovatele som uložil pre každý štát do série 19 dátových množín, ktoré predstavujú stav od roku 1990 do roku 2008. Každá dátová množina je počas behu zhlukovacieho algoritmu normalizovaná lineárnou transformáciou do intervalu $\langle 0,1 \rangle$ s cieľom minimalizácie rozdielneho vplyvu atribútov na posudzovanie vzdialenosti.

Je nutné poznamenať, že tieto dáta boli vyberané s minimálnou znalosťou významu týchto ukazovateľov, preto sa pre odborníkovi nemusia pozdávať ako veľmi relevantné. To ale nevadí, pretože pre naše potreby je podstatné, že sa nejedná o umelé dáta. Na priloženom CD sa nachádzajú v súbore „`wdi_series.mat`“.

6.2 Priebeh metód na sérii reálnych dát

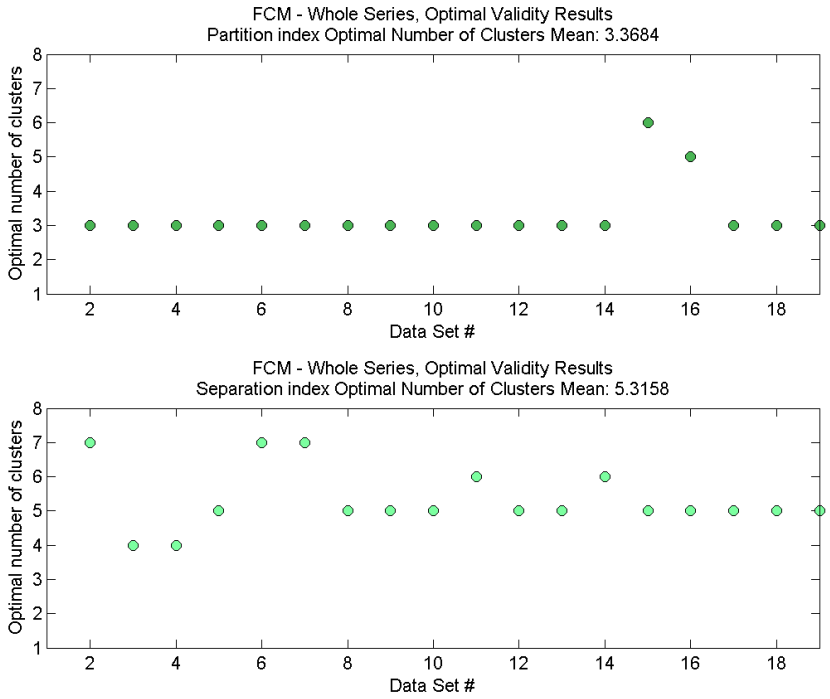
Pri aplikácii modifikovaných metód na reálne dáta budeme postupovať podobným spôsobom, ako v prípade umelých dát. Najprv určíme optimálny počet zhlukov a potom preskúmame samotný priebeh metód.

Optimálny počet zhlukov určíme ako priemer optimálnych počtov zhlukov každej dátovej množiny zo série. Dôvodom je, že chceme sledovať, ako sa zhluky v čase vyvíjajú, zväčšujú, zmenšujú a pohybujú. Ak by sa počas zhlukovania spojili alebo rozdelili, mohli by sme ich natrvalo stratiť, čo nechceme.



Obr. 27: Optimálne počty zhlukov pre jednotlivé dátové množiny zo série „`wdi_series.mat`“, zhlukovanie *k*Medoids.

V prípade metódy *k*Medoids určíme optimálny počet zhlukov $k = 3$.



Obr. 28: Optimálne počty zhlukov pre jednotlivé dátové množiny zo série „`wdi_series.mat`“, zhlukovanie FCM. Výsledok nebol až tak jednoznačný.

V prípade zhlukovania FCM je určenie optimálneho počtu zhlukov komplikovanejšie. Priemerné hodnoty partition koeficientu a koeficientu entrópie boli rovné dvom. Preto sme určili optimálny počet zhlukov $k = 3$.

Teraz, keď sme určili optimálny počet zhlukov, nám ostáva vyriešiť už len jeden problém, a to, ako vhodne ilustrovať priebeh zhlukovania na viacrozmerných dátach. Jedna z možností je použiť metódu analýzy hlavných komponent (Principal Component Analysis – PCA), čo je proces zníženia počtu dimenzií s čo najmenšou stratou informácií. Nevýhoda tejto metódy je, že vytvorí syntetické parametre, ktorých význam je pre človeka, ktorý nie je odborníkom ani na pôvodné dáta, ťažko pochopiteľný. Z toho dôvodu som vylúčil použitie takýchto metód.

Namiesto toho mi napadlo vizualizovať dané zhlukovanie postupom podobným, ako v kapitole 4 a 5, s tým rozdielom, že za osy x a y použijem vhodné, dva rôzne atribúty dát. Otázkou je už len zvoliť tie najvhodnejšie. Keď sa zamyslíme nad tým, ako fungujú zhlukovacie metódy *k*Medoids a FCM, je jasné, že hlavným prvkom v určovaní príslušnosti k zhlukom je práve vzdialenosť, ktorej modifikácia je aj myšlienkou navrhnutých modifikácií týchto metód. Pripomeňme si, že euklidovská vzdialenosť medzi dvoma n-dimenzionálnymi objektmi i a j je definovaná ako:

