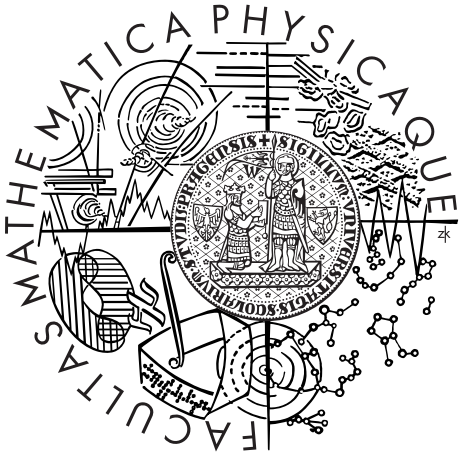
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Jana Hricová

Metoda k-průměrů

Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: prof. RNDr. Jaromír Antoch, CSc.

Studijní program: Matematika

Studijní obor: Finanční matematika

Praha 2011

Chcela by som poďakovať všetkým, ktorí mi akýmkoľvek spôsobom pomohli pri spracovaní tejto bakalárskej práce. Moje poďakovanie patrí najmä vedúcemu práce, prof. RNDr. Jaromírovi Antochovi, CSc., za vedenie a cenné pripomienky pri spracovaní práce.

Osobitné poďakovanie patrí mojim rodičom a mojim najbližším, bez ktorých podpory a pomoci by som to určite nezvládla.

Prehlasujem, že som túto bakalársku prácu vypracoval(a) samostatne a výhradne s použitím citovaných prameňov, literatúry a ďalších odborných zdrojov.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona v platnom znení, predovšetkým skutočnosť, že Univerzita Karlova v Prahe má právo na uzavretie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe dňa 3.8.2011 Podpis autora

Názov práce: Metoda k-průměrů

Autor: Jana Hricová

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedúci bakalárskej práce: prof. RNDr. Jaromír Antoch, CSc., Katedra pravděpodobnosti a matematické statistiky

Abstrakt:

Táto bakalárska práca pojednáva predovšetkým o štatistickej metóde k-priemerov, ktorá je súčasťou rozsiahlej množiny metód a algoritmov určených pre zhlukovú analýzu dát. Výsledky zhlukovej analýzy majú široké využitie napríklad pri ďalšej vedeckej činnosti, ale aj v marketingu, vedení firiem, poisťovníctve atď. Štatistické metódy zhlukovej analýzy vytvárajú z analyzovaných dát zhluky, ktoré sú tvorené podobnými objektmi. Podobnosť objektov je vyjadrená pomocou mier podobnosti, prípadne nepodobnosti.
Cieľom tejto práce bolo predstaviť algoritmus k-priemerov. Ide o nehierarchickú metódu, ktorá vyžaduje predom určeného počtu hľadaných zhlukov. V prostredí matematického softvéru Matlab sme aplikovali tento algoritmus na simulované a reálne dáta a výsledky interpretovali pomocou grafických a číselných výstupov.

Klúčové slová: k-priemerov, zhluková analýza, miera podobnosti, obrysový graf

Title: k-means method

Author: Jana Hricová

Department: Department of Probability and Mathematical Statistics

Supervisor: prof. RNDr. Jaromír Antoch, CSc., Department of Probability and Mathematical Statistics

Abstract:

This thesis deals with the statistical method k-means, which is a part of an extensive set of methods and algorithms designed for cluster analysis of data. Results of the cluster analysis are widely used in other scientific activities, but also in marketing, management or in insurance etc. Statistical methods for cluster analysis are creating clusters from analyzed datasets, which consist of similar objects. Similarity of two objects is expressed by dis-/similarity measure.
The aim of this thesis was to introduce the k-means algorithm. This is a non-hierarchical method with given number of output clusters as input. We have applied this algorithm in the enviroment of mathematical software Matlab on simulated and real data and have interpreted the results using graphical and numerical outputs.

Keywords: k-means, cluster analysis, dissimilarity measure, silhouette

Obsah

Úvod

Každodenný život nám prináša akýsi „súboj“ s množstvom informácií a skupín dát, ktoré k nám prúdia z rôznych meraní či pozorovaní. V dobe, kedy sú Internet a počítač dôležitou súčasťou nášho života, je už len samotné surfovanie na Internete príkladom, kde sa stretávame s množstvom dát, v ktorých vyhľadávame, ktoré separujeme a z ktorých selektujeme to, čo je pre nás potrebné. Samotné rozlišovanie a delenie do skupín ako jedna z najprimitívnejších aktivít ľudstva hrá dôležitú rolu v histórii ľudského vývoja. Rui Xu a Donald C. Wunsch [1] uvádzajú ako príklad v prírode delenie všetkých prírodných objektov na tri skupiny: rastlinstvo, živočíšstvo a minerály. Zvieratá sú na základe biologickej taxonómie ďalej rozdelené do kategórií na základe druhu, triedy atď. Na konci máme pomenovanie každého zvieraťa, teda rozlišujeme mačku od psa, žraloka od delfína, či kačicu od sliepky. Pomocou tohto delenia vieme interpretovať vlastnosti jednotlivých skupín (tried), ktoré vznikli skúmaním podobnosti charakteristík jednotlivých zvierat.

Množstvo dát v rôznych oblastiach vedy si vyžiadalo potrebu ich analýzy v zmysle redukcie a segmentácie. Práve tieto témy sú predmetom zhlukovej analýzy (Cluster analysis), ktorá skúma a analyzuje podobnosť sledovaných dát. Najznámejšie algoritmy zhlukovej analýzy si predstavíme v kapitolách tejto práce. Algoritmy zhlukovej analýzy majú široké využitie napríklad v oblastiach ako marketing, biológia a zdravotníctvo, poisťovníc tvo, plánovanie výstavby miest a infraštruktúry, analýza a prognóza spotreby vody v domácnostiach, definícia cieľových skupín spotrebiteľov, typizácia správania mladistvých vo voľnom čase, štúdie zemetrasení alebo klasifikácia elektronických dokumentov.[2]

Prvá kapitola tejto bakalárskej práce sa venuje úvodu do zhlukovej analýzy. Predstavuje pojem a význam zhlukovej analýzy ako štatistického oboru. V úvode kapitoly sú vysvetlené miery podobnosti, prípadne nepodobnosti, ktoré sa využívajú pri určovaní podobnosti objektov a premenných. Jednotlivé metódy zhlukovej analýzy sú popisované na základe ich klasifikácie, pričom v popise každej metódy pojednávame o jej matematickej formulácii a spôsobe, akým sa používa.

V Kapitole 2 sa zameriavame podrobnejšie na jednu z najznámejších zhlukovacích metód, ktorou je metóda k-priemerov. Vysvetľujeme jej zaradenie v rámci klasifikácie metód z predošlej kapitoly a uvádzame jej všeobecný algoritmus. Následne sú predstavené modifikované algoritmy metódy k-priemerov, ktoré sú príkladmi efektívnejších implementácií. V závere kapitoly je predstavený algoritmus k-medoidov, ktorý je založený na podobnom princípe zhlukovania dát.

Kapitola 3 je zameraná na praktickú aplikáciu metódy k-priemerov na reálne a simulované dáta. Na príkladoch simulovaných dát je aplikovaný algoritmus k-priemerov a jeho modifikovaný algoritmus k-priemerov^{++}. Druhá časť kapitoly je zameraná na aplikáciu tradičného algoritmu na finančné dáta, kde podrobne vysvetľujeme spracovanie dát spolu s interpretáciou výsledkov.

1. Úvod do zhlukovej analýzy

Termín zhluková analýza a jej definícia boli prvý krát sformulované roku 1939 Robertom C. Tryonom, profesorom psychológie. Zhluková analýza je pojmom pre súbor metód (procedúr), vytvárajúcich logický postup riešenia následujúceho problému: Máme súbor dát o ktorých nič nevieme. V tomto súbore chceme nájsť množiny dát, v rámci ktorých môžeme hovoriť o podobnosti dát v danej množine a zároveň docieliť heterogenitu týchto množín navzájom. Zhluková analýza teda umožňuje vyhľadávať zoskupenia podobných dát, teda zaradiť objekty do skupín (zhlukov) tak, aby si dva objekty rovnakého zhluku boli viac podobné, než dva objekty z rôznych zhlukov.

Základným cieľom zhlukovej analýzy je zjednodušovanie, redukcia dát. Na základe spoločných charakteristík jednotlivých množín dát sa vytvára typológia. Vytvárajú sa typy, ktoré sú charakterizované určitými vlastnosťami. Podľa nich dokážeme dané sledované údaje zaradiť do tried, ktoré sú jednoduchšie popísateľné. Klasifikácia týchto tried je založená na viacerých premenných, teda ide o viacdimenzionálne charakterizovanie tried. Samotná vizualizácia n-dimenzionálnych dát ako aj určenie zhlukov sú pri väčších dimenziách pre človeka náročné. Preto boli vytvorené metódy, ktoré to uľahčujú. Nesmieme takisto zabúdať na to, že použitím rôznych postupov zhlukovej analýzy na rovnaké dáta môžeme niekedy docieliť vytvorenie rôznej typológie, pretože typy objektov nie sú predom známe a vytvárajú sa počas samotného zhlukovania. Tým teda môže dochádzať k rôznej interpretácii dát.

S rozvojom matematicky orientovaných vedných odborov je analýza dát rôzne zaraďovaná, rôzne klasifikovaná a teda pre rovnakú problematiku sú používané rôzne terminológie.

Zhlukovaním budeme rozumieť postup, pri ktorom nie je predom známa príslušnosť žiadneho objektu, takisto nie je známy ani počet skupín. Cieľom bude klasifikovať všetky objekty zahrnuté do analýzy.

Postup zhlukovania môžeme zhrnúť do niekoľkých dôležitých bodov:

1. Voľba miery podobnosti dát
2. Voľba metódy (kritéria) zhlukovej analýzy
3. Stanovenie počtu zhlukov
4. Interpretácia výsledkov

1.1 Miery podobnosti

Prvým problémom pri zhlukovej analýze je určenie *podobnosti dvoch objektov*. Aby mohla byť podobnosť meraná, musí byť každý objekt charakterizovaný pomocou svojich vlastností. Napríklad rastlina je charakterizovaná tvarom listov, farbou kvetov atď.

Ako bolo vyššie spomenuté, cieľom zhlukovania je vytvoriť zhluky objektov, ktoré sú si najviac podobné. Preto dôležitú úlohu zohráva určenie, akým

spôsobom bude zisťovaná podobnosť. Na to slúžia *miery podobnosti*, ktorých voľba závisí od typu dát.

Miery podobnosti v ideálnom prípade nadobúdajú hodnoty z intervalu $\langle 0, 1\rangle$. Metódy zhlukovej analýzy sú však často založené na *mierach nepodobnosti*, prípadne vzdialenosti. Každú mieru podobnosti môžeme previesť na mieru nepodobnosti a to nasledujúcim spôsobom:

Uvažujme mieru podobnosti $S \in \langle 0, 1\rangle$. Potom mieru nepodobnosti $D \geq 0$ získame odpočítaním miery S od jedničky, tj. $D = 1 - S$.

1.1.1 Kvantitatívne dáta

Podobnosť objektov

Podobnosť objektov $\mathbf{x}_i$ a $\mathbf{x}_j$ budeme zapisovať ako $S(\mathbf{x}_i, \mathbf{x}_j)$, skrátene S_{ij}. Platí $S_{ij} = S_{ji}$.

Mieru nepodobnosti objektov $\mathbf{x}_i$ a $\mathbf{x}_j$ budeme zapisovať ako $D(\mathbf{x}_i, \mathbf{x}_j)$, prípadne skrátene D_{ij}. Ak by sme usporiadali jednotlivé hodnoty miery nepodobnosti sledovaných objektov do matice, išlo by o symetrickú maticu s nasledujúcimi vlastnosťami:

- $D_{ij} \geq 0$ (prvky matice sú nezáporné)
- $D_{ii} = 0$ (diagonálne prvky sú nulové)

V prípade kvantitatívnych dát sa na vyjadrenie podobnosti dvoch objektov využívajú *miery vzdialenosti*. Tie sú založené na reprezentácii objektu v priestore, teda jednotlivými premennými sú súradnice daného objektu v priestore.

Uvedieme najznámejšie typy vzdialeností a spôsob ich výpočtu. Uvažujme dva n-dimenzionálne body $\mathbf{x}_i = [x_{i1}, x_{i2}, ..., x_{in}]$ a $\mathbf{x}_j = [x_{j1}, x_{j2}, ..., x_{jn}]$, ktorých vzájomnú vzdialenosť zisťujeme.

Najznámejším a najbežnejšie používaným vyjadrením vzdialenosti dvoch viacrozmerných objektov je ***euklidovská vzdialenosť***.

$$D_E(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \tag{1.1}$$

Ak je splnená trojuholníková nerovnosť, tj.

$$D_{iy} \leq D_{ij} + D_{jy},$$

hovoríme o *metrike*.

Ďalšími používanými typmi vzdialeností sú:

Minkowského vzdialenosť

$$D_M(\mathbf{x}_i, \mathbf{x}_j) = (\sum_{k=1}^{n}(x_{ik} - x_{jk})^r)^{\frac{1}{r}} \tag{1.2}$$

Je zovšeobecnením predchádzajúcej metriky ($r = 2$).

manhattanská vzdialenosť (mestských blokov)

$$D_B(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{n} |x_{ik} - x_{jk}| \tag{1.3}$$

Redukuje vplyv extrémnych vzdialeností.

Čebyševova vzdialenosť

$$D_C(\mathbf{x}_i, \mathbf{x}_j) = max_k(|x_{ik} - x_{jk}|)$$

Podobnosť premenných

Zatiaľ čo vzdialenosť predstavuje vzťah objektov kvantitatívnych dát, v prípade vyjadrenia vzťahu medzi premennými hovoríme o *závislosti*. Miery intenzity vzájomnej štatistickej závislosti teda vyjadrujú podobnosť dvoch premenných.

Ako základná miera podobnosti dvoch kvantitatívnych premenných sa používa výberový ***Pearsonov korelačný koeficient***, ktorý je pre k-tú a l-tú premennú daný predpisom

$$r_{kl} = \frac{\sum_{i=1}^{n}(x_{ik} - \bar{x}_k)(x_{il} - \bar{x}_l)}{\sqrt{\sum_{i=1}^{n}(x_{ik} - \bar{x}_k)^2 \sum_{i=1}^{n}(x_{il} - \bar{x}_l)^2}},$$

kde $\bar{x}_k$ je aritmetický priemer hodnôt k-tej premennej, analogicky $\bar{x}_l$ je priemer l-tej premennej. Hodnoty korelačného koeficientu sa nachádzajú v intervale od -1 do 1. Hovoríme, že premenné sú nezávislé, ak je hodnota korelačného koeficientu rovná 0.

Interpretácia posúdenia podobnosti nie je jednoznačná. V knihe „Shluková analýza dat“ [6] autori uvádzajú dva spôsoby interpretácie, ktoré závisia na charaktere hodnôt a pohľade analytika.

V prvom prípade hodnota -1 znamená maximálny nesúhlas. Potom prevod na mieru nepodobnosti je daný predpisom $D_{kl} = 1 - r_{kl}$, teda rovnakým spôsobom ako bolo spomenuté pri prevodoch medzi mierami podobnosti a nepodobnosti.

V druhom prípade uvažujeme ekvivalentne obe hodnoty -1 a 1 ako maximálny súhlas medzi dvomi premennými. Potom pre výpočet nepodobnosti volíme buď vzťah $D_{kl} = 1 - r_{kl}^2$, alebo vzťah $D_{kl} = 1 - |r_{kl}|$.

V prípade, že $\bar{x}_k = 0$ a $\bar{x}_l = 0$, získavame *kosínovú mieru*

$$S_K(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{i=1}^{n} x_{ik}x_{il}}{\sqrt{\sum_{i=1}^{n} x_{ik}^2 \sum_{i=1}^{n} x_{il}^2}} \tag{1.4}$$

1.1.2 Binárne dáta

V týchto prípadoch sa na určenie podobnosti využívajú asociačné koeficienty, pričom vzorce budú uvádzané pomocou početností v kontingenčnej tabuľke. V našom prípade budeme uvažovať 4-polnú kontingenčnú tabuľku, ktorú v literatúre nájdeme aj pod názvom asociačná tabuľka (od toho potom odvodený názov koeficienty asociácie). Predstavíme si niektoré z nich.[3]

Uvažujme objekty A a B a ich asociačnú tabuľku 2×2, kde 1 znamená prítomnosť premennej a 0 opak.

		A	
		1	0
B	1	a	b
	0	c	d

Miery podobnosti, nepodobnosti a vzdialenosti

V prípade binárnych premenných je potrebné rozlišovať symetrické a asymetrické premenné.

Nepodobnosť dvoch *symetrických* premenných môžeme zistiť pomocou *koeficientu nezhody*. Vyjadruje podiel počtu objektov, pri ktorých sú odlišné hodnoty dvoch sledovaných premenných, k celkovému počtu objektov.

Podobnosť môžeme merať pomocou ***Sokalovho a Michenerovho koeficientu prostej zhody (simple matching)***, ktorý vyjadruje podiel objektov so zhodnými hodnotami k celkovému počtu objektov.

$$S_{SM} = \frac{a+d}{a+b+c+d} \quad , \tag{1.5}$$

kde S_{SM} je z intervalu (0,1). Potom vzdialenosť objektov (miera nepodobnosti) je vyjadrená vzťahom

$$D = 1 - S_{SM}$$

Nepodobnosť dvoch *asymetrických* premenných môžeme vyjadriť pomocou ***Jaccardovho koeficientu***, kde jeho hodnotu získame podielom počtu objektov, u ktorých sú odlišné hodnoty dvoch sledovaných premenných, k počtu objektov, u ktorých sa aspoň pri jednej premennej vyskytuje hodnota 1.

$$D_J = \frac{b+c}{a+b+c}$$

Je používaný aj *Jaccardov koeficient podobnosti* daný vzorcom

$$S_J = \frac{a}{a+b+c} \quad ,$$

kde S_J je takisto z intervalu (0,1).