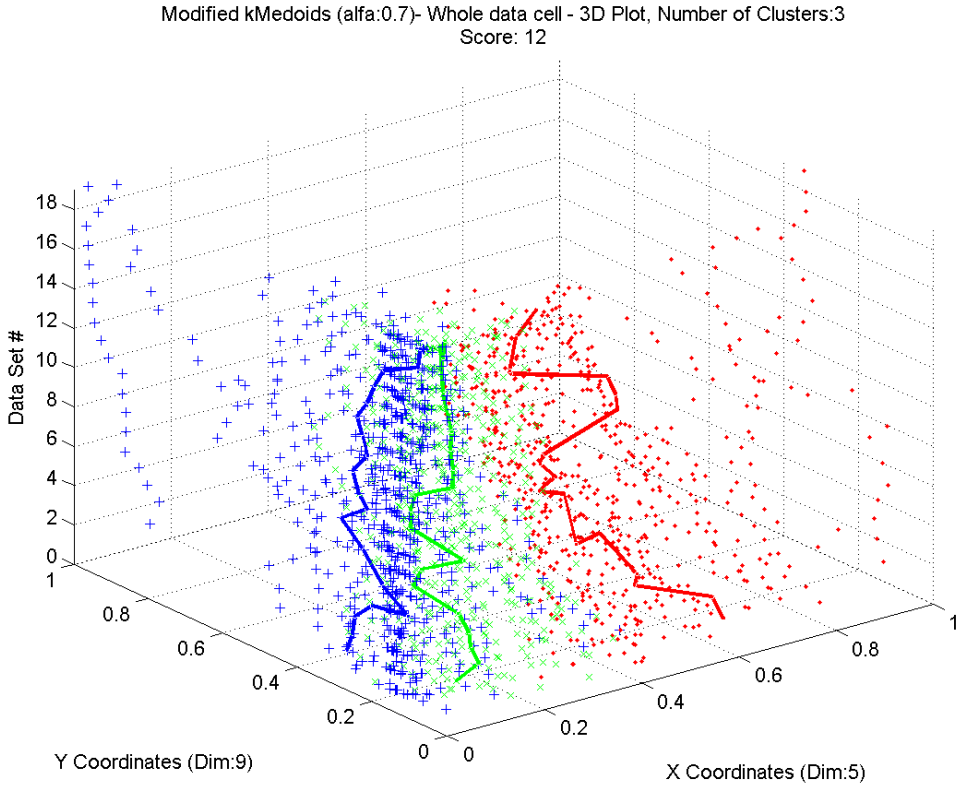
$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2}$$

Z definície je možné povedať, že v procese zhlukovania na viacrozmerných dátach má na vzdialenosť rôzny vplyv každá dimenzia. To, že niektorý atribút ma veľký vplyv na vzdialenosť, znamená aj to, že má veľký vplyv na priradzovanie zhlukov. Tento myšlienkový postup vedie k tomu, že výsledok zhlukovania sa pokúsime vizualizovať práve pomocou najvplyvnejších atribútov. Najvplyvnejšie atribúty sú tie, ktoré v najväčšej miere ovplyvňujú vzdialenosť medzi objektmi, a teda výsledné priradenie objektov k zhlukom.

Obrázok 29 zobrazuje výsledok aplikácie modifikovanej metódy *k*Medoids na spomínaných dátach zo Svetovej banky. Parameter $\alpha = 0.7,\ k = 3$ a vizualizujeme pomocou skladania dvojrozmerných grafov výsledných zhlukovaní, kde os x zobrazuje piaty atribút a os y deviaty atribút. Piaty a deviaty atribút dátového objektu reprezentujú nasledujúce vlastnosti štátu:

5. **Incidence of tuberculosis (per 100,000 people)**: výskyt tuberkulózy v prepočte na 100 000 ľudí,

9. **Merchandise trade (% of GDP)**: obchod s tovarom, percentuálny podiel z hrubého domáceho produktu.



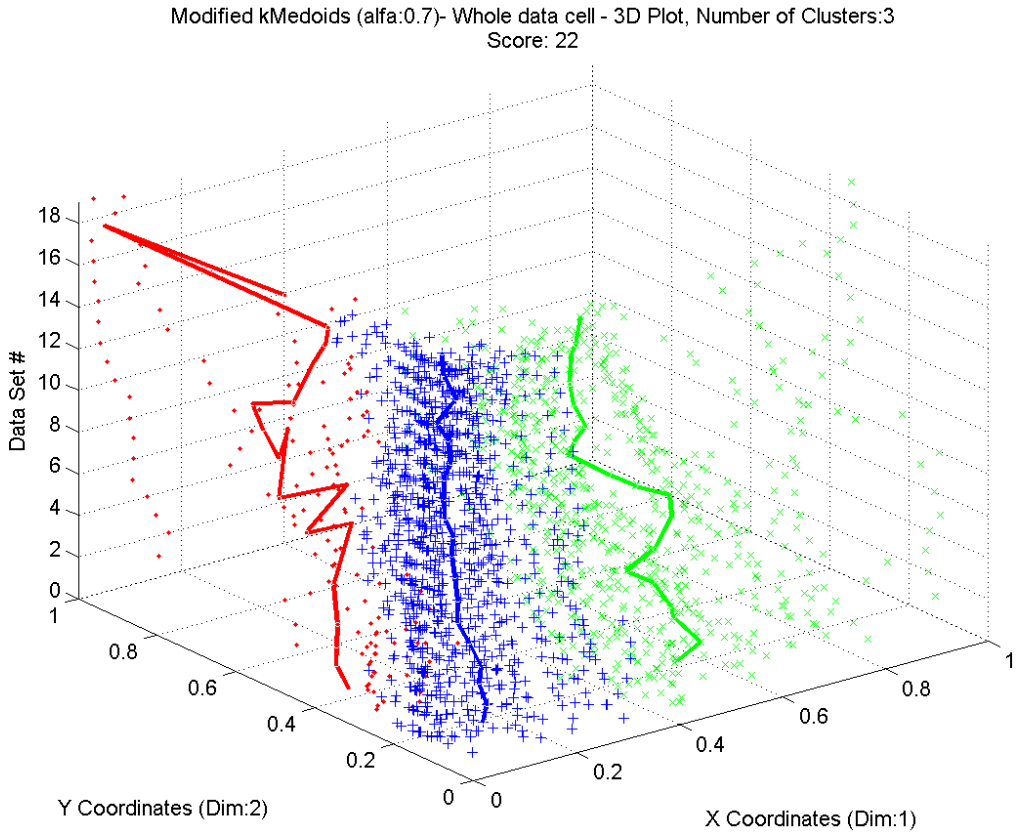
Obr. 29: Vizualizácia behu modifikácie algoritmu *k*Medoids na dátach „`wdi_series.mat`“. Parameter $\alpha = 0.7, k = 3$. Vizualizácia cez piatu a deviatu dimenziu.

Výsledné zobrazenie ukazuje, že výsledok zhlukovania je v značnej miere závislý od hodnoty piateho a deviateho parametru.

Je to vidno hlavne z faktu, že zhluky sú v tomto prípade v značnej miere rozdelené, neprekrývajú sa. Je možné vidieť ich časový vývoj. Hodnota našej klasifikačnej funkcie hovorí, že počas zhlukovania došlo k 12 výmenám príslušnosti k zhlukom. To je najskôr prezentované modrým zhlukom, ktorý s narastajúcim časom zmenšoval z pravej strany svoju veľkosť.

Pripomeňme, že nie každý beh takto nakonfigurovaného algoritmu a jeho vstupu podáva tak vhodný výsledok. Príčinou je náhodná inicializačná fáza tejto metódy. Na druhej strane, praxou overený postup pri zhlukovaní hovorí, že každé zhlukovanie je rozumné aplikovať niekoľkokrát a následne vybrať to najvhodnejšie.

Počas tohto zhlukovania mi napadlo porovnať tieto výsledky so zhlukovaním na dvojrozmernej verzii série dát „`wdi_series.mat`“, kde prvá dimenzia bude reprezentovať piaty rozmer z týchto dát a druhá dimenzia zase deviaty rozmer. Nasleduje priebeh algoritmu *k*Medoids (obr. 30) na takto vytvorených dátach s rovnakými vstupnými parametrami ako v predošlom prípade.

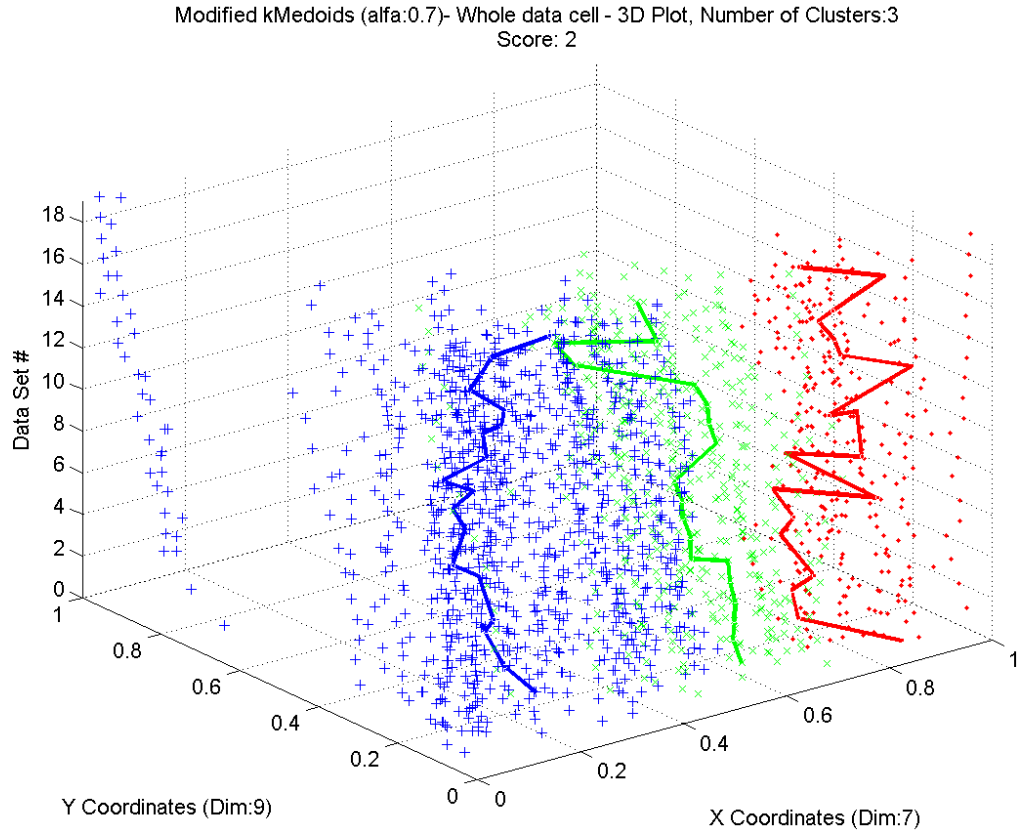


Obr. 30: Vizualizácia behu modifikácie algoritmu *k*Medoids na dvojrozmernej verzii dát „`wdi_series.mat`“. Parameter $\alpha = 0.7, k = 3$. Vizualizácia cez prvú (pôvodne piatu) a druhú (pôvodne deviatu) dimenziu.

Porovnaním výsledkov zistíme, že dvojrozmerné zhlukovanie prinieslo ostrejšie delenie zhlukov, čo je spôsobené faktom, že výsledné priradenia nie sú ovplyvnené zvyšnými dimenziami.

To podporuje naše rozhodnutie vizualizovať výsledok zhlukovania pomocou najvplyvnejších atribútov. Hodnoty našej klasifikačnej funkcie v týchto dvoch príkladoch sú porovnateľné.

Na obrázku 31 je vidno, že aj zobrazenie cez iné vhodné dimenzie umožňuje sledovať vývoj zhlukov v čase, a to z dôvodu, že zhluky sú viacmenej pekne ostro oddelené, neprekrývajú sa. Takto vhodne kombinovaných vizualizácií by sme mohli, ale nemuseli nájsť ešte zopár. Všetky rôzne od nich by však neposkytovali v tak vysokej miere ostré zhlukovania, interpretácia ich výsledkov by bola veľmi komplikovaná, a to z dôvodu, že nie všetky atribúty vplývajú na výsledné zhlukovanie v takej miere, aby sme pomocou ich vizualizácie boli schopní získať oddelené zhluky, z ktorých je možné vyčítať ich vývoj. Na našich dátach sme našli len dve kombinácie podávajúce rozumne vhodné výsledky.