Opäť mieru nepodobnosti získame odpočítaním od jedničky, tj.

$$D = 1 - S_J$$

Kosínova miera pre dve binárne premenné vyjadrená pomocou početností sa nazýva ***Ochaiaov koeficient***. Výpočet prebieha pomocou vzorca

$$S_O = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$$

Ako mieru vzdialenosti uvedieme ***štvorcovú euklidovskú vzdialenosť***, čo predstavuje súčet početností $b+c$. Rovnako ***Hammingova vzdialenosť*** (manhattanská vzdialenosť aplikovaná na binárne dáta) je vyjadrená tým istým súčtom. Teda platia vzťahy

$$D_{ES} = D_B = b + c$$

Teda z toho plynie vzorec pre ***euklidovskú vzdialenosť***, ktorú môžeme taktiež použiť ako mieru vzdialenosti binárnych dát.

$$D_E = \sqrt{b + c}$$

Medzi euklidovskou vzdialenosťou a koeficientom prostej zhody platí vzájomný vzťah

$$D_E = \sqrt{(a + b + c + d)(1 - S_{SM})}$$

Miery závislosti

V tejto podkapitole spomenieme len niektoré z používaných mier. Jednou zo základných mier vyjadrenia závislosti je ***pomer šancí***. Je vyjadrením krížových súčinov, to znamená, že

$$\vartheta = \frac{ad}{bc} = \frac{a/b}{c/d}$$

Tento pomer nadobúda hodnoty od nula do nekonečna, v prípade nezávislosti medzi premennými nadobúda hodnotu 1. Hodnota blížiaca sa nule indikuje silnú závislosť.

Na základe pomeru krížových súčinov je navrhnutá aj miera ***Yuleovo Q***, pre ktorú platí

$$Q = \frac{ad - bc}{ad + bc} = \frac{ad/bc - 1}{ad/bc + 1} = \frac{\vartheta - 1}{\vartheta + 1}$$

Hodnoty tohto koeficientu sa pohybujú v intervale $\langle -1, 1 \rangle$, pričom krajné hodnoty nadobúda vtedy, ak je niektorá z hodnôt v asociačnej tabuľke nulová.

Ďalšou mierou intenzity závislosti je ***Pearsonov korelačný koeficient*** vyjadrený pomocou početností vzťahom

$$r = \frac{ad - bc}{\sqrt{(a + b)(a + c)(b + d)(c + d)}}$$

Korelačný koeficient môže nadobúdať krajné medze intervalu $\langle -1, 1 \rangle$ v prípade, že obe hodnoty na diagonále asociačnej tabuľky sú nulové.

Na tomto koeficiente je založený ***koeficient*** φ, počítaný podľa vzťahu

$$\varphi = \sqrt{\frac{\chi^2}{a + b + c + d}}$$

Rozdielom je obor hodnôt, ktorým je v tomto prípade interval $\langle 0, 1 \rangle$.

Ďalšie miery sú uvedené napríklad v knihe [6].

1.2 Klasifikácia metód zhlukovania

Metódy zhlukovej analýzy nebývajú klasifikované podľa matematických prostriedkov, ktoré používajú. Ide o klasifikáciu podľa cieľov, ku ktorým smerujú. Štandardne sa stretávame s delením podľa spôsobu organizácie objektov do zhlukov:

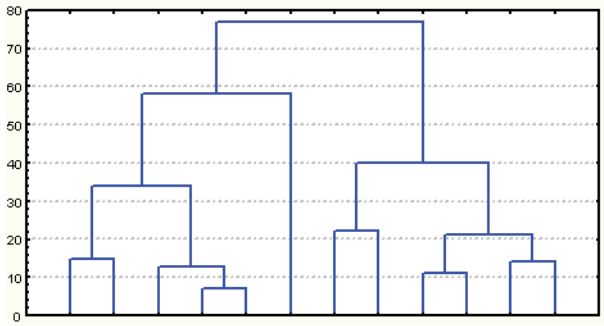
- Hierarchické metódy
- Nehierarchické metódy

1.2.1 Hierarchické metódy

Hierarchické zhlukovacie metódy sú založené na postupnom zhlukovaní objektov do hierarchického systému zhlukov, pričom ide o systém neprázdnych disjunktných množín pôvodnej množiny. Podľa toho, ako túto štruktúru zhlukov tvoríme, delíme hierarchické zhlukovacie metódy na:

- ***Aglometatívne hierarchické metódy***
- ***Divízne hierarchické metódy***.

Grafickému znázorneniu vytvorenej hierarchickej štruktúry hovoríme dendrogram (stromový diagram). Prednosťou dendrogramu je prehľadnosť nad celkovou stavbou skúmaného materiálu. V rámci neho môžeme v každom kroku odsledovať postupné spájanie sa jednotlivých zhlukov, ako je znázornené na obrázku 1.1



Obr. 1.1: Dendrogram

Aglomeratívne hierarchické metódy (postupy zdola nahor)

Proces zhlukovania začína od jednotlivých objektov. Teda na začiatku tvoria samotné dáta jednotlivé zhluky. Postupne sa vytvára stromová štruktúra, pri ktorej sa zmenšuje počet zhlukov zjednotením zhlukov predchádzajúcich. Zjednocujú sa najpodobnejšie zhluky, pričom prvý zhluk je vytvorený z objektov na základe matice (ne)podobnosti (viď miery vzdialenosti v podkapitole 1.1.1). Súčasťou algoritmu aglomeratívnych hierarchických metód je stanovenie hodnoty D_{max}, ktorá predstavuje užívateľom zvolenú maximálnu vzdialenosť dvoch spájajúcich sa zhlukov a podľa ktorej sa následne v každej iterácii určuje, či sa dané zhluky zlúčia. Ak hodnota vzdialenosti dvoch uvažovaných zhlukov presahuje hodnotu D_{max}, algoritmus predčasne končí, čo znamená, že požadovaný počet zhlukov nie je dosiahnutý. V takom prípade je potrebné prahovú hodnotu D_{max} zvýšiť.

Algoritmus 1 Aglomeratívne hierarchické metódy

Vstup: Dáta $X_1, ..., X_n$.
Vstup: C (*požadovaný počet zhlukov ; ak C=1 dostávame dendrogram*)
Vstup: D_{max} (*maximálna zvolená vzdialenosť dvoch zhlukov pri zlúčení*)

1: Inicializácia C, $X_i = C_i$ pre každé $i = 1, ..., n$ (*Symbolom C_i označíme i-tý zhluk*)
2: $temp = n$ (*temp pomocná premenná predstavuje aktuálny počet zhlukov*)
3: **repeat**
4: Pre jednotlivé dvojice zhlukov C_i a C_j napočítame maticu vzdialeností s prvkami D_{ij} (*Vzdialenosť D_{ij} dvoch zhlukov určíme pomocou jednej z metód popísaných nižšie*)
5: Nájdeme dva najbližšie zhluky C_i a C_j a zistíme, či $D_{ij} \leq D_{max}$ (*ak presahuje hodnotu D_{max}, algoritmus končí*).
6: Zlúčime zhluky C_i a C_j a $temp = temp - 1$.
7: **until** $temp = C$

Výstup: Jednotlivé získané zhluky, prípadne dendrogram, vzdialenosti medzi centrami každého zhluku a informáciu, či skončil algoritmus predčasne.

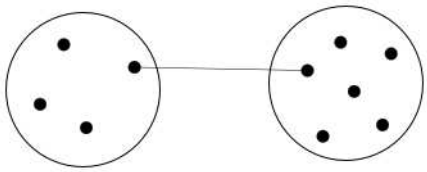
Vzdialenosť ((ne)podobnosť) zhlukov môže byť stanovená pomocou rôznych aglomeratívnych algoritmov:

- ***Metóda najbližšieho suseda (Single linkage method)***

  Ide o jednu z najjednoduchších metód zhlukovania. Princípom tejto metódy je, že vzdialenosť medzi dvomi zhlukmi je definovaná ako vzdialenosť medzi najbližším párom objektov, kde sú do úvahy brané páry pozostávajúce z objektov dvoch rôznych zhlukov. Označme $D(r, s)$ vzdialenosť dvoch zhlukov. Potom

  $$D(r, s) = \min\{d_{ij} : i \in r, j \in s\}$$

  V ďalšej fázi zhlukovania sa spájajú zhluky s najmenšou $D(r, s)$. Grafické znázornenie ponúka obrázok 1.2.



Obr. 1.2: Single linkage

- ***Metóda najvzdialenejšieho suseda (Complete linkage method)***

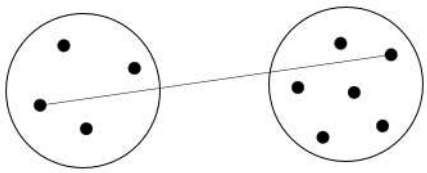
  Metóda Complete linkage (obrázok 1.3), nazývaná aj metódou najvzdialenejšieho suseda je opakom predchádzajúcej metódy. Vzdialenosť medzi

zhlukmi je teraz definovaná ako vzdialenosť medzi najvzdialenejšou dvojicou objektov. Každá dvojica je opäť tvorená z dvoch objektov navzájom rôznych zhlukov.

Predpokladajme opäť, že $D(r, s)$ je vzdialenosť dvoch zhlukov. Potom

$$D(r, s) = \max\{d_{ij} : i \in r, j \in s\}$$

Maximálna hodnota dvoch objektov rôznych zhlukov je prehlásená za vzdialenosť $D(r, s)$ zhlukov r a s. V dalšej fáze hierarchického zhlukovania sú zlúčené zhluky pre ktoré je $D(r, s)$ minimálna.



Obr. 1.3: Complete linkage

- ***Metóda priemernej väzby suseda (Average linkage method)***

  Tu je vzdialenosť $D(r, s)$ medzi dvoma zhlukmi definovaná ako priemer vzdialeností medzi všetkými pármi objektov, pričom každý pár je tvorený objektami dvoch rôznych zhlukov.

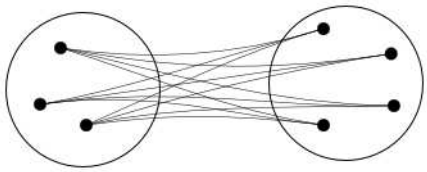
  Označme:

  - r a s sú dve skupiny objektov (zhluky)
  - N_r a N_s je počet objektov v jednotlivých zhlukoch r a s
  - d_{ij} vzdialenosť dvoch objektov $i \in r$ a $j \in s$

  Potom

$$D(r, s) = \frac{1}{N_r \times N_s} \sum_{i=1}^{N_r} \sum_{j=1}^{N_s} d_{ij}$$

  K zlúčeniu zhlukov dochádza v prípade, kde $D(r, s)$ je minimálna pre zhluky r a s (Obrázok 1.4).



Obr. 1.4: Average linkage

- ***Centroidná metóda (Centroid linkage method))***
  Pri tejto metóde je vzdialenosť dvoch zhlukov definovaná ako euklidovská vzdialenosť centroidov („priemerný prvok“) dvoch disjunktných zhlukov. Označme

  – centroid zhluku r

  $$\bar{x}_r = \frac{1}{N_r}\sum_{i=1}^{N_r} x_{ri}$$

  – centroid zhluku s analogicky $\bar{x}_s$

  Vzdialenosť $D(r,s)$ je definovaná predpisom

  $$D(r,s) = \|\bar{x}_r - \bar{x}_s\|_2$$

  Vylepšením tejto metódy je *Vážená párová metóda (Weighted pair-group centroid)* , ktorá definuje vzdialenosť $D(r,s)$ ako euklidovskú vzdialenosť vážených centroidov.
  Potom platí

  $$D(r,s) = \|\bar{\bar{x}}_r - \bar{\bar{x}}_s\|_2 \quad ,$$

  kde $\bar{\bar{x}}_r = \frac{1}{N_r}\sum_{i=1}^{N_r} w_i x_{ri}$ je vážený centroid zhluku r, $w_i \geq 0, \sum_i w_i = 1$. Ak zhluk r vznikol zjednotením zhlukov p a q, potom

  $$\bar{x}_r = \frac{1}{2}(\bar{x}_p + \bar{x}_q).$$

  Analogicky definujeme vážený centroid zhluku s.

- ***Wardova metóda (Ward's hierarchical clustering method)***

  Wardov algoritmus (Ward, 1963) je bežne používaný postup, ktorý je určený pre dáta metrických priestorov. Ward navrhol zhlukovací postup smerujúci k vytvoreniu množín spôsobom, ktorý minimalizuje straty spojené s každým zhlukovaním. Máme zadaných n skupín. Tento postup ich počet zníži na $n-1$ navzájom disjunktných skupín tak, že uvažuje spojenie všetkých možných $\frac{n*(n-1)}{2}$ dvojíc a v každom kroku zjednotí dva zhluky, ktorých zlúčenie má za následok minimálne zvýšenie „straty informácie“. Táto strata informácie je matematicky vyjadrená ako chyba súčtu štvorcov ESS (error sum-of-squares). Pre zhluk r platí:

  $$ESS_r = \sum_{l\in r}\sum_{i=1}^{n} x_{il}^2 - \frac{1}{n}(\sum_{i=1}^{n} x_{ir})^2$$

  Podľa Wardovej metódy sú v každom kroku spájané zhluky, pre ktoré je nárast ESS minimálny. Vzdialenosť dvoch zhlukov r a s je daná predpisom

  $$D(r,s) = N_r N_s \frac{\|\bar{x}_r - \bar{x}_s\|_2^2}{(N_r + N_s)} \quad ,$$

  kde $\|.\|_2$ je euklidovská metrika a $\bar{x}_r$ a $\bar{x}_s$ sú definované ako v predchádzajúcej metóde.

Divízne hierarchické metódy

Postup zhlukovania môže byť aj opačný ako pri aglomeratívnych hierarchických metódach. Vytvárame rozklad pôvodnej množiny, ktorý je postupným zjemnením predchádzajúceho zhluku a iteratívne zvyšujeme počet jednotlivých zhlukov. Na konci by sme mali dôjsť k samotným jednoprvkovým zhlukom.

Algoritmus 2 Divízne hierarchické metódy

Vstup: Dáta $\mathbf{x}_1, ..., \mathbf{x}_n$, tvoriace jeden zhluk.
Vstup: C (*požadovaný počet zhlukov*)
1: Inicializácia C.
2: $temp = 1$ (*temp pomocná premenná predstavuje aktuálny počet zhlukov*)
3: **repeat**
4: Pre jednotlivé dvojice objektov x_i a x_j, $i, j = 1, ..., n$ napočítame maticu vzdialeností s prvkami D_{ij} (*Vzdialenosť D_{ij} dvoch objektov viď podkapitola 1.1.1*)
5: Nájdeme dva najvzdialenejšie objekty $\mathbf{x}_i$ a $\mathbf{x}_j$ (*Tieto objekty budú reprezentantmi dvoch podmnožín, na ktoré pôvodnú množinu rozdelíme*)
6: **for all** $\mathbf{x}_l$ také, že $l = 1, ..., n, l \neq i, j$ **do**
7: **if** vzdialenosť $D(\mathbf{x}_l, \mathbf{x}_i) < D(\mathbf{x}_l, \mathbf{x}_j)$ **then**
8: prvok $\mathbf{x}_l$ patrí do zhluku reprezentanta $\mathbf{x}_i$
9: **else**
10: prvok $\mathbf{x}_l$ patrí do zhluku reprezentanta $\mathbf{x}_j$
11: **end if**
12: **end for**
13: **until** $temp = C$
Výstup: Jednotlivé získané zhluky, prípadne dendrogram, vzdialenosti medzi centrami každého zhluku.

1.2.2 Nehierarchické metódy

Nehierarchické zhlukovacie metódy nevytvárajú na rozdiel od hierarchického prístupu spomínanú stromovú štruktúru. Tieto metódy začínajú z náhodne vybraného alebo užívateľom definovaného rozdelenia, pričom pôvodné rozdelenie je iteratívne optimalizované. Snáď najbežnejším nehierarchickým algoritmom je algoritmus k-priemerov (k-means). Tento algoritmus si bližšie popíšeme v nasledujúcej kapitole.

Algoritmy nehierarchických metód sú ľahšie implementovateľné a oproti hierarchickým metódam veľmi efektívne. Jedinou nevýhodou je, že predom musí byť pevne určený počet zhlukov. Záleží aj na počiatočnej inicializácii jednotlivých centier zhlukov, ktoré sa síce počas výpočtu menia, ale pri neideálnych dátach môžu byť výsledné pozorovania rozdielne.

V poslednej dobe bolo navrhnutých mnoho nových algoritmov, ktoré buď modifikujú tradičné metódy alebo sa vydávajú novými smermi. Z toho dôvodu vzniká terminologický problém. Takisto aj v prípade nehierarchických metód zhlukovej analýzy nie je jednoznačne určená ich klasifikácia. Je mnoho delení z rôznych

pohľadov, preto ponúkneme skôr prehľad skupín nehierarchických zhlukovacích metód, ktorá však nepredstavuje pevne stanovenú klasifikáciu týchto metód:

Jedno-priechodové metódy (Single-pass methods)

Ide o jednoduchú metódu, pri ktorej je súbor dát spracovávaný len raz. Nevýhodou tejto metódy je, že výsledkom môžu byť veľké zhluky a takisto to, že vytvorené zhluky sú závislé na poradí, v ktorom bol súbor dát spracovaný. Tento algoritmus je predovšetkým využívaný na zhlukovanie dokumentov, pričom na toto zhlukovanie sa využívajú aj hierarchické zhlukovacie metódy. Ich výhodou je, že stále sú výsledkom rovnaké zhluky.

Pri tomto zhlukovaní sa napočítava matica podobnosti, pričom miery podobnosti sú odlišné ako pri numerických zhlukovacích metódach. Uvedieme príklad z knihy [6] pre dva textové dokumenty, u ktorých je sledovaný výskyt piatich slov. Ak zaznamenáme jednotlivé početnosti výskytu sledovaných slov v stanovenom poradí ako vektor, môžu byť oba dokumenty charakterizované napríklad následujúcim spôsobom:

$$\mathbf{x}_1 = [2\,0\,1\,0\,3],$$

$$\mathbf{x}_2 = [0\,1\,0\,2\,0].$$

Vidíme, že žiadne slovo sa nevyskytuje v oboch dokumentoch súčasne, teda predpokladáme, že koeficient podobnosti sa rovná nule. Tomuto predpokladu vyhovuje napríklad kosínova miera, viď vzorec
Jednoduchý popis algoritmu môžeme zhrnúť do následujúcich krokov:

Algoritmus 3 Jedno-priechodové metódy

Vstup: Dáta $D_1, ..., D_n$ (*predpokladajme, že dané objekty sú napríklad dokumenty*)
Vstup: S_{max} (*prahová hodnota podobnosti dvoch objektov*)
1: $D_1 = C_1$ (D_1 *dokument bude reprezentovať prvý zhluk* C_1)
2: **for all** D_i, $i = 1, ..., n$ **do**
3: vypočítame podobnosť S k danému reprezentantovi existujúceho zhluku.
4: **if** podobnosť $S < S_{max}$ **then**
5: objekt D_i prehlásime reprezentantom nového zhluku.
6: **else**
7: prvok D_i priradíme existujúcemu zhluku a určíme centroid tohto zhluku (*v prvom kroku je to centroid zhluku pozostávajúceho z objektov* D_i *a* D_1)
8: **end if**
9: **end for**
Výstup: Objekty zaradené do zhlukov.

Metódy využívajúce premiestňovanie (Relocation methods)

Ide o metódy, ktoré optimálne určujú rozloženie a organizáciu objektov, ale vyžadujú inicializáciu počtu zhlukov. Priradenie k zhlukom je jednoznačné. Do tejto kategórie patrí algoritmus k - *priemerov*, ktorý podrobne opíšeme v Kapitole 2.

Táto skupina metód takisto zahŕňa jeho modifikácie, teda metódy *k-medoidov*,

k-modov a *k-histogramov*. Prvé dva algoritmy majú využitie pri dátových súboroch obsahujúcich len kvantitatívne premenné.

Metódy ***k-modov*** a ***k-histogramov*** [6] vychádzajú z toho, že každý zhluk je reprezentovaný m-rozmerným vektorom údajov, ktorý obsahuje buď modálne (najčastejšie zastúpené) kategórie jednotlivých premenných (metóda k-modov) alebo údaje o početnostiach kategórií jednotlivých premenných (metóda k-histogramov). Pritom sú využívané špeciálne miery nepodobnosti. Napríklad pri algoritme k-modov využívame *koeficient prostej zhody*, viď vzorec (1.5), prípadne od neho odvodenú mieru nepodobnosti. Vtedy je m-rozmerný vektor modálnych kategórií špeciálnym typom centroidu, obdobne ako vektor priemerov, či mediánov.

Algoritmus k-modov nie je vhodný pre asymetrické binárne dáta.

Fuzzy zhluková analýza

Ide o takzvané prekrývajúce sa zhlukovanie, teda v prípade metód tejto zhlukovej analýzy nedostávame disjunktné zhluky dát. Fuzzy zhluková analýza je omnoho obšírnejšia ako si ju predstavíme v následujúcom texte. Na príklade jedného z algoritmov si ukážeme výpočet *mier príslušnosti*, určujúcich príslušnosť prvku k danému zhluku.

Označme $\mathbf{x}_i$ ľubovoľný objekt a C_h zhluk. Označme u_{ih} mieru príslušnosti objektu $\mathbf{x}_i$, $i = 1, ..., n$ ku zhluku C_h, $h = 1, ..., k$. Táto miera je základným výstupom fuzzy zhlukovej analýzy. Musia platiť následujúce podmienky [6]:

1. $0 \leq u_{ih} \leq 1$ pre všetky $i = 1, ..., n$ a všetky $h = 1, ..., k$ (teda príslušnosť prvku nemôže byť negatívna)

2. $\sum_{h=1}^{k} u_{ih} = 1$ pre všetky $i = 1, ..., n$ (celková príslušnosť (membership) je súčtom jednotlivých mier príslušnosti)

Ak má každý objekt rovnakú mieru príslušnosti vo všetkých zhlukoch, hovoríme o *úplnom fuzzy zhlukovaní*. Na druhej strane, ak má každý objekt mieru príslušnosti 1 v niektorom zo zhlukov a nulovú mieru príslušnosti vo všetkých ostatných zhlukoch, hovoríme o *pevnom (disjunktnom) zhlukovaní*.

Uvedieme stanovenie týchto mier pomocou algoritmu *FANNY*, ktorý vyvinuli Kaufman and Rousseeuw. Podobne ako všetky algoritmy zhlukovania, *FANNY* má tiež svoje silné a slabé stránky, ktoré je potrebné brať do úvahy pri interpretácii získaných výsledkov.

FANNY je rozšírením zhlukovacej metódy fuzzy k-priemerov, a preto je nehierarchickým algoritmom, ktorý vyžaduje, aby bol vopred stanovený počet zhlukov. Okrem toho nevýhodou môže byť závislosť výsledného rozmiestnenia zhlukov a poradí spracovávaných prvkov.

Čo sa týka výhod tohto algoritmu, výstupom nie je len zoskupenie prvkov do rôznych zhlukov a stanovenie ich miery príslušnosti, ale aj informácie o šírke obrysového grafu (silhouette). O obrysových grafoch bližšie pojednávame v podkapitole 3.1.1. Tento graf nám poskytuje informáciu o kvalite zaradenia daného objektu a môžeme ho využiť aj pri iných metódach zhlukovej analýzy.

Hodnoty grafu sa pohybujú v intervale $\langle -1, 1\rangle$, pričom hodnoty blízke 1 ukazujú, že prvky majú najvhodnejšie zaradenie do zhluku, hodnoty blízke 0 ukazujú, že dané prvky môžu byť zaradené aj v inom zhluku a hodnoty blížiace sa zápornej krajnej hodnote indikujú nesprávne zaradenie.[7]

Algoritmus *FANNY* vychádza z matice nepodobností (viď miery nepodobnosti v podkapitole 1.1.1). Pozitívom je, že je použiteľný na číselné premenné (vtedy vychádza z matice vzdialeností) ako aj na kategoriálne premenné (vychádza z matice nepodobností), prípadne na kombináciu oboch.

Miery príslušnosti získame minimalizáciou funkcie

$$f = \sum_{h=1}^{k} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} u_{ih}^2 u_{jh}^2 D_{ij}}{2 \sum_{j=1}^{n} u_{jh}^2} \quad ,$$

kde D_{ij} je miera nepodobnosti (známa) a u_{ih}, u_{jh} sú miery príslušnosti (neznáme).

Pre ohodnotenie fuzzy zhlukovania sa počíta ***Dunnov koeficient rozkladu***

$$F(k) = \frac{1}{n} \sum_{i=1}^{n} \sum_{h=1}^{k} u_{ih} \quad ,$$

ktorý nadobúda hodnoty z intervalu $\langle \frac{1}{k}, 1\rangle$, k je počet zhlukov.

Rozlíšime dve situácie, popísané vyššie:

V prípade, že ide o *úplné fuzzy zhlukovanie* dostávame: všetky $u_{ih} = \frac{1}{k}$, $i = 1, ..., n$ a $h = 1, ..., k$. Potom

$$F(k) = nk\frac{1}{nk^2} = \frac{1}{k}$$

Druhou extrémnou možnosťou, ktorá bola spomenutá, bolo *pevné (disjunktné) zhlukovanie*, pre ktoré platí:
miera príslušnosti u_{ih} prvku $\mathbf{x}_i$, $i = 1, ..., n$ ku zhluku C_h, $h = 1, ..., k$ je rovná 1, pre $i = 1, ..., n$ sú všetky ostatné miery príslušnosti $u_{il} = 0$, kde $l = 1, ..., k$ a $l \neq h$. Potom

$$F(k) = \frac{n}{n} = 1$$

Normalizovaný tvar koeficientu rozkladu má tvar

$$F'_k = \frac{F_k - \frac{1}{k}}{1 - \frac{1}{k}} = \frac{kF_k - 1}{k - 1} \quad ,$$

ktorého hodnoty sa pohybujú v intervale $\langle 0, 1\rangle$.

2. Metódy k-priemerov a k-medoidov

Najznámejšou a najpoužívanejšou metódou v rámci nehierarchických zlukovacích metód je metóda k-priemerov. Algoritmus tejto metódy často nadväzuje na už vopred predspracované dáta. Teda často krát sa stretávame s kombináciou hierarchickej metódy a následného použitia metódy k- priemerov (k-means). Pozmenenou verziou tohto algoritmu je metóda k-medoids, ktorú si spolu s metódou k-priemerov (a jej modifikáciami) podrobne predstavíme v tejto kapitole. V oboch prípadoch ide o iteratívne metódy, ktoré vytvárajú disjunktné zhluky dát.

2.1 Metóda k-priemerov (k-means method)

Metóda k-priemerov je jednoduchým spôsobom, ako triediť daný súbor dát na určitý počet zhlukov (predpokladáme k zhlukov). Hlavnou myšlienkou je definovať k centroidov, teda jeden pre každý zhluk. *Centroid* je definovaný všeobecne ako vektor, pre ktorý platí, že súčet vzdialeností jednotlivých objektov v zhluku k tomuto vektoru je minimálny. Použitím euklidovskej vzdialenosti (1.1) je centroidom vektor priemerov a ide o metódu k-priemerov (pri použití manhattanskej vzdialenosti (1.3), je centroidom vektor mediánov a ide o metódu *k-mediánov*).

2.1.1 Popis algoritmu

Vstupným parametrom pre tento algoritmus je k, čo predstavuje počet zhlukov, do ktorých sa majú objekty nášho dátového súboru $\mathcal{X} \subset \mathbb{R}^m$ rozdeliť. Každý objekt je charakterizovaný hodnotami premenných, ktoré tvoria m-rozmerné vektory. Tieto zhluky sa majú vyznačovať vysokou podobnosťou v rámci objektov, ktoré ich tvoria a nepodobnosťou zhlukov navzájom. Podobnosť objektov, vyjadrená ako ich vzdialenosť, je počítaná vzhľadom k centru zhluku (centroidu). Treba poznamenať, že centrum zhluku nemusí byť tvorené dátovým objektom skúmaného súboru dát. Snahou je zaradiť objekty do zhlukov tak, aby bola minimalizovaná variabilita vnútri zhluku.

Nech množina n objektov $\{\mathbf{x}_i, i = 1, ..., n\}$ sú m-rozmerné vektory, kde x_{il} označuje hodnotu l-tej premennej i-tého objektu a c_{hl} predstavuje hodnotu l-tej premennej centroidu $\mathbf{c}_h$ zhluku C_h, $h = 1, ..., k$.

Vnútorná variabilita h-tého zhluku je potom definovaná [6] ako funkcia

$$S^{(h)} = \sum_{\mathbf{x}_i \in C_h} \sum_{l=1}^{m} (x_{il} - c_{hl})^2$$

a celková vnútrozhluková variabilita zhlukov (*total within cluster variance*) ako

$$S = \sum_{h=1}^{k} S^{(h)} = \sum_{h=1}^{k} \sum_{\mathbf{x}_i \in C_h} \sum_{l=1}^{m} (x_{il} - c_{hl})^2 \quad , \tag{2.1}$$

Cieľom je minimalizovať súčet štvorcov vzdialenosti objektov v zhlukoch od ich centroidov, a to cez k zhlukov, tj.

$$S^\star = min\{S\ \} \tag{2.2}$$

Algoritmus 4 Metóda k-priemerov

Vstup: Množina n objektov $\{\mathbf{x}_i,\ i = 1, ..., n\}$.
Vstup: k (*požadovaný počet zhlukov*)
1: Inicializácia $\mathbf{c}_h,\ h = 1, ..., k$ ($\boldsymbol{c}_h = [c_{h1}, ..., c_{hm}]$ *je centroid h-tého zhluku*)
2: **repeat**
3: **for all** $h \in \{1, ..., k\}$ **do**
4: **for all** $i \in \{1, ..., n\}$ **do**
5: Výpočet vzdialenosti objektu $\mathbf{x}_i$ od centroidov $\mathbf{c}_h$ (*vzdialenosť počítaná pomocou euklidovskej vzdialenosti, viď vzorec (1.1)*)
6: Priradenie prvku $\mathbf{x}_i$ k najbližšiemu centroidu $\mathbf{c}_h$ (*označme C_h h-tý zhluk s centrom $\boldsymbol{c}_h$*)
7: **end for**
8: Aktualizujeme centrá $\mathbf{c}_h$ novovzniknutých zhlukov C_h pomocou

$$\mathbf{c}_h = \frac{1}{n_{C_h}} \sum_{x_j \in C_h} x_j$$

(*n_{C_h} je počet objektov $\boldsymbol{x}_j$ priradených do zhluku C_h*)
9: **end for**
10: **until** je splnené optimalizačné kritérium 2.2, prípadne množina centroidov je nemenná
Výstup: priradenie objektov dátového súboru do k zhlukov

Je známe, že algoritmus k-priemerov je všeobecne v praxi veľmi využívaný hlavne kvôli rýchlosti a jednoduchosti. Problémom však môže byť to, že nezaručuje nájdenie globálneho optima, tj. často končí v niektorom lokálnom minime, čo nemusí byť vo všetkých prípadoch postačujúce.

Pri všeobecnom algoritme inicializujeme počiatočné centrá zhlukov náhodne. Preto môže niekedy dôjsť k situácii, že nedostaneme optimálne riešenie a je odporúčané, aby sa algoritmus aplikoval viackrát s rôznou počiatočnou inicializáciou. Jedným zo spôsobov, ktorý minimalizuje vplyv nevhodného počiatočného výberu centier je algoritmus k-priemerov^{++}. Tento algoritmus sa zameriava na optimálne, čo najviac rovnomerné, umiestnenie počiatočných centier medzi dátové objekty.

2.2 Algoritmus k-priemerov^{++}

Postup metódy pozostáva z dvoch častí. Prvou je inicializačný proces, ktorým sa metóda odlišuje od všeobecnej metódy k-priemerov a druhou časťou je samotný klasický algoritmus popísaný v kapitole 2.1.1. Optimalizáciu klasického algoritmu k-priemerov skúmali a analyzovali David Arthur a Sergei Vassilvitskii [4] zo Standfordskej univerzity. Ich experimenty ukázali, že tento spôsob inicializácie nielen zrýchľuje klasický algoritmus, ale prináša aj lepšie a presnejšie výsledky z hľadiska optimálneho riešenia.

Algoritmus 5 Metóda k-priemerov^{++}

Vstup: Množina n objektov $\mathcal{X} = \{\mathbf{x}_i,\ i = 1, ..., n\}$.
Vstup: k (*požadovaný počet zhlukov*)
1: Výber inicializačných centier pomocou inicializačného algoritmu (6)
2: Aplikácia klasického algoritmu k-priemerov (4) s vynechaním prvého kroku.
Výstup: priradenie objektov dátového súboru do k zhlukov

Výhodou tohoto algoritmu je možnosť zameniť druhú časť za akúkoľvek inú variantu algoritmu k-priemerov a prípadne docieliť ešte lepšiu rýchlosť.

2.2.1 Inicializačný algoritmus

Tento algoritmus bol vyvinutý na zvýšenie efektivity a presnosti zhlukovania pomocou metódy k-priemerov. Teraz sa pozrieme na spôsob inicializácie centier zhlukov.

Označme $D(x)$ ako minimálnu vzdialenosť prvku x od centra z množiny $\mathcal{C}$ doposiaľ vybraných centier. Vzdialenosť medzi objektom a centrom zhluku počítame pomocou euklidovskej vzdialenosti, viď vzorec (1.1). V priebehu algoritmu je postupne vytváraná množina $\mathcal{C}$. Vždy je vybraný objekt $x' \in \mathcal{X}$ s najvyššou hodnotou pravdepodobnosti

$$P(x') = \frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2} \tag{2.3}$$

Algoritmus 6 Inicializačný algoritmus metódy k-priemerov^{++}

Vstup: Množina n objektov $\mathcal{X} = \{\mathbf{x}_i,\ i = 1, ..., n\}$.
Vstup: k (*požadovaný počet zhlukov*)
1: Vyberieme náhodne centrum c_1 z množiny $\mathcal{X}$.
2: Vytvorme množinu $\mathcal{C} = \{c_1\}$ ($\mathcal{C}$ *je množina inicializačných centier*)
3: **for** $i = 2$ **to** k **do**
4: Vyberáme centrum $c_i = x'$, kde $x' = \arg\max_{x \in \mathcal{X}} P(x)$
5: $\mathcal{C} = \mathcal{C} \cup \{c_i\}$
6: **end for**
Výstup: $\mathcal{C}$ množina k vybraných inicializačných centier

2.3 Dvojfázový algoritmus k-priemerov

Ďalším problémom štandardného algoritmu k-priemerov je jeho citlivosť na odľahlé osamotené objekty v rámci dátového súboru. Tento problém rieši dvojfázový algoritmus k-priemerov, ktorý odhalí dané objekty. Ide vlastne o modifikovaný algoritmus, ktorý v prvej fáze využíva heuristiku: „*Ak je vkladaný objekt veľmi vzdialený od všetkých doterajších centier zhlukov, je následne prehlásený za centrum nového zhluku.*“ Autori článku [5] zistili, že výsledkom tejto fázy algoritmu je stav, kedy všetky objekty zhluku, sú buď odľahlé objekty alebo nie je odľahlý ani jeden. Teda dostávame presné zaradenie prvkov.