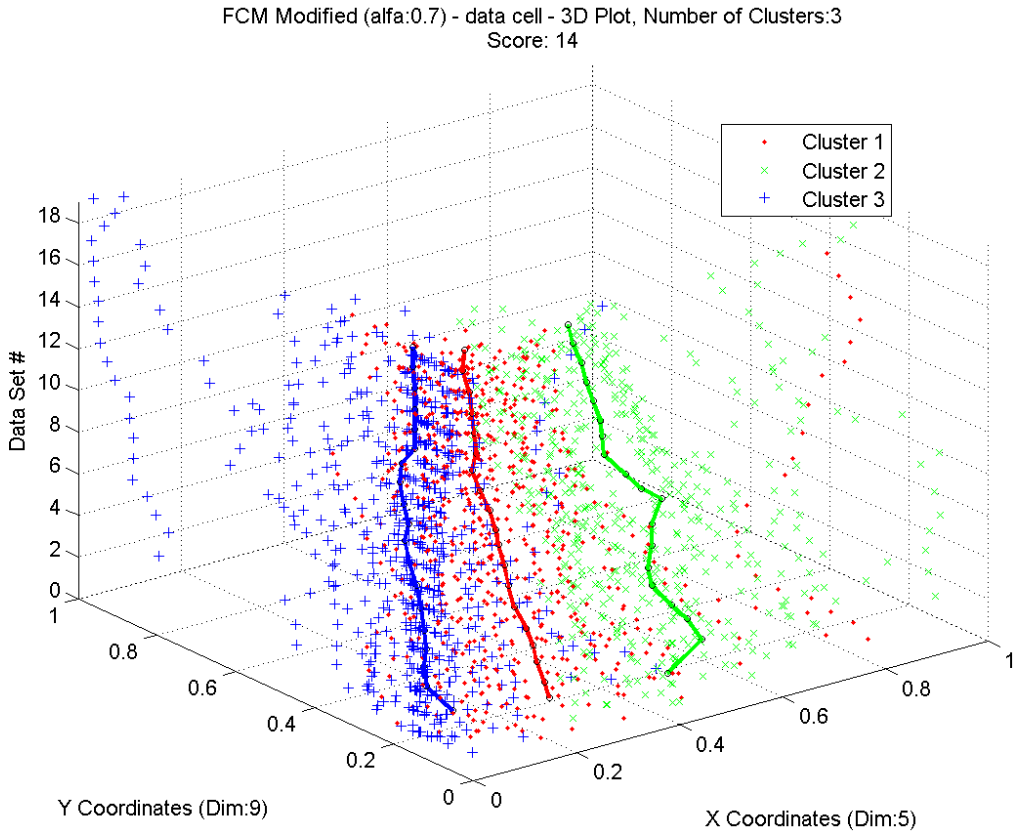


Obr. 31: Vizualizácia behu modifikácie algoritmu *k*Medoids na dátach „wdi_series.mat“. Parameter $\alpha = 0.7, k = 3$. Vizualizácia cez siedmu a deviatu dimenziu.

Dodám, že siedmy atribút dátového objektu určuje hrubú mieru pôrodnosti štátu v prepočte na tisíc ľudí.

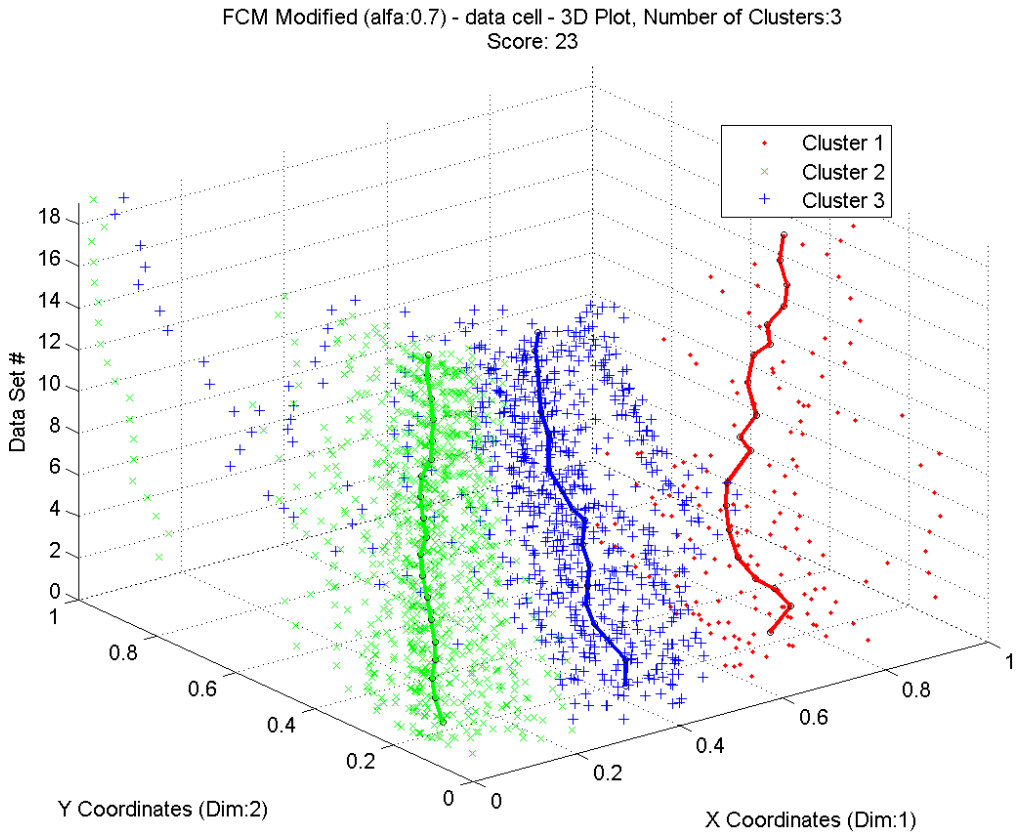
Na rozdiel od metódy *k*Medoids, pri použití modifikácie FCM neboli výsledky v prípade použitia na viacrozmernej verzii reálnych dát až tak vhodné.

Ilustrujeme dva prípady. Prvý (obr. 32), viacrozmerné dáta, vizualizácia cez piaty a deviaty rozmer. Druhý (obr. 33), dvojrozmerné dáta a vizualizácia cez prvú a druhú dimenziu (v skutočnosti piatu a deviatu). Parametre zhlukovania: $\alpha = 0.7$, $\beta = 0.75$ a $c = 3$.



Obr. 32: Vizualizácia behu modifikácie algoritmu FCM na dátach „`wdi_series.mat`“. Parameter $\alpha = 0.7, \beta = 0.75$ a $c = 3$. Vizualizácia cez piatu a deviatu dimenziu.

V prípade viacrozmernej verzie zhlukovania modifikácia FCM vo výsledku neposkytla až tak separované zhluky, ako metóda *k*Medoids, čo ilustruje objekt patriaci do prvého zhluku, ktorý sa nachádza v pravej časti obrázku 32. Je to spôsobené práve tým, že dáta sú viacrozmerné a teda posudzovanie príslušnosti k zhluku nie je tak jednoduché, čo potvrdzuje obrázok 33, ktorý vizualizuje vhodný priebeh zhlukovania metódou FCM na dvojrozmernej verzii dát vzniknutých zložením hodnôt piateho a deviateho atribútu potvrdzuje.