Ak vychádzame z definície zhlukovej analýzy, vieme, že objekty priradené do spoločného zhluku sú si podobné. Zhluk, ktorý je vytváraný počas prvej fázy tohto algoritmu však takéto vlastnosti nemá. Ide o zhluk, ktorý je označený ako tzv. zbytkový, pozostávajúci z pár prvkov značne sa odlišujúcich od ostatných zhlukov. Anglickým pomenovaním týchto prvkov je termín *outliers*. V praxi sa pri zhlukovaní tieto objekty buď zanedbávajú alebo je im priradená minimálna váha, čím neovplyvňujú zhlukovací proces. Pri vysokom počte dát je však tento spôsob dosť obtiažny a len ťažko by sme odhalili, ktoré prvky máme zanedbať. Preto bol vytvorený dvojfázový algoritmus, ktorý pomerne jednoducho tento prípad vyrieši.

Autori pritom upozorňujú na bočný efekt, ktorým je vyšší počet zhlukov vzniknutých v prvej fáze algoritmu ako je počet zhlukov štandardného algoritmu k-priemerov. Druhá fáza algoritmu sa zameriava na presné určenie odľahlých prvkov a nadväzuje na výsledné zhluky prvej fázy.

Ako vyplýva z názvu, algoritmus pozostáva z dvoch fáz:

- Modifikovaný proces metódy k-priemerov (Modified k-means process (MKP))
- Proces detekcie odľahlých prvkov (Outliers-finding process (OFP))

2.3.1 FÁZA 1: Modifikovaný proces algoritmu k-priemerov

Ako sme popisovali vyššie, je obtiažne nájsť odľahlé objekty vo veľkých dátových súboroch. Teraz si podrobnejšie zadefinujeme proces ich zhlukovania.
Prvá fáza algoritmu sa odlišuje tým, že nevytvára predom pevný počet zhlukov, ale ich konečný počet sa pohybuje v určitom rozmedzí.

Algoritmus začína rovnako ako štandardná metóda k-priemerov. Označme k' ako nastaviteľný počet zhlukov a položme $k' = k$. Inicializačnú množinu centier zhlukov $\mathcal{C} = \{\mathbf{c}_1, ..., \mathbf{c}_{k'}\} \in \mathbb{R}^n$ vytvoríme napríklad náhodným výberom k' objektov. V rámci iterácií nebudeme počítať len aktualizovanú polohu nového centra, ale zameriame sa aj na najkratšiu vzdialenosť medzi dvomi ľubovoľnými centrami zhlukov danú predpisom

$$D_{min}(\mathbf{c}_i, \mathbf{c}_h) = \min_{i,h=1,..,k',i\neq h} \{\sum_{j=1}^{n} (c_{ij} - c_{hj})^2\} \quad , \tag{2.4}$$

Pre každý objekt $\mathbf{x}_i$ vypočítame vzdialenosť k najbližšiemu centru zhluku vzťahom

$$D_{min}(\mathbf{x}_i, \mathbf{c}_h) = \min_{h=1,...,k';i=1,...,n} \{\sum_{j=1}^{n} (x_{ij} - c_{hj})^2\} \quad , \tag{2.5}$$

V niektorých prípadoch môže dôjsť k rozdeleniu dátového súboru do $k' = n$ zhlukov, preto sa definuje hodnota maximálne povoleného počtu zhlukov k_{max} a platí $k \leq k_{max} \leq n$.
Algoritmus pozostáva z následujúcich krokov:

Algoritmus 7 Modifikovaný proces algoritmu k-priemerov (MKP)

Vstup: Množina n objektov $\mathcal{X} = \{\mathbf{x}_i, i = 1, ..., n\}$.
Vstup: k', k_{max} (*nastaviteľný počet zhlukov*)
1: Náhodne určíme k' centier zhlukov (*položme na začiatku* $k' = k$)
2: **repeat**
3: Vypočítame $D_{min}(\mathbf{c}_j, \mathbf{c}_h)$ (*viď vzorec (2.4)*)
4: **for** $i = 1$ **to** n **do**
5: Vypočítame $D_{min}(\mathbf{x}_i, \mathbf{c}_h)$ (*viď vzorec (2.5)*)
6: **if** $D_{min}(\mathbf{x}_i, \mathbf{c}_h) > D_{min}(\mathbf{c}_j, \mathbf{c}_h)$ **then**
7: $k' = k' + 1$
8: $\mathbf{x}_i = \mathbf{c}_{k'}$ (x_i *je centrom nového zhluku* $C_{k'}$)
9: **if** $k' > k_{max}$ **then**
10: dva najbližšie zhluky C_{k1} a C_{k2} sa spájajú do jedného zhluku C_k^*
11: $k' = k_{max}$
12: **end if**
13: **else**
14: Priradíme $\mathbf{x}_i$ do najbližšieho zhluku.
15: **end if**
16: **end for**
17: **until** dosiahneme predom daného počtu iterácií alebo je splnené kritérium pre zastavenie (stabilizujú sa centrá zhlukov, tj. ich pozície sa v podstate nemenia)
Výstup: k' centier spolu s rozdelením dátového súboru $\mathcal{X}$ do k' zhlukov.

2.3.2 FÁZA 2: Proces detekcie odľahlých prvkov

V tejto fáze algoritmu nájdeme odľahlé prvky a následne rozdelíme objekty do požadovaných k zhlukov. Môžeme použiť niektorú z hierarchických zhlukovacích metód, popísané v kapitole 1.2.1), ktoré na základe veľkej vzdialenosti odhalia odľahlé prvky. K nájdeniu odľahlých prvkov je vhodná zhlukovacia metóda založená na princípe minimálnej kostry. Keďže je celková časová zložitosť hierarchických metód vyššia ako pri metóde založenej na konštrukcii minimálnej kostry, budeme sa v tejto časti zaoberať druhou možnosťou detekcie. Potrebné základné pojmy z teórie grafov sú definované v prílohe A.

Z prvej fázy sme obdržali k' centier zhlukov, ktoré budeme považovať za k' uzlov. Z nich sa stanú vrcholy úplného ohodnoteného grafu nazývaného *strom* a budú využité pri konštrukcii minimálnej kostry. Každá hrana daného grafu je ohodnotená vzdialenosťou daných dvoch centier, ktoré spája.
Vstupom algoritmu je množina $\mathcal{C} = \{\mathbf{c}_1, ..., \mathbf{c}_{k'}\} \in \mathbb{R}^n$ centrier zhlukov obdržaných v prvej fáze. Nech les F je prázdny.

Pripomenieme, že platí vzťah

$$k \leq k' \leq k_{max} \leq n \quad ,$$

kde n je počet objektov dátového súboru a hodnoty k', k, k_{max} sme definovali v 1. fáze algoritmu.

Algoritmus 8 Proces detekcie odľahlých prvkov

Vstup: Množina $\mathcal{C}$ centrier zhlukov (*ich počet je* k')
Vstup: Nech les F je prázdny.
Vstup: k (*požadovaný počet zhlukov*)
1: Vytvoríme minimálnu kostru z prvkov množiny $\mathcal{C}$ (*vznikol strom*)
2: Vytvorený strom vložíme do lesa F
3: **repeat**
4: Zrušíme hranu s najväčším ohodnotením v rámci všetkých stromov v lese F a nahradíme pôvodný strom obsahujúci danú hranu dvoma novovzniknutými podstromami
5: Objekty z podstromov s veľmi malým počtom uzlov označíme za odľahlé (*prípadne ich vylúčime z ďalšieho spracovávania*)
6: **until** počet stromov v lese F je rovný hodnote k
Výstup: rozdelenie dátového súboru $\mathcal{X}$ do k zhlukov, prípadne množina odľahlých objektov

---

2.4 Metóda k-medoidov (k-medoids method)

Metóda k-medoidov patrí do skupiny nehierarchických metód a taktiež vytvára v rámci sledovaných dát k disjunktných zhlukov, ktoré vzniknú z n objektov skúmaného dátového súboru $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbb{R}^d$. Rovnako ako metóda k-priemerov je určený pre kvantitatívne premenné a vychádza z počiatočného rozdelenia objektov do k zhlukov. Ako centrá zhlukov berieme priamo objekty $\mathbf{m}_i \in \mathcal{X}, i = 1, ..., k$, nazývané *medoidy*. *Medoid* môžeme definovať ako konkrétny objekt súboru dát, ktorý sa vyznačuje tým, že jeho pozícia je vzhľadom ku všetkým objektom daného súboru dát najviac centralizovaná.

2.4.1 Popis algoritmu

Algoritmus k-medoidov, v anglickej literatúre ho nájdeme pod názvom *Partitioning Around Medoids (PAM) algorithm*, bol navrhnutý Kaufmanom a Rousseeuwom. Ich metóda je obzvlášť výhodná, ak chceme z náhodných dát získať reprezentatívne objekty, charakterizujúce jednotlivé zhluky. To môže byť veľmi vhodné pri následnej interpretácii výsledkov.

Algoritmus PAM môžeme rozdeliť do dvoch fáz:

1. Fáza „budovania“ (building phase)
2. Fáza „výmeny“ (swap phase)

Postup prvej fázy

Prvá fáza spočíva v prvotnom rozklade dátového súboru $\mathcal{X}$. Rozklad súboru $\mathcal{X}$ je v priebehu fázy budovania uskutočnený postupným výberom k reprezentatívnych objektov. Najprv vyberieme počiatočný medoid, ktorý je určený tak, aby súčet vzdialeností jednotlivých objektov od tohoto vybraného bol minimálny. Pre sprehľadnenie následujúceho pseudokódu vysvetlíme voľbu počiatočného medoidu vopred.

Voľba počiatočného medoidu

Počiatočným medoidom budeme rozumieť vektor $\mathbf{m}_1 \in \mathcal{X}$, ktorý môže byť zároveň centroidom $\bar{\mathbf{x}}$ dátovej matice $\mathbf{X}$. V prípade, že nespĺňa tieto vlastnosti, to znamená, že centroid dátovej matice nie je jej prvkom, potom za počiatočný medoid volíme prvok, pre ktorý platí:

$$\mathbf{m}_1 = \arg\min_{\mathbf{m}_i \in \mathcal{X}} \{D(\mathbf{m}_i, \bar{\mathbf{x}})\} \quad , \tag{2.6}$$

kde $D(\mathbf{m}_i, \bar{\mathbf{x}})$ predstavuje vzdialenosť objektu $\mathbf{m}_i$ od centroidu $\bar{\mathbf{x}}$ vypočítanú podľa niektorej zo vzdialeností definovaných v podkapitole 1.1.1.

Algoritmus 9 Metóda k-medoidov, fáza 1

Vstup: Dátová matica $\mathbf{X}$

Vstup: k (*požadovaný počet zhlukov*)

1: Určenie počiatočného medoidu $\mathbf{m}_1$ podľa vzťahu 2.6
2: $\mathcal{M} = \{\mathbf{m}_1\}$ (*množina doposiaľ vybraných medoidov*)
3: $pocet = 1$ (*počet vybraných medoidov*)
4: **repeat**
5: **for** $i, j \in \{2, ..., n\}$, $j \neq i$ **do**
6: Vypočítame hodnotu minimálnej vzdialenosti $D(\mathbf{m}_l, \mathbf{x}_i)$, $\mathbf{m}_l \in \mathcal{M}$
7: Vypočítame hodnoty vzdialeností $D(\mathbf{x}_j, \mathbf{x}_i)$
8: Určíme veličinu $Z_j = \sum_i C_{ij}$, kde $C_{ij} = |D(\mathbf{m}_l, \mathbf{x}_i) - D(\mathbf{x}_j, \mathbf{x}_i)|$ (Z_j *je celkový zisk, ktorý dosiahneme tým, že zaradíme objekt* $\boldsymbol{x}_j$ *do* $\mathcal{M}$)
9: Vyberieme prvok $\mathbf{x}_j$, ktorého celkový zisk je maximálny a prehlásime ho za ďalší z vybraných medoidov $\mathbf{m}_j$
10: $\mathcal{M} \cup \{\mathbf{m}_j\}$
11: $pocet = pocet + 1$
12: **end for**
13: **until** $pocet = k$

Výstup: Množina $\mathcal{M} = [\mathbf{m}_1, ..., \mathbf{m}_k]$

Postup druhej fázy

Po určení k reprezentatívnych objektov začína druhá fáza. V nej sa snažíme zlepšiť pôvodný rozklad dátovej matice $\mathbf{X}$. Chceme teda nájsť podmnožinu $\{\mathbf{m}_1, ..., \mathbf{m}_k\} \subset \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, aby bolo dosiahnuté minimum funkcie

$$\sum_{i=1}^{n} D(\mathbf{m}_{li}, \mathbf{x}_i) \quad , \tag{2.7}$$

kde $\mathbf{m}_{li}$ predstavuje medoid l-tého zhluku, ku ktorému je priradený i-tý objekt, viď [6] strana 86.

V tejto fáze teda budeme zisťovať, aký vplyv na (2.7) má výmena prvku $\mathbf{x}_i \in \mathcal{M}$ za prvok $\mathbf{x}_h$, ktorý nie je súčasťou množiny $\mathcal{M}$.

Efekt výmeny $\mathbf{x}_i$ za objekt $\mathbf{x}_h$, vzhľadom k doposiaľ nevybranému prvku $\mathbf{x}_j$, označíme C_{jih}. Výpočet C_{jih} môžeme nájsť v popise algoritmu 10, pretože závisí od viacerých skutočností.

Algoritmus 10 Metóda k-medoidov, fáza 2

Vstup: Množina $\mathcal{M} = [\mathbf{m}_1, ..., \mathbf{m}_k]$

repeat
  for all $j, h \in \{1, ..., n\}$ **and** $i \in \{1, ..., k\}$ **do**
    if $(D(\mathbf{x}_j, \mathbf{m}_l) < D(\mathbf{x}_j, \mathbf{x}_i)$ **and** $D(\mathbf{x}_j, \mathbf{m}_l) < D(\mathbf{x}_j, \mathbf{x}_h),\ \mathbf{m}_l \neq \mathbf{x}_i, \mathbf{x}_h)$ **then**
      $C_{jih} = 0$
    else if $D(\mathbf{x}_j, \mathbf{x}_i) < D(\mathbf{x}_j, \mathbf{m}_l)$ **then**
      if $D(\mathbf{x}_j, \mathbf{x}_i) < D(\mathbf{x}_j, \mathbf{m}_w), \mathbf{m}_w \in \mathcal{M} \setminus \{\mathbf{m}_l\}$ ($\boldsymbol{m}_w$ *je druhý najbližší medián*) **then**
        $C_{jih} = D(\mathbf{x}_j, \mathbf{x}_h) - D(\mathbf{x}_j, \mathbf{x}_i)$
      else
        $C_{jih} = D(\mathbf{x}_j, \mathbf{m}_w) - D(\mathbf{x}_j, \mathbf{m}_l)$
      end if
    else if $D(\mathbf{x}_j, \mathbf{m}_l) < D(\mathbf{x}_j, \mathbf{x}_i)$**and** $D(\mathbf{x}_j, \mathbf{x}_h) < D(\mathbf{x}_j, \mathbf{m}_l)$ **then**
      $C_{jih} = D(\mathbf{x}_j, \mathbf{x}_h) - D(\mathbf{x}_j, \mathbf{m}_l)$
    end if
  end for
  for all $i \in \{1, ..., k\}$ **and** $h \in \{1, ..., n\}$ **do**

$$T_{ih} = \sum_j C_{jih}$$

  (T_{ih} *je vyjadrením celkového efektu*)
  end for
  if $\min T_{ih} < 0$ **then**
    Vymeníme prvky $\mathbf{x}_i$ a $\mathbf{x}_h$
  end if
until $\min T_{ih} \geq 0$

Výstup: aktualizovaná množina $\mathcal{M}$, priradenie objektov dátového súboru do k zhlukov

3. Praktické využitie metódy k-priemerov

Praktickú časť aplikácie algoritmu k-priemerov sme sa rozhodli robiť pomocou matematického programu Matlab, ktorý poskytuje užívateľsky príjemne prostredie pre prácu s dátovými súbormi.

3.1 Simulované dáta

V prvej časti sa zameráme na aplikáciu algoritmu na simulované dáta. Zvolili sme generovanie dát z dvojrozmerného normálneho rozdelenia s vektorom stredných hodnôt $\mu = (\mu_1, \mu_2)^T$ a kovariančnou maticou $\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$, ktoré je definované hustotou

$$f(x,y) = \frac{1}{2\pi\sqrt{\sigma_1^2\sigma_2^2(1-\rho^2)}}$$

$$\times \exp\{-\frac{1}{2(1-\rho^2)}\{\frac{(x-\mu_1)^2}{\sigma_1^2} - 2\rho\frac{x-\mu_1}{\sigma_1}\frac{y-\mu_2}{\sigma_2} + \frac{(y-\mu_2)^2}{\sigma_2^2}\}\}, \quad (x,y) \in R^2$$

Ukážka kódu 3.1 ukazuje vytvorenie jedného zhluku, pozostávajúceho z 500 objektov generovaných pomocou dvojrozmerného normálneho rozdelenia so strednou hodnotou $\mu = (4.5, 5.9)$ a kovariančnou maticou $\Sigma = \begin{pmatrix} 0.9 & 0.5 \\ 0.5 & 0.9 \end{pmatrix}$ v programe Matlab spolu s vykreslením bodov v dvojrozmernom priestore.

```
1 mi = [4.5, 5.9]; Sigma = [0.9 0.5; 0.5 0.9];
        r2 = mvnrnd(mi, Sigma, 500);
3 plot(r2(:,1),r2(:,2),'c.');
```

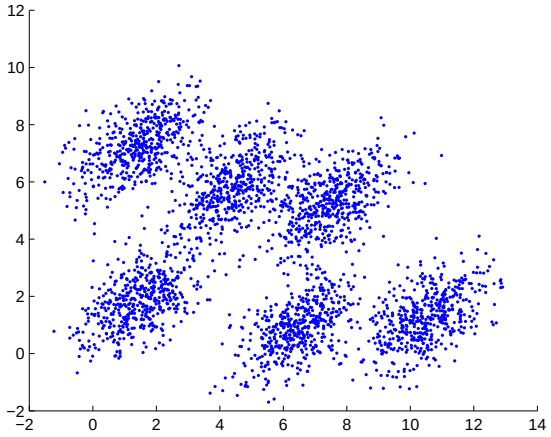
Ukážka kódu 3.1: Generovanie zhlukov

Dáta sme generovali s cieľom vytvoriť v dvojrozmernom priestore pár graficky viditeľných zhlukov, viď obrázok 3.1 a pomocou grafických výstupov sledovať, ako sa algoritmus správa pri zmene niektorých parametrov. Ako vidíme na obrázku, vytvorili sme 6 zhlukov, ktorých hraničné objekty sa v niektorých miestach prekrývajú. Pozrieme sa na to, ako sa líši priradenie objektov do zhlukov pri zmene miery vzdialenosti (viď kapitola 1.1.1).

3.1.1 Aplikácia algoritmu k-priemerov

Implementácia použitej funkcie

Pre daný experiment bol vytvorený cyklus, ktorý aplikoval algoritmus k-priemerov pre rôzne k, $k \in \{1, ..., 8\}$ na generované dáta, viď ukážka kódu 3.2. V cykle sme využili funkciu kmeans, ktorú ponúka program Matlab. Ukážka kódu zobrazuje použitie kmeans na dátovú maticu r s daným počtom požadovaných zhlukov k.



Obr. 3.1: Nagenerované dvojrozmerné dáta

Ako miera vzdialenosti je v ukážke určená euklidovská vzdialenosť. Parameter *replicates* udáva počet rôznych inicializačných rozmiestnení k zhlukov. V našom prípade sa funkcia k-priemerov spustí 100 krát pre dané k. Pri behu funkcie sme na konci každej iterácie vykreslili stavy výpočtov, pričom v premennej rC sme si uchovávali súradnice jednotlivých centroidov. Súradnice centroidov boli ďalším výstupom implementovanej funkcie, ktoré sme nechali vykresliť do grafického výstupu spolu s farebne odlíšenými zhlukmi. Na výpočet vzdialenosti sme v prvom prípade použili euklidovskú vzdialenosť, viď vzorec (1.1) a jeho grafický výstup sme porovnali s grafickým výstupom pri použití manhattanskej vzdialenosti, viď vzorec (1.3).

```
ptsymb = {'bs','r^','md','go','c+','yo','m.','rx'};
for k=1:8
    [rIDs, rC]=kmeans(r, k,'Distance','sqeuclidean','replicates'
        ,100)
    gplotmatrix(r, r, rIDs);

   for x=1:k
    s = sprintf('%f %f',rC(x,1), rC(x,2));
    disp(s);
   end
   disp(' ');

    for i = 1:k
        clust = find(rIDs==i);
        plot(r(clust,1),r(clust,2),ptsymb{i});
        hold on;
    end
    plot(rC(:,1),rC(:,2),'ko', 'MarkerSize', 14,  'LineWidth',2);
    plot(rC(:,1),rC(:,2),'kx', 'MarkerSize', 14,  'LineWidth',2);
    grid on
    pause;
end
```

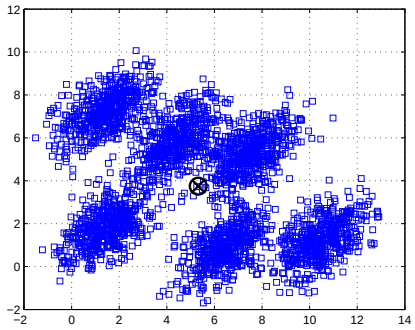
Ukážka kódu 3.2: Implementácia použitej funkcie

Výsledky predvádzaných výpočtov

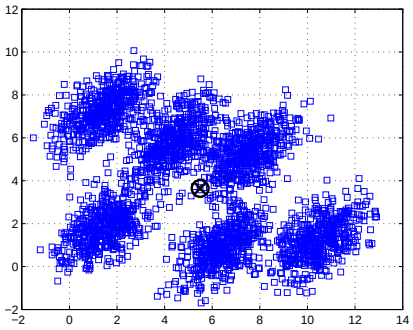
Grafické výstupy centier zhlukov a rozdelenie objektov do samotných zhlukov boli následovné:

1. iterácia: Vidíme, že dátový súbor v oboch prípadoch tvorí jeden zhluk, pretože $k = 1$. Súradnice ich centroidov sú však odlišné, ako ukazuje následujúca tabuľka.

	D_E	D_B
rC_1	[5.319214, 3.746849]	[5.492739, 3.642884]



Obr. 3.2: euklidovská vzdialenosť

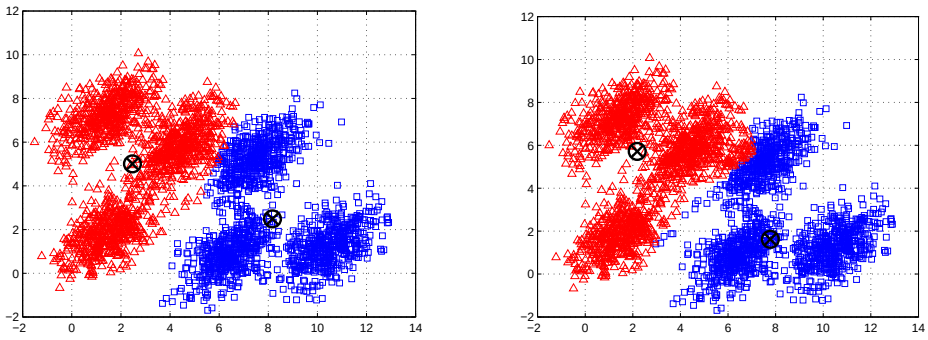


Obr. 3.3: manhattanská vzdialenosť

Ďalšie grafy uvádzame stále analogicky s prvou iteráciou, teda naľavo graf s použitím euklidovskej metriky a napravo graf pre manhattanskú vzdialenosť.

2. iterácia: Pre $k = 2$ sme dostali dva zhluky, ktoré sa opäť odlišujú ako ukazuje tabuľka centroidov. Pripomíname, že centroidy zhlukov nemusia byť súčasťou dátového súboru.

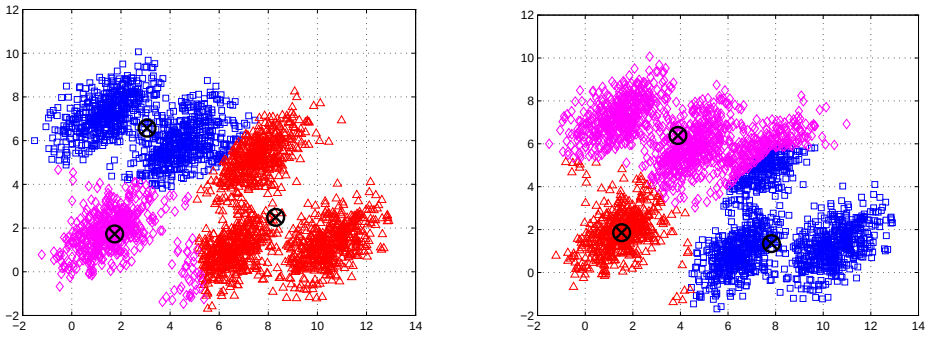
	D_E	D_B
rC_1	[8.172932, 2.488015]	[7.776040, 1.619744]
rC_2	[2.469299, 5.004006]	[2.183861, 5.713812]



Obr. 3.4: Rozdelenie na dva zhluky pri daných vzdialenostiach

3. iterácia: Tentokrát pre predom určené $k = 3$ už môžeme pozorovať značné rozdiely v priradení prvkov do jednotlivých zhlukov. Opäť pre porovnanie, súradnice centroidov sa nachádzajú v tabuľke nižšie, pričom centroidy zdanlivo rovnakých zhlukov sú uvádzané v rovnakom riadku.

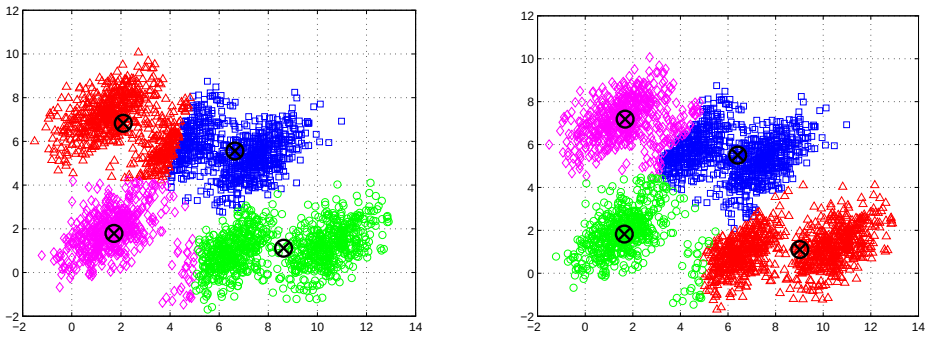
	D_E	D_B
rC_1	[3.064176, 6.569619]	[3.894394, 6.387330]
rC_2	[8.305299, 2.486924]	[7.824595, 1.348487]
rC_3	[1.744527, 1.728845]	[1.529269, 1.856788]



Obr. 3.5: Tri zhluky pri euklidovskej a manhattanskej vzdialenosti

4. iterácia: V tomto prípade, kedy $k = 4$, sú rozdelenia objektov do zhlukov na obrázkoch skoro totožné. Vidno niekoľko objektov na hraniciach zhlukov, ktoré sú zaradené rôzne.
Súradnice centroidov:

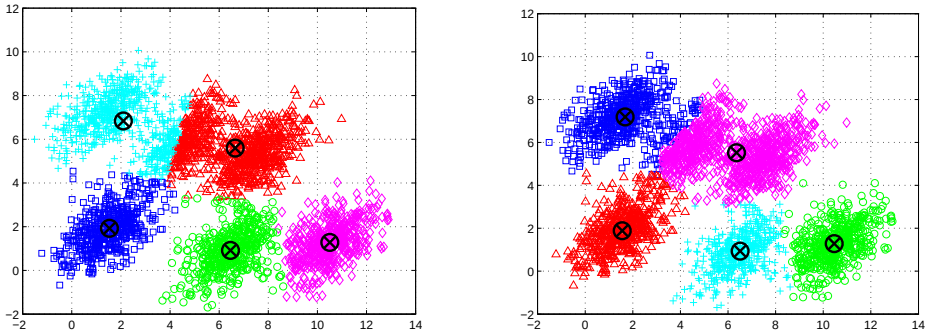
	D_E	D_B
rC_1	[6.641229, 5.560357]	[6.406809, 5.505064]
rC_2	[2.095862, 6.825713]	[1.678904, 7.185717]
rC_3	[1.718911, 1.787909]	[1.642511, 1.830500]
rC_4	[8.623260, 1.108283]	[9.017823, 1.105554]



Obr. 3.6: Rozdelenie na štyri zhluky pri euklidovskej a manhattanskej vzdialenosti

5. iterácia: Vidíme, že na obrázkoch je rozdielne najmä rozdelenie horných splývajúcich „troch“ zhlukov, do dvoch zhlukov. Celkovo sa však rozloženie centroidov skoro zhoduje.

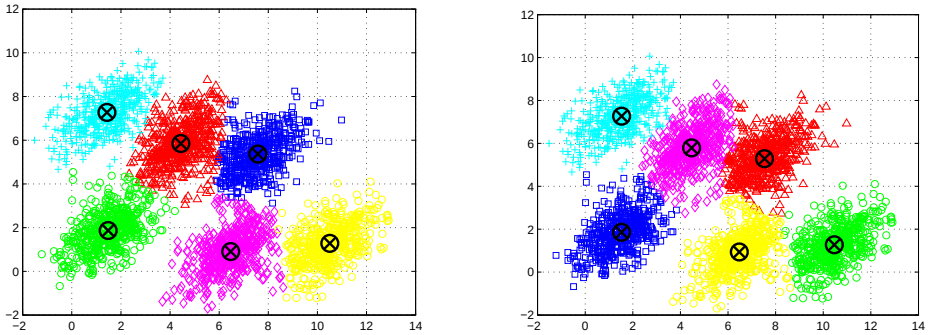
	D_E	D_B
rC_1	[1.533061, 1.922236]	[1.557759, 1.883460]
rC_2	[6.655655, 5.597244]	[6.356996, 5.518841]
rC_3	[10.506313, 1.279960]	[10.461705, 1.284008]
rC_4	[6.463715, 0.911274]	[6.509043, 0.936033]
rC_5	[2.101606, 6.836357]	[1.678706, 7.188527]



Obr. 3.7: Rozdelenie do 5-tich zhlukov pri daných vzdialenostiach

6. iterácia: Vidíme, že aj v tomto prípade je trochu rozdiel v priradení objektov do zhlukov.

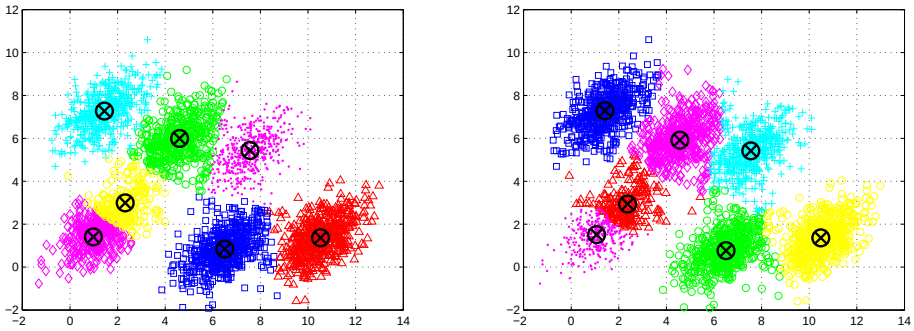
	D_E	D_B
rC_1	[7.577947, 5.367653]	[7.533426, 5.285296]
rC_2	[4.442112, 5.848289]	[4.472114, 5.790735]
rC_3	[6.468563, 0.902947]	[6.476683, 0.939083]
rC_4	[1.480049, 1.866707]	[1.526487, 1.856788]
rC_5	[1.430746, 7.269655]	[1.530887, 7.266252]
rC_6	[10.506313, 1.279960]	[10.463303, 1.273382]



Obr. 3.8: Rozdelenie do 6-tich zhlukov pri daných vzdialenostiach

7. iterácia: V tomto prípade je počet požadovaných zhlukov nadhodnotený. Vidíme, že niektorý zo zhlukov je rozdelený na dva menšie.

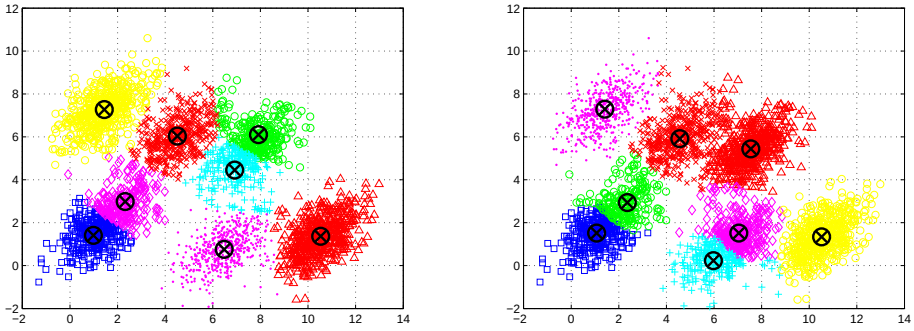
	D_E	D_B
rC_1	[6.5058 0.8416]	[6.5050 0.7647]
rC_2	[10.5263 1.3675]	[10.4865 1.3616]
rC_3	[0.9826 1.4072]	[1.0688 1.5147]
rC_4	[4.6069 5.9945]	[4.5653 5.9135]
rC_5	[1.4483 7.2711]	[1.4132 7.2925]
rC_6	[2.3211 2.9949]	[2.3696 2.9354]
rC_7	[7.5633 5.4380]	[7.5453 5.4189]



Obr. 3.9: Rozdelenie do 7-mich zhlukov pri daných vzdialenostiach

8. iterácia: Po aplikácii k-priemerov s rôznymi mierami vzdialeností vidíme, že sa rozdelenie líši podstatne, najmä výberom zhlukov, ktoré rozdelí na menšie. V tomto prípade uvedieme len hodnoty 4 centroidov vzniknutých menších zhlukov. Ostatné hodnoty centroidov sú skoro totožné s predchádzajúcim umiestnením centroidov rovnakých zhlukov.