Obr. 33: Vizualizácia behu modifikácie algoritmu FCM na dvojrozmernej verzii dát „`wdi_series.mat`“. Parameter $\alpha = 0.7, \beta = 0.75$ a $c = 3$. Vizualizácia cez prvú (pôvodne piatu) a druhú (pôvodne deviatu) dimenziu.

Pri aplikovaní modifikovanému algoritmu FCM sme sa dostali k uspokojivým výsledkom najmä v zhlukovaní na dvojrozmernej verzii reálnych dát. Pri aplikácii metódy FCM na viacrozmernej verzii dát sme dospeli k nevhodnej zhlukovej príslušnosti niektorých objektov. Je nutné dodať, že spomínaný jav je z veľkej časti spôsobený dátovou viacrozmernosťou.

V 5. kapitole sme pomocou týchto metód dostali vhodné výsledky a metóda FCM sa javila ako stabilnejšia. V tomto prípade je v rámci viacrozmerných dát stabilnejšia naopak metóda *k*Medoids. Pri dvojrozmernej verzii zhlukovania boli obe metódy porovnateľné. Výsledky, ktoré sme dosiahli, podávali v rámci rozumných nastavení vstupných parametrov na reálnych dátach veľmi dobré výsledky v sledovaní vývoja zhlukov v čase, čo sa dá hodnotiť ako úspech z hľadiska nášho cieľa.

6.3 Interpretácia výsledkov

Počas aplikovania nami modifikovaných metód na reálne dáta zo Svetovej banky sme dospeli k zaujímavým poznatkom. Zistili sme, že viacrozmerné dáta je vhodné vizualizovať pomocou dvojrozmerných projekcií výsledkov umiestnených do trojrozmerného priestoru. Najvhodnejšie výsledky dosiahneme, ak tieto dvojrozmerné grafy jednotlivých medzivýsledkov zhlukovania na jednotlivých dátových množinách zo série budeme ilustrovať v rámci atribútov, ktoré najviac vplývajú na výsledné prideľovanie objektov k zhlukom. Vhodný výsledok sa dá poznať vďaka ostrosti rozdelenia zhlukov, inými slovami, čím najmenej sa zhluky prekrývajú, tým viac dané atribúty ovplyvňujú výsledné priradenia.

V rámci takýchto viacrozmerných dát môže, ale nemusí existovať viacero kombinácií dvoch atribútov, ktoré majú na priebeh algoritmu väčší vplyv, než ostatné. V prípade, že existuje väčšie množstvo kombinácií, naše výsledné zhlukovanie priberá na hodnote použitia. Ak existuje len zopár takých kombinácií, je vhodnejšie použiť príslušné dvojrozmerné zhlukovania, ktoré podávajú lepšie interpretovateľné výsledky. Je nutné spomenúť, že posudzovanie vhodnosti výsledku zhlukovacej metódy je v značnej miere subjektívne.

V experimentoch s metódou *k*Medoids sme ilustrovali kombináciu atribútov, pomocou ktorých podávala vizualizácia pozitívne výsledky z hľadiska nášho cieľa. Z dôvodu nedostatku znalostí v danej oblasti rozoberieme konkrétne dvojicu atribútov z obrázku 30, ktorý je výsledkom zhlukovania dvojrozmernej verzie reálnych dát, a teda sú vhodnejšie pre interpretáciu.

5. **Incidence of tuberculosis (per 100,000 people)**: výskyt tuberkulózy v prepočte na 100 000 ľudí. Tento atribút nie je potrebné detailnejšie popisovať.

9. **Merchandise trade (% of GDP)**: obchod s tovarom, percentuálny podiel z hrubého domáceho produktu. Je to súčet exportu a importu tovaru štátu vydelený hodnotou hrubého domáceho produktu v amerických dolároch.

V tomto prípade rozdelil zhlukovací algoritmus štáty v rámci výsledkov do 3 zhlukov.

Červený zhluk reprezentuje štáty s vysokou hodnotou tovarového obchodu z HDP štátu a s nízkym výskytom tuberkulózy. Vysoká hodnota tovarového obchodu z HDP značí, že príjmy a ekonomika štátu sú extrémne závislé od exportu a importu tovaru. Takýto profil by mal v skutočnosti sedieť na štáty svetových exportérov tovaru. V skutočnosti, zhlukovanie do tejto kategórie v rámci posledného zhlukovania priradil 4 štáty, ich počet sa samozrejme počas celého behu mohol meniť. Každopádne sú to štáty Belgicko, Hong Kong spolu s Čínou, Malajzia a Singapur. Ak sa nad tým zamyslíme, všetky štáty okrem Belgicka, sú mimo iného svetové jednotky v exporte rôznych elektronických zariadení a komponentov.

Belgicko je krajina, kde je nízky výskyt tuberkulózy, pretože sa jedná o vyspelú krajinu, no z jej geografickej polohy a najmä malej rozlohy je celkom logické, že je prítomná vysoká hodnota tovarového obchodu. Jej unikátnosť v rámci svetového exportu a importu môže spočívať v diamantovom trhu alebo trhu s čokoládou. Koniec dátovej série zaznamenáva počiatok poklesu hodnôt, čo sa dá vysvetliť práve dopadom ekonomickej krízy, pretože mimo finančného sektoru zasiahla aj výrobný sektor.

Modrý zhluk reprezentuje štáty s nízkou úrovňou tovarového obchodu, ale s nízkym výskytom tuberkulózy. V reálnom svete by to prirodzene mali byť vyspelé štáty, ekonomicky silné štáty, alebo štáty, ktoré prechádzajú zmenou medzi centrálne riadeným hospodárstvom a trhovým hospodárstvom, prípadne malé krajiny s trhovým hospodárstvom. Z druhého pohľadu by do týchto dát mohli patriť práve maximálne chudobné štáty. Podstatné je, že tam patria buď práve vyspelé štáty, alebo práve maximálne chudobné a nie ich kombinácia. Každopádne, v prípade vyspelého štátu, nižšia úroveň tovarového obchodu indikuje, že štát mimo exportu a importu tovaru, importuje a exportuje napríklad služby. Inými slovami, zdrojmi príjmu a generovania HDP nie je len vývoz hmotného tovaru, ale mnoho iného, ako napríklad terciárny sektor. Do tejto kategórie by mali v skutočnosti zapadať vyspelé ekonomiky sveta, prípadne významné z hľadiska svetoznámej dostupnosti nerastnej suroviny alebo produkcie a exportu rôznych iných komodít. Algoritmus klasifikoval do tejto kategórie celkovo 61 štátov, preto uvediem len niekoľko z nich: Dánsko, Veľká Británia, USA, Maďarsko, Taliansko, Nórsko, Egypt, Čína, Japonsko, Brazília, Austrália, Rakúsko, Argentína, Mexiko a mnoho iných. Tieto štáty majú spoločné to, že v rámci svojho kontinentu, patria k najvyspelejším, ako napríklad Egypt, ktorý je vďaka obrovskému zisku z cestovného ruchu považovaný za jednu z najvyspelejších krajín v Afrike.