	D_E	D_B
rC_1	[0.9826, 1.4072]	[1.068831 1.514656]
rC_2	[2.3190, 2.9857]	[2.359467 2.935425]
rC_3	[6.9193, 4.4518]	[5.972957 0.228027]
rC_4	[7.9207, 6.1010]	[7.042229 1.516902]



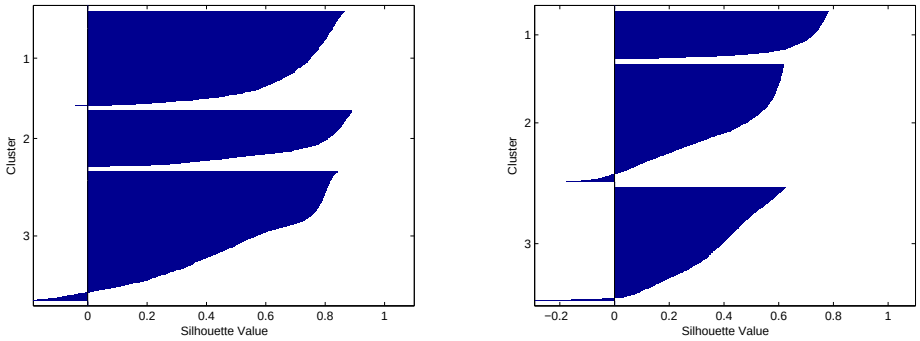
Obr. 3.10: Rozdelenie do 8-mich zhlukov pri daných vzdialenostiach

Veľmi výhodným pri určovaní kvality zaradenia daných objektov do určitého počtu zhlukov je *obrysový graf* (angl. silhouette). Obrysové grafy vyjadrujú miery príslušnosti jednotlivých objektov ku zhlukom (viď Fuzzy zhluková analýza v podkapitole 1.2.2), pričom hodnoty sa pohybujú v rozmedzí od -1 do 1. Hodnota miery príslušnosti (silhouette value) $s(i)$ pre každý objekt vyjadruje podobnosť objektu s objektmi v spoločnom zhluku v porovnaní s ostatnými objektmi iných zhlukov. Je definovaná vzťahom

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

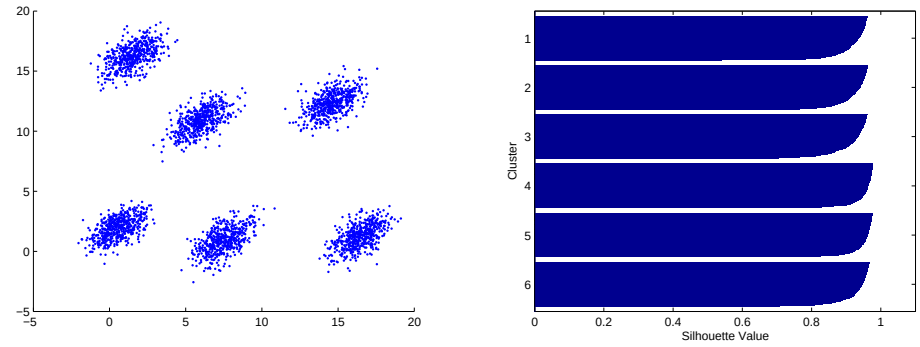
kde $a(i) = \frac{1}{n_A} \sum_{a \in A} d(a, i)$ je priemerná vzdialenosť objektu i od ostatných objektov a spoločného zhluku A a $b(i) = \min\{\frac{1}{n_C} \sum_{c \in C} d(c, i);\ C \neq A\}$ predstavuje najmenšiu priemernú vzdialenosť objektu i od objektov c iného zhluku C.

Uvedieme pár príkladov obrysových grafov na generovaných dátach z predchádzajúcej aplikácie. Napríklad pre $k = 3$ pri danom dátovom súbore s použitím daných vzdialeností dostávame:



Obr. 3.11: Obrysové grafy pri daných mierach vzdialeností pre $k = 3$

Vieme, že dátový súbor pozostáva zo 6 zhlukov a požadovaný počet centroidov sme v tomto prípade podhodnotili. Vidíme, že hodnoty mier príslušnosti v jednotlivých zhlukoch nedosahujú ani krajnú hodnotu 1. Ak by sa približovali krajnej hodnote, išlo by o pevné disjunktné zhlukovanie, viď obrázok 3.12

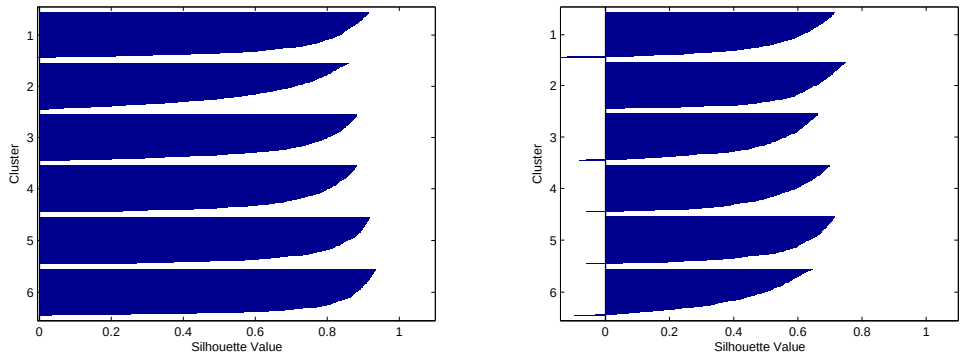


Obr. 3.12: Obrysový graf disjunktných zhlukov pri euklidovskej vzdialenosti pre $k = 6$

Miery príslušnosti blízke hodnote 0 vyjadrujú, že objekty môžu byť zaradené do iného zhluku, teda priradenie nie je jednoznačné. Ako vidíme (obrázok 3.11), v prípade mannhattanskej vzdialenosti sú miery príslušnosti menšie v porovnaní s obrysovým grafom euklidovskej vzdialenosti, teda pri danej voľbe počtu zhlukov sú výsledky ťažšie interpretova teľné. Šírka daných obrysov je takisto rozdielna, čo môže byť ďalším ukazovateľom nesprávneho zaradenia.

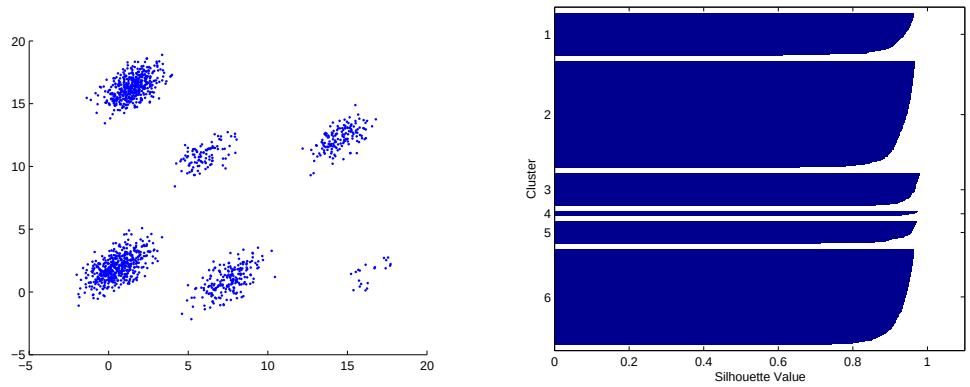
Záporné hodnoty indikujú nesprávne zaradenie daných objektov. V tomto prípade môžu predstavovať nesprávnu voľbu počtu zhlukov. Ako ukazuje nasledujúce porovnanie (obrázok 3.13), pri voľbe $k = 6$ sa hodnoty mier približujú hodnote 1, čo predstavuje skoro jednoznačné priradenie objektov do zhlukov. Analogicky pri voľbe mannhattanskej vzdialenosti sú miery príslušnosti menšie, dokonca v niektorých prípadoch je hodnota miery príslušnosti záporná.

Rovnaká šírka daných obrysov je vyjadrením toho, že vzniknuté zhluky ob-



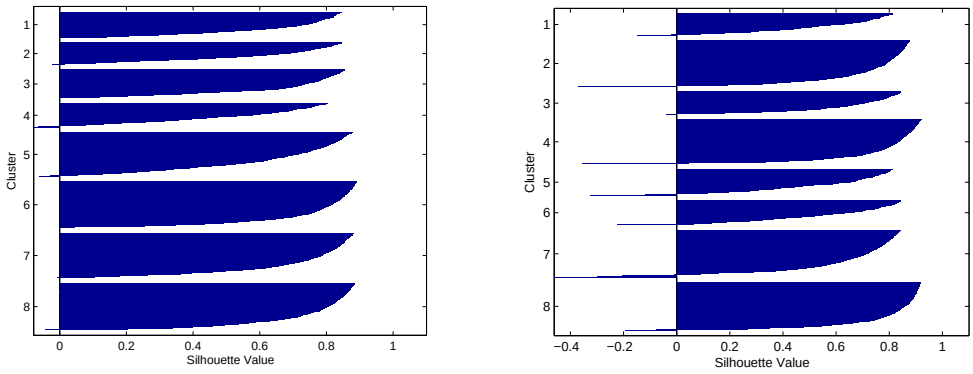
Obr. 3.13: Obrysové grafy pri daných mierach vzdialeností pre $k = 6$

sahujú približne rovnaký počet objektov. V našom prípade vieme, že dáta boli generované z rovnakých zhlukov, ktoré boli náhodne posunuté a v prípade voľby k by sme za kvalitné a správne rozdelenie vybrali obrysový graf, ktorý má nielen nadpriemerné hodnoty mier príslušnosti, ale aj približne rovnaké šírky obrysov, čo potvrdzuje obrázok 3.13. Všeobecne však neplatí, že každé optimálne rozdelenie do zhlukov by malo pozostávať zo zhlukov s rovnakým počtom objektov. Príkladom sú následujúce disjunktné dáta s rôznym počtom objektov v zhluku:



Obr. 3.14: Obrysové grafy pri daných mierach vzdialeností pre $k = 6$

V prípade, keď sme daný počet k nadhodnotili na $k = 8$, pozorovali sme rozdelenie dvoch generovaných zhlukov na dva menšie. Z obrysového grafu na obrázku 3.15 jasne vidíme, že vzniknuté zhluky sú úzke a ich miery príslušnosti sú nižšie v obidvoch prípadoch.



Obr. 3.15: Obrysové grafy pri daných mierach vzdialeností pre $k = 6$

3.1.2 Aplikácia algoritmu k-priemerov^{++}

Metóda k-priemerov, ktorá je súčasťou množiny v Matlabe implementovaných funkcií, ponúka ako svoju súčasť funkciu *replicates*, ktorú sme využívali pri dátach v predošlej aplikácii algoritmu. Podstatou tejto funkcie je, že v nami určenom počte opakovaní určí množinu inicializačných centrier a z nich vyberie tie, ktorých účelová funkcia S, viď vzorec (2.1), je minimálna.

V kapitole 2 sme predstavili algoritmus k-priemerov^{++} (viď Algoritmus 5), ktorý je určený na eliminovanie vplyvu nevhodného výberu inicializačných centier. V nasledujúcich príkladoch si ukážeme ako tento algoritmus pracuje na generovaných dátach.

Implementácia algoritmu k-priemerov^{++}

Algoritmus tvoria dve funkcie. Prvá vytvára množinu inicializačných centier a do prostredia Matlabu bola implementovaná iným užívateľom. Zdrojový kód je možné stiahnuť na stránke [8]. Druhá funkcia rozdeľuje objekty do zhlukov a ide o samotný algoritmus k-priemerov, ktorý vychádza z množiny inicializačných centier prvej funkcie.

Algoritmus užívateľa predstavoval kompletnú implementáciu algoritmu k-priemerov^{++}, z ktorej sme vyňali len inicializačnú fázu. Pôvodný algoritmus bol implementovaný tak, že oproti štandardnej implementácii algoritmu k-priemerov v Matlab-e, vyžadoval na vstupe transponovanú maticu dát. Preto sa v kóde objavuje pri vstupe a výstupe inicializačnej fázy transponovanie vstupnej a výstupnej matice. Uvedieme aj zdrojový kód (viď Ukážka kódu 3.3), pomocou ktorého sme dospeli k vytvoreniu množiny inicializačných centier. Množina je označená premennou C. Dátová matica je označená premennou X a premennou D je označená

matica vzdialeností medzi objektami z X a najbližším centrom z množiny C. Ostatné premenné sú len pomocné.

```
function [C] = inicializacia(X,k)

C = X(:,1+round(rand*(size(X,2)-1)));
temp = ones(1,size(X,2));
for i = 2:k
    D = X-C(:,temp);
    D = sqrt(dot(D,D));
    C(:,i) = X(:,find(rand < cumsum(D)/sum(D),1));
    [tmp,temp] = max(bsxfun(@minus,2*real(C'*X),dot(C,C).'));
end

C=C';
```

Ukážka kódu 3.3: Inicializačný algoritmus

Ukážka kódu 3.4 zobrazuje implementáciu algoritmu k-priemerov^{++}. Výstupom z našej implementácie je grafické znázornenie rozdelenia inicializačných centier prvej fázy spolu s výpisom ich súradníc. Ďalším výstupom je farebne rozlíšené znázornenie rozdelenia objektov do zhlukov a opätovný výpis súradníc inicializačných centier, pri ktorom sme zistili, že inicializačné centrá sú vyberané tak optimálne, že centrá zhlukov v druhej fáze algoritmu sa líšia len v minimálnom počte prípadov a to dokonca len v stotinách alebo tisícinách.

Inicializačný algoritmus je vhodný nielen pre aplikáciu k-priemerov, ale vo všetkých prípadoch, kedy je potrebné predom určiť k počiatočných centier.

```
ptsymb = {'bs','r^','md','go','c^','yo','b.','r.','m.','g.','c.','y.
    ','bx','rx','mx','gx','cx','yx','b+','r+','m+','g+','c+','y+'};

k=21;
[rC]=inicializacia(r',k);

clf;
plot(r(:,1),r(:,2),'c.');
hold on;

plot(rC(:,1),rC(:,2),'ko', 'MarkerSize', 14, 'LineWidth',2);
plot(rC(:,1),rC(:,2),'kx', 'MarkerSize', 14, 'LineWidth',2);
grid on

for x=1:k
    s = sprintf('%f %f',rC(x,1), rC(x,2));
    disp(s);
end
disp(' ');
pause;
[rIDs, rC]=kmeans(r, k,'Distance','sqeuclidean', 'start', rC);
gplotmatrix(r, r, rIDs);

for x=1:k
    s = sprintf('%f %f',rC(x,1), rC(x,2));
    disp(s);
end
```

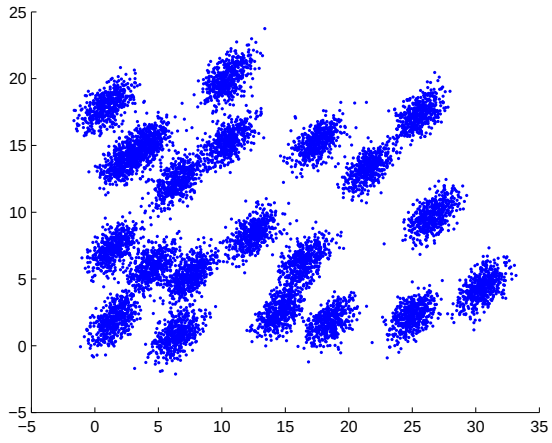
```
    disp(' ');
28  clf;
    for i = 1:k
30      clust = find(rIDs==i);
        plot(r(clust,1),r(clust,2),ptsymb{i});
32      hold on;
    end
34  plot(rC(:,1),rC(:,2),'ko', 'MarkerSize', 14,  'LineWidth',2);
    plot(rC(:,1),rC(:,2),'kx', 'MarkerSize', 14,  'LineWidth',2);
36
    grid on
```

Ukážka kódu 3.4: Algoritmus k-priemerov^{++}

Generovanie dátového súboru

Pre algoritmus k-priemerov^{++} sme vytvorili nový súbor umelo vytvorených dát. Každý objekt je popísaný dvomi kvantitatívnymi premennými a je rozšírením pôvodného súboru (Obrázok 3.1).

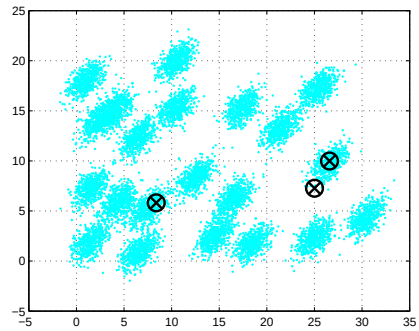
Výsledný dátový súbor pozostáva z 21 zhlukov. Každý zhluk je tvorený 500 objektmi a bol vytvorený pomocou skriptu 3.1. V našej implementácii je výsledná matica dát označovaná premennou r. Grafický výstup dátového súboru zobrazuje obrázok 3.16. Súbor tvorený dvojrozmernými dátami sme vybrali z dôvodu prehľadného grafického znázornenia.



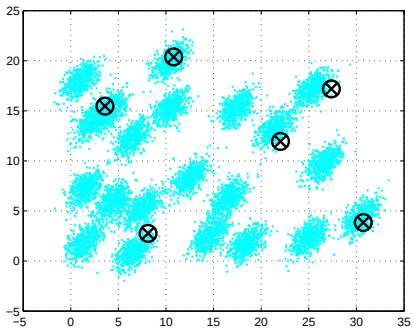
Obr. 3.16: Dátový súbor

Výsledné pozorovania

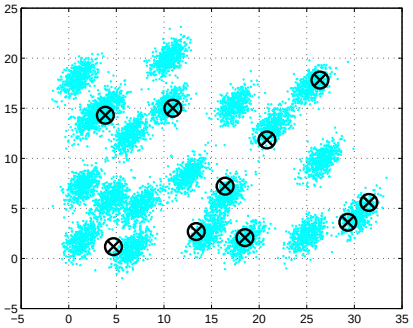
Na obrázku je zachytený postupný vývoj vytvárania množiny inicializačných centier. Vidíme, že inicializačný algoritmus rozmiestňuje centrá rovnomerne medzi všetky objekty. Výsledky sme porovnali s metódou k-priemerov, pričom bola použitá funkcia *replicates* a dospeli sme k tomu, že k-priemerov potrebuje vykonať vždy viac iterácií, aby bola účelová funkcia minimálna, pri metóde k-priemerov^{++} je počet iterácií v druhej fáze, kedy je aplikovaný klasický algoritmus k-priemerov, podstatne nižší. Môžeme povedať, že výsledky zhlukovania pomocou tohto algoritmu patria k najkvalitnejším vzhľadom k výslednej účelovej funkcii S, viď vzorec 2.1.



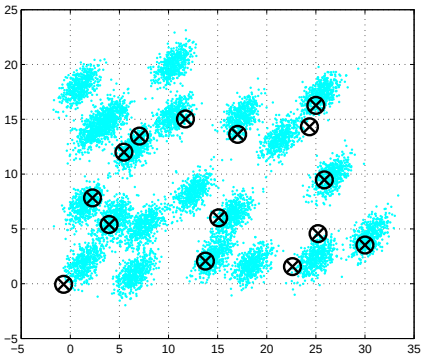
(a) pre 3 zhluky



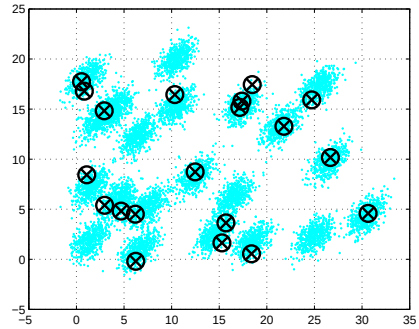
(b) pre 6 zhlukov



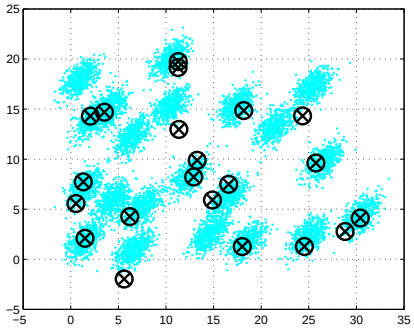
(c) pre 10 zhlukov



(d) pre 15 zhlukov



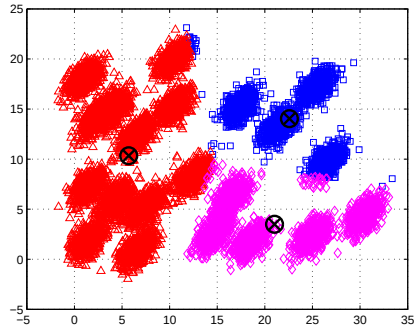
(e) pre 20 zhlukov



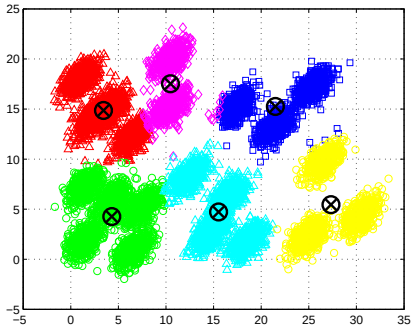
(f) pre 21 zhlukov

Obr. 3.17: Inicializačné rozmiestnenie centier pri metóde k-priemerov^{++}

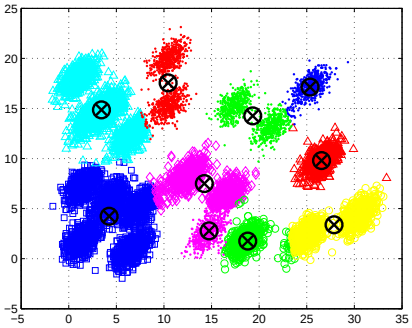
Takto vytvorené množiny inicializačných centier boli vstupom pre druhú fázu algoritmu k-priemerov^{++}. Na výsledných obrázkoch 3.18 dochádza k občasným posunom centier oproti počiatočnému rozmiestneniu, hlavne pri nižšom počte požadovaných počiatočných centrách. Posun nastáva z dôvodu, že inicializačné rozmiestenie centier vychádza vždy z náhodne voleného prvého centra.



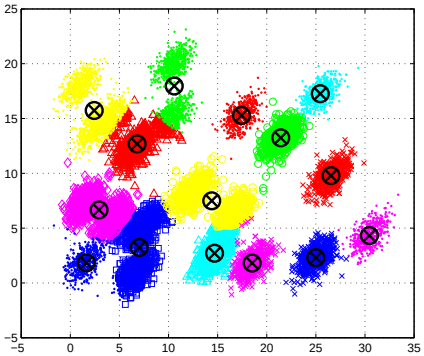
(a) pre 3 zhluky



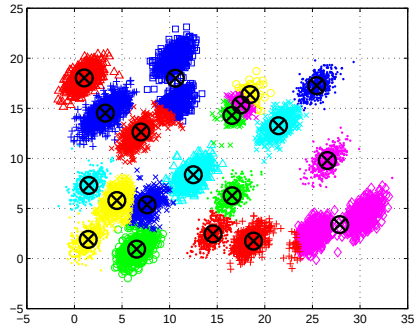
(b) pre 6 zhlukov



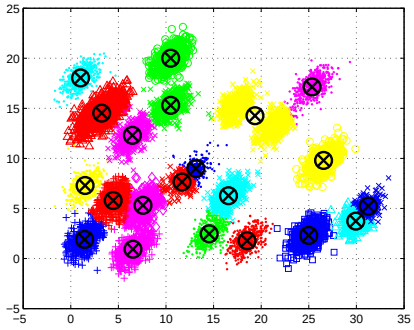
(c) pre 10 zhlukov



(d) pre 15 zhlukov



(e) pre 20 zhlukov



(f) pre 21 zhlukov

Obr. 3.18: Finálne rozmiestnenie centier pri metóde k-priemerov^{++}

3.2 Reálne dáta

3.2.1 Popis dátového súboru

Dátový súbor s názvom BANK pozostáva zo 600 objektov, ktoré sú charakterizované 11-timi atribútmi. Každý objekt predstavuje zákazníka banky. O zákazníkovi máme k dispozícii nasledujúce údaje:

vek (age) udáva vek daného zákazníka banky. Ide o kvantitatívnu premennú.

pohlavie (sex) udáva pohlavie daného zákazníka banky. Ide o kategoriálnu premennú, rozlišuje pohlavie MUŽ (MALE) alebo ŽENA (FEMALE).

región (region) udáva oblasť, v ktorej má daný zákazník trvalé bydlisko. Daný údaj rozlišuje 4 typy oblastí: CENTRUM MESTA (INNER-CITY), MESTO (TOWN), VIDIEK (RURAL), PREDMESTIE (SUBURBAN)

príjem (income) udáva objem príjmu v USD.

rodinný stav (married) vyjadruje, či je daný zákazník/-čka ženatý/vydatá.

deti (children) udáva počet detí. Ide o hodnoty z množiny $\{0, 1, 2, 3\}$.

auto (car) vyjadruje, či je daný zákazník vlastníkom auta alebo nie.

sporiaci účet (save account) vyjadruje, či daný zákazník vlastní tento typ účtu.

bežný účet (current account) vyjadruje, či daný zákazník vlastní tento typ účtu.

hypotéka (mortage) vyjadruje, či už banka poskytla tomuto zákazníkovi hypotekárny úver.

PEP (personal equity plan) ide o daňovo privilegovaný investičný akciový účet, ktorý slúži ako podpora vlastníctva akcií medzi širšou populáciou. Údaj vyjadruje, či daný zákazník vlastní tento typ účtu.

3.2.2 Predspracovanie dát

Daný dátový súbor bolo potrebné predspracovať, pretože obsahoval kategoriálne premenné. Úpravy bolo potrebné urobiť na všetkých atribútoch okrem veku, príjmu a počtu detí. Použili sme nasledujúce numerické označenie:

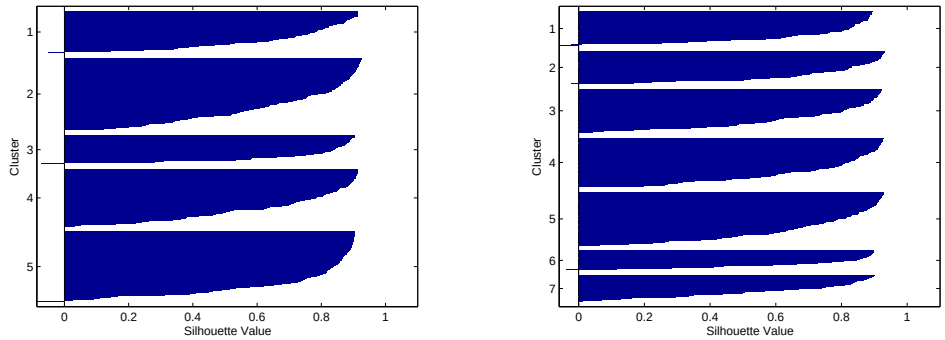
FEMALE=1, MALE=2
INNER-CITY=1, TOWN=2, RURAL=3, SUBURBAN=4
binárne hodnoty YES=1, NO=0

3.2.3 Priebeh aplikácie algoritmu

Na predpripravené dáta sme aplikovali algoritmus k-priemerov pre rôzne hodnoty k, ktoré sme postupne volili z množiny $\{2, ..., 9\}$. Pri každej aplikácii sme volili ako výstup obrysový graf, ktorý je pri viac ako trojrozmerných dátach jedinou možnou grafickou interpretáciou výsledných zhlukov. Zároveň nám poskytuje informáciu o kvalite zaradenia jednotlivých objektov.

Pre $k = 2, ..., 4$ obrysové grafy ukázali značné rozdiely vo vytvorených zhlukoch. Tie buď obsahovali záporné hodnoty mier príslušnosti vo veľkom množstve, čo naznačovalo zlé zaradenie objektov alebo obsahovali množstvo objektov s podpriemernou mierou príslušnosti, čo naznačovalo, že dané objekty môžu byť zaradené v inom zhluku, teda v tomto prípade bolo potrebné zvýšiť počet inicializačných centier.

Pre $k = 5$ (obrázok 3.19(a)) vidíme, že dané zhluky obsahujú nerovnomerné



(a) pre $k = 5$ (b) pre $k = 7$

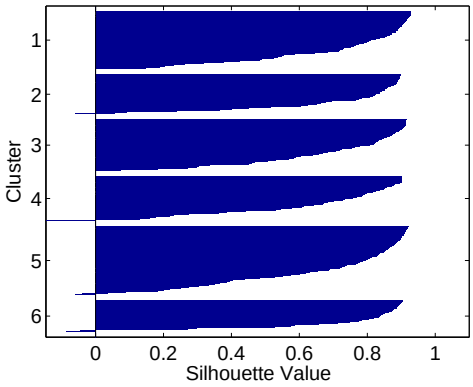
Obr. 3.19: Obrysové grafy

rozdelenie objektov, pričom zhluk 3 je príliš malý v porovnaní s ostatnými zhlukmi. Navyše zhluky 2 a 4 obsahujú veľké množstvo podpriemerných hodnôt mier príslušnosti, čo nás vedie k domneniu, že dané hodnoty môžu tvoriť ďalší zhluk.

Pre $k = 7$ (obrázok 3.19(b)) vidíme, že zhluky 6 a 7 sú v porovnaní s ostatnými zhlukmi príliš malé, pričom pri detailnejšej analýze daných objektov sme dospeli k tomu, že sú výsledkom rozdelenia jedného zhluku na dva menšie a veľmi podobné zhluky.

Predchádzajúce pozorovania ukázali, že optimálnou voľbou počtu inicializačných centier je $k = 6$. Oproti predchádzajúcim rozdeleniam nedostávame najvyššie priemerné hodnoty mier príslušnosti, avšak obrysový graf poukazuje na rovnomerné rozdelenie jednotlivých objektov do zhlukov. Zhluky neobsahujú veľa objektov s nízkou hodnotou miery príslušnosti, čo by naznačovalo, že zhluk je nutné rozdeliť na dva menšie. Na druhej strane neexistujú dva zhluky, ktoré by bolo nutné spojiť. Vhodnosť tohto výberu môžeme ukázať aj na príklade simulovaných dát. Ak porovnáme obrysové grafy 3.20 a 3.13, vidíme podobnosti v separácii zhlukov. Graf simulovaných dát predstavuje určité optimum a graf reálnych dát pre $k = 6$ predstavuje priblíženie sa tomuto optimu.

Záporné hodnoty sa vyskytujú aj v prípade rozdelenia do šiestich zhlukov. Keďže sa jedná o reálne dáta, musíme očakávať, že jednotlivé výsledné zhluky nebudú natoľko homogénne ako pri simulovaných dátach. Dáta z reálneho sveta



Obr. 3.20: Obrysový graf pre $k = 6$

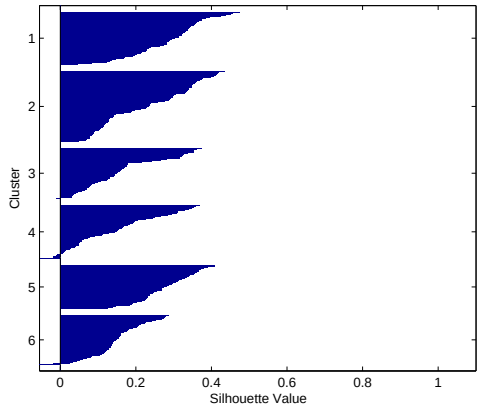
obsahujú často odľahlé hodnoty, ktoré sa prejavujú v deformácii jednotlivých obrysov a záporných hodnotách miery príslušnosti.

V prípade reálnych dát je v niektorých prípadoch nevyhnutnou súčasťou zhlukovania dát ich normalizácia. Ide buď o normalizáciu jednotlivých atribútov alebo normalizáciu celého dátového súboru. Aj v našom prípade sme vyskúšali normalizáciu najprv jednotlivých atribútov, kde sme skúšali normalizovať **vek**, **príjem** a **počet detí**, potom sme vyskúšali normalizáciu celého dátového súboru.

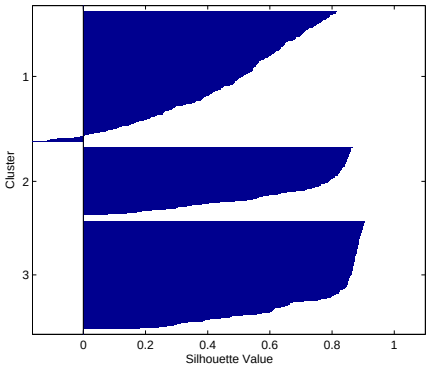
Experimenty ukázali, že normalizácia atribútov **vek** a **počet detí** s ponechaním ostatných hodnôt dát vôbec neovplyvňuje rozdelenie do zhlukov, ktoré sme prezentovali.

Normalizácia atribútu **príjem** dosiahla približne rovnaké rozdelenie ako pri normalizácii celého súboru. Tento fakt vyjadruje dôležitosť atribútu **príjem**, ktorý je hlavným atribútom ovplyvňujúcim rozdelenie do jednotlivých zhlukov.

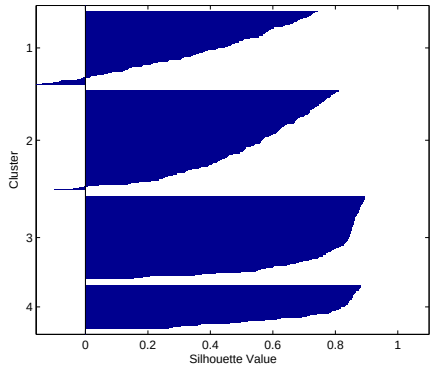
Normalizáciou celého dátového súboru sme dosiahli horšie rozdelenie v porovnaní s pôvodným dátovým súborom, ako vidíme na obrázku 3.21. Vidíme, že miery príslušnosti v jednotlivých zhlukoch sú značne rozdielne. Takisto, čo sa týka interpretácie vzniknutých zhlukov, vzájomná heterogenita zhlukov nebola výrazná. V konečnom dôsledku sme pre interpretáciu výsledkov zvolili rozdelenie do zhlukov pôvodného súboru.