Zelený zhluk reprezentuje štáty s nízkou úrovňou tovarového obchodu a vysokým výskytom tuberkulózy. Nízka úroveň tovarového obchodu v tomto prípade indikuje, že štát je závislý najmä od dovozu surovín, a to buď z dôvodu nevhodných klimatických podmienok alebo chudoby, čo vplýva samozrejme negatívne aj na úroveň zdravotníctva. Do tohto profilu by mali logicky zapadať štáty tretieho sveta, resp. štáty s nefunkčnými ekonomikami a veľmi nízkou úrovňou zdravotníctva. Zhlukovacia metóda do tejto kategórie priradila celkovo 32 štátov. Medzi tieto štáty patria napríklad: Bangladéš, Bhután, Pobrežie Slonoviny, Kamerun, Ghana, Gambia, Indonézia, India, Mozambik, Pakistan, Uganda, Uzbekistan, Nepál, Keňa a im podobné. Dá sa povedať, že skutočne ide o štáty s nízkou úrovňou zdravotníctva, či už z dôvodu klimatických alebo sociálnych podmienok. V ich vývoji pozorujeme, že tieto štáty majú tendenciu kolísať z hľadiska hodnoty výskytu tejto choroby. Výskyt tuberkulózy má v skutočnosti tendenciu kolísať, pretože tento vírus mutuje a stáva sa po čase odolným voči vakcínam a liekom. Po čase je vyvinutý nový typ vakcíny, ktorý na isté obdobie redukuje výskyt tejto choroby. Pohyb ťažiska zeleného zhluku tento jav odráža.

Výsledok zhlukovej analýzy môže, ale nemusí byť ľahko interpretovateľný. Záleží to od rôznych faktorov. Jeden z najdôležitejších je potreba odborníka na danú oblasť.

Nami vytvorené dedukcie sú príliš subjektívne, pretože sú založené na limitovanej znalosti tejto tematiky.

6.4 Zhrnutie kapitoly

V tejto kapitole sme aplikovali vybrané metódy na reálne dáta zo Svetovej banky, lepšie povedané na ich vybranú podmnožinu. Pomocou validačných ukazovateľov sme určili optimálny počet zhlukov, ako priemernú hodnotu optimálneho počtu zhlukov jednotlivých zhlukovaní na dátových množinách zo série dát. Pri samotnej aplikácii metód na reálne dáta sme narazili na problém, ako vhodne ilustrovať výsledky zhlukovania na viacrozmerných dátach.

Jednou z možností je metóda analýzy hlavných komponent, ktorá slúži k redukcii počtu dimenzií s čo najmenšou stratou informácií. To sme z dôvodu neprehľadnosti syntetických atribútov zavrhli. Namiesto toho sme vytvorili vizualizáciu viacrozmerných dát pomocou skladania dvojrozmerných grafov výsledných zhlukovaní jednotlivých dátových množín do trojrozmerného priestoru. Tieto dvojrozmerné grafy boli ilustrované pomocou viacerých kombinácii dvoch dominantných atribútov, ktoré najviac ovplyvňovali vzdialenosť v procese zhlukovania, a teda aj celkové výsledné priraďovanie objektov k zhlukom. Takto sme ilustrovali priebeh modifikovanej metódy *k*Medoids, pri ktorej sme pre porovnanie demonštrovali zhlukovanie na dvojrozmernej verzii týchto reálnych dát, ktoré vznikli zložením hodnôt spomínaných dominantných atribútov. Tieto výsledky boli veľmi porovnateľné, čo podporilo našu ideu, že dané atribúty skutočne najviac ovplyvňovali výsledok zhlukovania. Priebeh metódy FCM tentokrát nebol jednoznačne vhodnejší.

Veľmi dôležitým krokom analýzy priebehu zhlukovacej metódy je interpretácia výsledkov zhlukovania. V tejto kapitole sme subjektívne zanalyzovali význam konkrétneho výsledku, ktorý sme v rámci experimentov na reálnych dátach získali. Zistili sme, že naša metóda poskytla celkom logické a vysvetliteľné zhlukovanie.

Je nutné spomenúť, že rozdiel medzi zhlukovaním na týchto viacrozmerných dátach a na ich dvojrozmernej verzii zloženej z týchto dominantných atribútov, je v ich použití. Viacrozmerným zhlukovaním môže užívateľ pokryť viacero závislostí, kde nemusí len jeden z druhov vizualizácie podávať odborníkovi nejaký význam. Naproti tomu, dvojrozmerná verzia zhlukovania je síce viac jednoúčelová, no častokrát vieme, že chceme sledovať konkrétny vývoj práve na základe týchto atribútov.

Cieľom bolo navrhnúť postup, ktorý umožní sledovať vývoj zhlukov v čase, čo sa nám podarilo. Nami navrhnuté modifikácie vzdialeností sme parametrizovali, čo dokonca umožňuje tieto modifikácie aplikovať nie len na metódy *k*Medoids a FCM, ale na všetky zhlukovacie metódy založené na výpočte vzdialeností dvoch dátových objektov, čo môže poskytnúť ešte zaujímavejšie výsledky, ale z dôvodu obmedzeného časového priestoru sme sa k tomu nedostali.

7. Záver

Cieľom tejto diplomovej práce bolo zvoliť metódy, prípadne navrhnúť vlastné modifikácie metód zhlukovej analýzy, ktoré by umožňovali sledovať vývoj zhlukov dát v čase a tento cieľ sa nám splniť podarilo.

Navrhli sme postup, ktorý je založený na úprave inicializačnej fázy zhlukovania a modifikácii výpočtu vzdialeností medzi objektmi, pričom sme tieto úpravy parametrizovali. Inými slovami, previazali sme medzi sebou zhlukovania na jednotlivých dátových množinách.

Navrhnuté úpravy sme aplikovali na dve zhlukovacie metódy – *k*Medoids a fuzzy *c*Means. Posúdili sme vhodnosť použitia týchto metód nie len na dvojrozmerných umelých dátach, ale aj na ich viacrozmerných verziách, kde sme sledovali najmä vplyv parametru indikujúceho stupeň modifikácie vzdialenostnej matice v procese zhlukovania na hodnotu nami navrhnutej klasifikačnej funkcie. Táto klasifikačná funkcia určuje, koľkokrát došlo počas behu algoritmu na sérii dát k zmene príslušností objektov k zhlukom. Výsledky našich experimentov boli pozitívne.

Overené postupy sme aplikovali na reálne dáta zo Svetovej banky. Ich predspracovanie zahrnulo vhodný výber štátov a ukazovateľov, hlavne z hľadiska úplnej dostupnosti údajov. V rámci našej snahy správne vizualizovať výsledky zhlukovania na viacrozmerných dátach sme dospeli k vhodnému výsledku, ak sme zvolili vizualizáciu na základe tých atribútov, ktoré majú najväčší vplyv na výsledok zhlukovania. Všimli sme si, že rozdiel medzi zhlukovaním na viacrozmerných a dvojrozmerných dátach spočíva v šírke použitia ich výsledkov. Zatiaľ čo viacrozmerné zhlukovanie môže, ale nemusí poskytnúť náhľad do viacerých zaujímavých závislostí, dvojrozmerné zhlukovanie poskytuje konkrétny pohľad na jednu úlohu. K pochopeniu týchto závislostí je potrebná dobrá vizualizácia, ktorú sme dokázali poskytnúť. Boli sme schopní získať výsledok, ktorého význam bolo možné logicky interpretovať, čo je kritický krok v pozitívnej akceptácii výsledku zhlukovania. Tento krok môže byť veľmi komplikovaný, najmä ak užívateľ nie je odborníkom na tematiku, z ktorej dáta pochádzajú.

Tak ako pri každom diele, aj tu existujú smery, ktorými by sa práca mohla vydať. Nesporná výhoda týchto modifikácií je ich myšlienka, ktorá spočíva v úprave posudzovania vzdialeností. My sme túto ideu z dôvodu časového obmedzenia aplikovali na dve zhlukovacie metódy, no vďaka tomu, že sme tieto modifikácie kompletne parametrizovali a takmer všetky zhlukovacie metódy fungujú na báze posudzovania vzdialenosti, bolo by veľmi zaujímavé aplikovať tieto modifikácie na rôzne iné metódy, pri ktorých by sme možno dospeli k ešte zaujímavejším výsledkom. Mimo toho, zovšeobecnenie našej metódy by mohlo slúžiť na detekciu nezvyčajne tvarovaných zhlukov v dátach. V podstate, ľubovoľné dáta by sme mohli narezať namiesto času pomocou ktorejkoľvek inej dimenzie a sledovať tak vývoj, resp. tvar z hľadiska tejto dimenzie.

Zoznam literatúry

[1] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy: *Advances in Knowledge Discovery and Data Mining*, 1996

[2] J. Han, M. Kamber, J. Pei: *Data Mining: Concepts and Techniques*, 2006

[3] J. Kopáček: *Matematická analýza nejen pro fyziky*, Matfyzpress, 2007

[4] D. M. J. Tax, R. Duin, D. De Ridder: *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*, 2004

[5] L. Kaufman, P. J. Rousseeuw: *Finding Groups in Data*, 1990

[6] D.D.Mari, S.Kotz: *Correlation and dependence*, 2001

[7] R. V. Hogg, A. T. Craig:*Introduction to Mathematical Statistics, 5th ed.* 1995

[8] Ch. Romesburg: *Cluster Analysis for Researchers*, 2004

[9] R. Xu, D. C. Wunsch II: *Clustering*, 2009

[10] L. Kaufman, P. J. Rousseeuw: *Finding Groups in Data*, 2005

[11] K. Florek, J. Lukaszewicz, H. Perkal, H. Steinhaus, S. Zubrzycki: *Sur la liaison et la division des points d'un emsemble fini. Colloquium Mathematicum, 2*, 1951

[12] P.H.A. Sneath: *The applications of computers to taxonomy*, 1957

[13] L.L. McQuitty: *Hierarchical linkage analysis for the isolation of types*. *Education and Psychological measurements*, 1960

[14] R.R. Sokal, P.H.A. Sneath: *Principles of Numerical Taxonomy*, 1963

[15] P. Macnaughton-Smith: *Some Statistical and Other Numerical Techniques for Classifying Individuals, Studies in the causes of delinquency and the treatment of offenders*, 1965

[16] T. Zhang, R. Ramakrishnan, M. Livny: *BIRCH: An Efficient Data Clustering Method for Very Large Databases*, 1996

[17] S. Guha, R. Rastogi, K. Shim: *CURE: An efficient algorithm for clustering large databases*, 1998

[18] S. Guha, R. Rastogi, K. Shim: *ROCK: A robust clustering algorithm for categorical attributes*, 1999

[19] G. Karypis, E.H. Han, V. Kumar: *CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling*, 1999

[20] E. Forgy: *Cluster analysis of multivariate data: Efficiency versus interpretability of classification*, 1965

[21] J. C. *Bezdek: Pattern Recognition with Fuzzy Objective Function Algorithms*, 1981

[22] D.O. Hebb: *The organization of behavior*, 1949

[23] V. Kvasnička, Ľ. Beňušková, J. Pospíchal, I. Farkaš, P. Tiňo, A. Kráľ: *Úvod do teórie neurónových sietí*, 1997

[24] M. Halkidi, Y. Batistakis, M. Vazirgiannis: *Cluster validity methods:part I*, 2002

[25] M. Halkidi, Y. Batistakis, M. Vazirgiannis: *Cluster validity methods: part II*, 2002

[26] M. J. A. Berry, G. Linoff: *Data Mining Techniques for Marketing, Sales and Customer Support*, 1996

[27] S. Theodoridis, K. Koutroubas: *Pattern Recognition*, 1999

[28] D. L. Davies, D. W. Bouldin: Cluster Separation Measure, 1979

[29] M. Halkidi, Y. Batistakis, M. Vazirgiannis: On Clustering Validation Techniques, Journal of Intelligent Information Systems, 2001

[30] M. Halkidi, M. Vazirgiannis: Clustering Validity Assessment: Finding the Optimal Partitioning of a Data Set, 2001

[31] J. C. Dunn: *Well Separated Clusters and Optimal Fuzzy Partitions, Journal of Cybernetica*, 1974

[32] N. R. Pal, J. Biswas: *Cluster Validation using graph theoretic concepts, Pattern Recognition, Vol. 30, No. 4*, 1997

[33] S. Sharma: *Applied multivariate techniques*, 1996

[34] X.L. Xie, G. Beni, *A Validity Measure for Fuzzy Clustering*, 1991

[35] A.M. Bensaid, L.O. Hall, J.C. Bezdek, L.P. Clarke, M.L. Silbiger, J.A. Arrington, R.F. Murtagh: *Validity guided (re)clustering with applications to image segmentation*, 1996

[36] J. C. Bezdek: *Numerical Taxonomy with Fuzzy Sets*, 1974

[37] J. Abonyi, B. Balasko, B. Feil, Department of Process Engineering at the University of Veszprem, Hungary, 2005, http://www.fmt.vein.hu/softcomp/fclusttoolbox/

[38] Matlab FileExchange Central, *http://www.mathworks.com/matlabcentral/fileexchange/7473-clustering-and-data-analysis-toolbox*

[39] World Bank, *http://www.worldbank.org/*

Obsah priloženého CD

Na priloženom CD sa mimo textu diplomovej práce nachádzajú dáta, s ktorými som pracoval počas písania tejto diplomovej práce. Nachádzajú sa tu aj skripty metód, ktorými som sa dopracoval k prezentovaným výsledkom. Tieto skripty sú v maximálnej miere prispôsobené použitiu v tejto diplomovej práci.

- **/Data/** adresár s dátami, ktoré boli použité v diplomovej práci.
  - **/3D Heightmap Visualization/** adresár obsahujúci skripty použité k vizualizácii umelých dvojrozmerných dát „three_dots_2D.mat“.
  - **/Artificial Data Series Samples/** adresár obsahujúci skripty, ktorými boli vytvorené umelé viacrozmerné dáta.
    - **/Builder 10D from 2D - Fully Random/** generátor 4,6,8 a 10 rozmerných dát z pôvodných dvojrozmerných, náhodná verzia. Obsahuje aj tieto dáta.
    - **/Builder 10D from 2D - Partly Random/** generátor 4,6,8 a 10 rozmerných dát z pôvodných dvojrozmerných, pseudonáhodná verzia. Obsahuje aj tieto dáta.
    - **/Three Dots 2D/** základná dvojrozmerná verzia umelých dát.
  - **/World Bank/** adresár obsahujúci použité reálne, spracované dáta Svetovej banky.
    - **/Original WDI/** adresár obsahujúci originálny súbor dát z www.worldbank.org (**WDI_GDF.xlsx**).
    - **/wdi_series.mat** súbor použitej podmnožiny reálnych dát.
    - **/states.mat** zoznam štátov zo série wdi_series.mat, v matlab formáte

- **/Modified Methods/** adresár s metódami, členený podľa použitia v rámci umelých dát, reálnych dát a počtu dimenzií.
  - **/FCM Modified/** modifikovaná metóda FCM pre dvojrozmerné umelé dáta.
  - **/FCM Multi Modified/** modifikovaná metóda FCM pre ľubovoľný počet dimenzií na umelých dátach.
  - **/kMedoids Modified/** modifikovaná metóda kMedoids pre dvojrozmerné umelé dáta.
  - **/kMedoids Multi Modified/** modifikovaná metóda kMedoids pre ľubovoľný počet dimenzií na umelých dátach.
  - **/WDI Clustering/** adresár s modifikovanými zhlukovacími metódami s upravenými metódami vizualizácie pre použitie na reálnych dátach .

        - **/FCM Multi Modified/** modifikovaná metóda FCM pre použitie na reálnych dátach.
        - **/kMedoids Multi Modified/** modifikovaná metóda kMedoids pre použitie na reálnych dátach.

- **/Original Methods/** adresár s metódami pre ilustrovanie priebehu nemodifikovaných metód.
    - **/FCM Original/** aplikácia pôvodnej metódy FCM na umelé dvojrozmerné dáta.
    - **/kMedoids Original/** aplikácia pôvodnej metódy FCM na umelé dvojrozmerné dáta.

- **/Validity/** adresár so skriptami pre výpočet a ilustráciu optimálneho počtu zhlukov.
    - **/FCM Validity/** skript výpočtu optimálneho počtu zhlukov na ľubovoľných dátach pri metóde FCM.
    - **/kMedoids Validity/** skript výpočtu optimálneho počtu zhlukov ľubovoľných dátach pri metóde kMedoids.
    - **/kMedoids Validity (3D clustering)/** skript výpočtu optimálneho počtu zhlukov použitom pri 3D zhlukovaní na umelých dátach reprezentovaných ako jedna trojrozmerná matica.

- **/Text/** adresár s textom diplomovej práce.
    - **/Diplomova praca.pdf** text diplomovej práce vo formáte pdf.

- **/Original Toolbox/** adresár s pôvodným obsahom balíčka Fuzzy Clustering Toolbox.
    - **/FuzzyClusteringToolbox.pdf** manuál daného balíčka.

Pred spustením je nutné požadované skripty skopírovať na pevný disk. Všetky skripty sú spustiteľné v software *Matlab R2010b* pomocou súboru „**start.m**“ v príslušných adresároch, kde sa nachádzajú aj hlavné parametre daného skriptu. Každý skript, ktorého výsledok je nejaký obrázok, vytvorí v aktuálnom adresári podadresár **/results/**, kde sa tieto obrázky uložia.

Je nutné spomenúť, že všetky pôvodné metódy boli použité zo spomínaného Fuzzy Clustering Toolbox. Kód, ktorý obsahovali nebol takmer vôbec okomentovaný a obsahoval taktiež chyby, ktoré som musel odstrániť. Tento pôvodný kód som nekomentoval, pretože by to bolo časovo veľmi náročné, a nebolo to náplňou tejto práce. Modifikácie, ktoré som v nich uskutočnil sú okomentované.