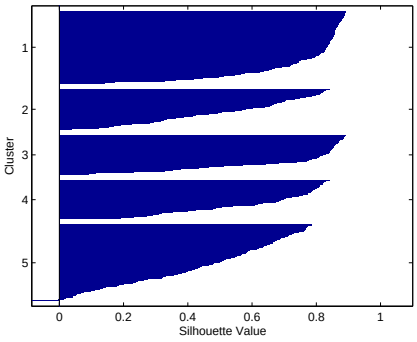
Obr. 3.21: Obrysový graf pre normalizovaný dátový súbor, $k = 6$



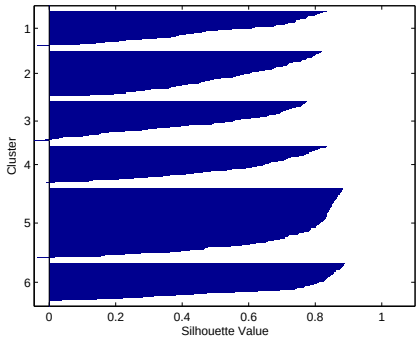
(a) pre 3 zhluky



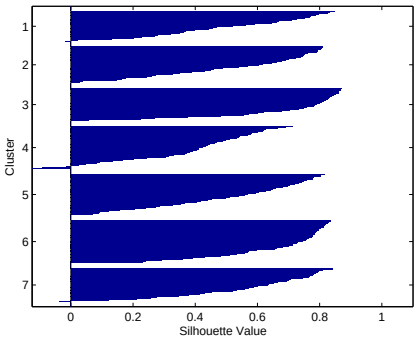
(b) pre 4 zhluky



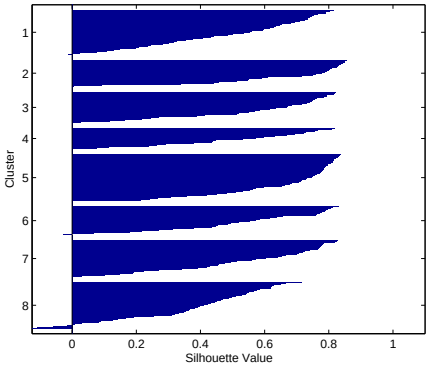
(c) pre 5 zhlukov



(d) pre 6 zhlukov



(e) pre 7 zhlukov



(f) pre 8 zhlukov

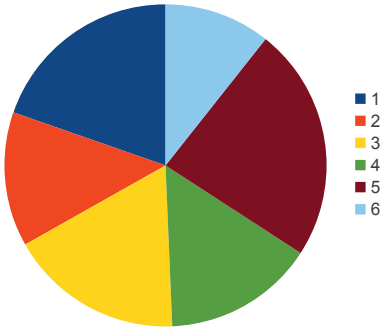
Obr. 3.22: Obrysové grafy pri normalizovaní atribútu **plat** vydelením hodnôt atribútu hodnotou 1000 (vyjadrenie platu v desiatkach tisícov)

Ako ďalší experiment sme skúsili normalizovať hodnoty atribútu **plat** vydelením hodnotou 1000, čím sme dostali vyjadrenie platov v desiatkach tisícov. Obrázok 3.22 zobrazuje obrysové grafy pre vyskúšané hodnoty k. Môžeme vidieť, podobne ako pri normalizácii celého dátoveho súboru, deformáciu jednotlivých obrysov. V porovnaní s pôvodnými dátami, aj v tomto prípade sa ukazuje byť optimálnou hodnotou $k = 6$. Dôvodom, prečo sme sa rozhodli tieto výsledky nepoužiť pri výslednej interpretácii je, že výsledné obrysy obsahujú často viac ako polovicu prvkov s podpriemernou mierou príslušnosti k zhluku, v ktorom sa nachádzajú. Tieto hodnoty priemerných mier príslušnosti sa pohybujú v okolí

hodnoty 0.6. To sa na jednotlivých grafoch (3.22) prejavuje ostro skosenými obrysmi, čo odporuje optimálnemu tvaru obrysov.

3.2.4 Interpretácia výsledkov

Rozdelenie do šiestich zhlukov prinieslo viacero zaujímavých pozorovaní. Ako bolo spomenuté, výrazným separačným atribútom bol **príjem**, ktorý je významným ukazovateľom v charakterizácii jednotlivých zhlukov. Výsledné rozdelenie zhlukov môžeme označiť za rozdelenie do platových skupín. Rozdelenie počtu jednotlivých objektov do zhlukov je navyše pomerne rovnomerné ako zobrazuje graf na obrázku 3.23.



Obr. 3.23: Graf počtu objektov priradených do zhlukov

Zhluk 1

Daný zhluk predstavuje skupinu 118 zákazníkov banky, ktorých príjem sa pohybuje v intervale $\langle 13950.4, 20114 \rangle$. Reprezentant tohto zhluku (centroid) má následujúce hodnoty:

age	sex	region	income	married	children
32.82	1.49	1.75	16826.54	0.73	0.32
car	save account	current account	mortage	pep	
0.48	0.59	0.75	0.36	0.41	

Vek osôb prislúchajúcich tomuto zhluku je rôznorodý, minimálny vek je 18 rokov, najstarším klientom je 58 ročný zákazník. V približne rovnakom pomere sú zastúpené obe pohlavia, žien je 60, mužov je 58.

Ako vidíme z hodnôt reprezentanta pri regióne, môžeme predpokladať, že zhluk pozostáva prevažne z obyvateľov miest a ich centier. Početnosti jednotlivých oblastí potvrdzujú naše pozorovanie, 64 klientov pochádza z centier miest, 30 klientov je všeobecne z miest, 14 klientov pochádza z vidieka a 10 klientov má trvalé bydlisko na predmestí.

86 klientov má založené rodiny a len 32 klientov má rodinný stav slobodný. Čo sa týka počtu detí, tam aj napriek vysokému počtu manželstiev je pomer bezdetných klientov a klientov s počtom detí 1 až 3 porovnateľný. 66 klientov má aspoň jedno dieťa, zvyšok je bezdetných. Táto informácia môže súvisieť práve s nízkym príjmom, kde vidíme, že priemerný príjem je okolo 16000, čo predstavuje

druhý najnižší príjem klientov v rámci 6-tich zhlukov.

Približne polovica, 61 klientov, vlastní auto, 57 klientov ho nevlastní. Sporiaci účet využíva len 48 klientov, ale bežný účet využíva 80 klientov. 75 klientov ešte nevyužilo hypotekárny úver, ktorý banka poskytuje a porovnateľné je to aj s pep účtom, ktorý využíva len 48 klientov.

Zhluk 2

Tento zhluk pozostáva z 81 klientov banky, ktorí predstavujú skupinu s najnižším príjmom. Jeho hodnoty sa pohybujú v intervale $\langle 5014.21, 13864.6\rangle$.

age	sex	region	income	married	children
24.90	1.53	1.88	11000.85	0.58	0.36
car	save account	current account	mortage	pep	
0.37	0.65	0.81	0.30	0.27	

Reprezentant zhluku zastupuje mladú generáciu klientov vo veku okolo 24 rokov, čo skutočne vystihuje túto skupinu. Na rozdiel od predošlého zhluku môžeme povedať, že tentokrát pozorujeme väčšiu homogenitu vekovej kategórie. Keby sme to interpretovali pomocou najnižšej a najvyššej hodnoty, nebolo by to správne, pretože zhluk zahŕňa aj klienta s vekom 50 rokov, čo je hodnota výrazne sa odlišujúca od ostatných. Miera príslušnosti tohoto klienta je však 0.8138. Teda opäť sa potvrdzuje, že atribút vek nehrá najdôležitejšiu rolu v rozdelení do zhlukov.

Menšie zastúpenie v tomto zhluku majú ženy, je ich 38. Mužov je 43. Oproti zhluku 1 priemerná hodnota vyjadrujúca zastúpenie oblastí trvalého bydliska je vyššia, o čom svedčia aj početnosti jednotlivých oblastí: len $\frac{1}{4}$ klientov býva buď na vidieku alebo v predmestských častiach, zatiaľ čo v prípade prvého zhluku to bola len $\frac{1}{5}$ klientov. Porovnateľný fakt je v rozdelení klientov v mestách, tak ako aj v prvom zhluku, $\frac{2}{3}$ klientov býva v centrách miest.

Viac ako polovica klientov je v manželskom zväzku, 47 klientov. Zaujímavé v tomto prípade sú údaje o počte detí, kde 33 klientov je bezdetných, pričom neplatí rovnosť slobodný = bezdetný, 20 klientov má 2 deti a 1 dieťa má len 18 klientov. Zákazníkov s 3 deťmi je iba 10.

Vzhľadom k nízkemu príjmu je očakávané, že väčšina klientov nevlastní automobil. Z 81 klientov vlastní auto len 30 klientov.

Približne $\frac{2}{3}$ klientov má zriadený sporiaci účet, čo predstavuje 53 klientov a len 15 z tejto skupiny klientov nevyužíva bežný účet v banke. Hypotekárne úvery využilo len 25 klientov, všetci však využívajú obidva typy predošlých účtov a väčšinou ide o klientov, ktorí majú aspoň jedno dieťa. Posledný typ účtu využívajú prevažne slobodní klienti, počet klientov je však veľmi nízky, len 22 z 81 využíva tento typ účtu.

Zhluk 3

Tretí zhluk je tvorený skupinou 105 ľudí, ktorých príjem sa pohybuje v intervale $\langle 27417.6, 35263.5\rangle$. V rámci rozdelenia do platových skupín, ide o skupinu s priemernou výškou príjmu.

age	sex	region	income	married	children
45.74	1.42	1.95	30858.03	0.63	0.37
car	save account	current account	mortage	pep	
0.58	0.68	0.75	0.39	0.51	

Ide o staršiu generáciu prevažne klientov okolo 40 až 50 rokov, i keď opäť tu nájdeme odľahlejšie hodnoty veku. Minimálny vek tejto skupiny je 30 rokov a maximálny až 67 rokov. Môžeme však potvrdiť, že centroid zhluku dostatočne vekovo vystihuje túto skupinu.

V tejto skupine klientov dominujú najmä ženy, v porovnaní s mužmi je ich počet 66. Rovnako ako v zhluku 2, $\frac{1}{4}$ klientov býva na vidieku a na predmestí, pomer klientov v mestečkách a v centrách miest je však v tomto prípade skoro porovnateľný. 42 klientov udáva INNER-CITY ako ich región a 36 klientov TOWN.

Rodinný stav ženatý/vydatá uvádzajú takmer $\frac{2}{3}$ klientov. V približne rovnakom pomere sa v tejto skupine nachádzajú klienti s aspoň jedným dieťaťom. 27 klientov má jedno dieťa a klientov s dvomi a tromi deťmi je rovnako 18. Teda skupina pozostáva najmä z rodín s viac ako 1 dieťaťom. Bezdetných klientov je 42.

Vidíme, že priemerná hodnota vyjadrujúca vlastníctvo automobilu je 0.5809. Predpokladáme, že tento údaj súvisí s vyšším počtom klientov s aspoň jedným dieťaťom. Takisto je tento údaj ovplyvnený vyšším príjmom v porovnaní s predošlými zhlukmi.

V tejto skupine klientov pozorujeme vyšší záujem o využívanie sporiacich účtov. 71 klientov využíva tento typ účtu a 79 klientov má zriadený bežný účet v banke. Takisto v porovnaní s predošlými zhlukmi sú využívané vo vyššej miere hypotekárne úvery. O posledný typ účtu má záujem viac ako polovica klientov.

Zhluk 4

Túto skupinu 91 klientov môžeme charakterizovať ako skupinu s nadpriemerným príjmom v rámci našich zhlukov. Príjem daných klientov sa pohybuje v intervale $\langle 35610.5, 46633 \rangle$.

age	sex	region	income	married	children
52.86	1.55	1.91	40308.17	0.66	0.34
car	save account	current account	mortage	pep	
0.53	0.82	0.79	0.32	0.46	

Veková kategória tohoto zhluku je generáciou prevažne päťdesiatnikov. Vekovo aj príjmovo by sme hierarchicky mohli tento zhluk zaradiť za skupinu klientov z tretieho zhluku. Ak si všimneme interval príjmu, skoro úplne nadväzuje na predošlý interval.

V tomto zhluku majú väčšie zastúpenie muži. Rozdiel však nie je veľký, 50 klientov sú muži a 41 klientov predstavujú ženy. Hodnoty oboch centroidov pri atribúte región sú porovnateľné, dokonca zastúpenie klientov na vidieku a v predmestských častiach je rovnaké. Táto skupina klientov preferuje viac centrá miest.

Aj v tomto zhluku je len $\frac{1}{3}$ slobodných klientov. Celkovo 43 klientov nemá dieťa a počet klientov s dvomi deťmi je 19, čo je viac ako klientov s jedným

dieťaťom, ich počet je 16 a viac ako klientov s tromi deťmi, ich počet je 13.

Automobil vlastní 48 klientov, pričom 43 klientov auto nemá. Pozorujeme však, že skoro vo všetkých prípadoch klienti vlastniaci automobil majú zriadený aj sporiaci účet v banke. V tomto prípade však vôbec túto hodnotu neovplyvňuje počet detí a nemôžeme tvrdiť, že každý klient s aspoň jedným dieťaťom auto vlastní.

Sporiaci účet je využívaný vo veľmi veľkej miere. Tento zhluk je druhým zhlukom spomedzi všetkých, ktorí najviac využíva tieto služby banky. Podobne je to aj s využitím bežného účtu, o čom svedčí aj hodnota atribútu centroidu 0.7912. Hodnota využitia hypotekárnych úverov je porovnateľná so zhlukom 2. Približne $\frac{2}{3}$ klientov nevyužilo tento produkt. Záujem o posledný typ účtu má menej ako polovica klientov tejto skupiny.

Zhluk 5

Tento zhluk tvorí najväčšiu skupinu klientov banky, ide o 141 klientov, ktorých príjem sa pohybuje v intervale $\langle 20236.2, 27056.5 \rangle$.

age	sex	region	income	married	children
42.40	1.53	1.97	23607.25	0.67	0.29
car	save account	current account	mortage	pep	
0.47	0.57	0.70	0.35	0.43	

Podľa veku nemôžeme presne charakterizovať túto skupinu, pretože klienti tohoto zhluku zastupujú všetky vekové kategórie. Minimálny vek klienta je 23 rokov, maximálny 67 rokov. Skôr by sme mohli povedať, že ide o pracujúcu vrstvu s mierne podpriemerným príjmom.

Obidve pohlavia majú skoro rovnaké zastúpenie, mužov je 74 a žien 67, čo nepredstavuje veľký rozdiel. Zaujímavým faktom však je, že počet skupín klientov s bydliskom v centrách miest (INNER-CITY) je 53, čo je rovnaká hodnota ako počet klientov s bydliskom v mestách (TOWN). Na vidieku žije 21 klientov a v predmestských častiach len 14.

Opäť okolo $\frac{2}{3}$ klientov je v manželskom zväzku. Zhluk sa vyznačuje najnižšou hodnotou atribútu **deti** centroidu. 70 klientov má viac ako jedno dieťa a až 71 je bezdetných.

Analogicky predpokladáme s výškou príjmu pokles počtu klientov, ktorí vlastnia automobil, čo je podobný ukazovateľ ako v prípade zhlukov 1 a 2.

Sporiaci účet využívajú $\frac{4}{7}$ klientov tejto skupiny. V rámci všetkých zhlukov môžeme charakterizovať túto skupinu ako klientov, ktorí majú najmenší záujem o bežné účty ponúkané bankou. Hypotekárne úvery využíva $\frac{1}{3}$ z nich. PEP účet využíva 61 klientov.

Zhluk 6

Tento zhluk tvorí 64 klientov banky a patrí medzi najmenší zhluk. Príjem klientov tejto skupiny je nadpriemerný a najvyšší. Hodnoty príjmov sa pohybujú v intervale $\langle 46870.4, 63130.1 \rangle$.

age	sex	region	income	married	children
61.80	1.48	2.13	53141.55	0.66	0.39
car	save account	current account	mortage	pep	
0.53	1	0.80	0.33	0.73	

Veková kategória tohoto zhluku je veľmi dobre popísateľná aj pomocou hraničných hodnôt. Minimálny vek klienta tejto skupiny dát je 52 rokov, maximálny 67. V tomto prípade dostávame presne definovateľnú vekovú skupinu, ktorú môžeme nazvať generáciou klientov pred dôchodkovým vekom alebo v dôchodku.

Pohlavie je opäť skoro rovnomerne zastúpené. Žien je 33 a mužov 31. Región sa tiež podstatne odlišuje od predošlých zhlukov, viac ako $\frac{1}{3}$ klientov dáva prednosť bývaniu na vidieku a v predmestských častiach.

Pomer rodinných stavov je rovnaký ako u predošlých zhlukov. Opäť $\frac{1}{3}$ je slobodných a $\frac{2}{3}$ majú založené rodiny.

V porovnaní s ostatnými zhlukmi, tento sa vyznačuje najvyšším pomerom klientov s aspoň jedným dieťaťom v porovnaní s bezdetnými klientmi. Prevažne tu patria klienti s dvomi deťmi.

Viac ako polovica vlastní automobil, čo je zaujímavé v porovnaní s interpretáciou počtu vlastníkov áut v predošlých zhlukoch. Príjem klientov je vysoký, ale zjavne si peniaze radšej nechávajú na sporiacom účte, pretože ako jediný zhluk má 100% klientov využívajúcich tento typ účtu.

Približne $\frac{1}{5}$ má zriadený bežný účet a $\frac{1}{3}$ využila hypotekárny úver. Výrazne vysoký záujem tejto skupiny klientov je o PEP účet, až 47 klientov.

Zhrnutie

Analýza BANK dátového súboru priniesla zaujímavé pozorovania. Jednoznačné rozdelenie môžeme sledovať v rozdelení príjmov. Jednotlivé intervaly charakterizujúce príjmy šiestich skupín na seba nadväzujú a na prvý pohľad je aj z výsledkov jasné, že atribút **príjem** zohral v zhlukovaní najpodstatnejšiu rolu. Podľa platových skupín by sme dostali následujúce usporiadanie zhlukov: 2,1,5,3,4,6.

Ďalším atribútom, podľa ktorého by sme mohli vytvoriť určitú hierarchiu týchto zhlukov je atribút **vek**. V tomto prípade už rozdelenie nie je až tak jednoznačné. Vidíme, že zhluky 3 a 5 zahŕňajú skoro podobné vekové skupiny, ak porovnáme hodnoty centroidov. V skutočnosti však zhluk 5, ako sme popisovali, zahŕňa všetky vekové kategórie a rozdiel medzi najmladším a najstarším klientom tohoto zhluku je 44 rokov. Ostatné vekové skupiny zahŕňajú približne klientov v rozmedzí 10 rokov. Usporiadanie podľa vekových skupín je totožné s predchádzajúcim, čo ukazuje, že vek zohral tiež určitú rolu v zhlukovaní.

Ostatné atribúty dotvárali konečné výsledky zhlukovania, ale neboli zanedbateľné. Pri interpretácii jednotlivých zhlukov sme našli niektoré súvislosti, ktoré sme vyššie popísali. Zaujímavým pozorovaním je, že sporiace účty využívajú najmä skupiny s nízkym príjmom alebo skupiny z najvyšším príjmom. V najväčšej miere však sporiace účty využíva skupina 24 ročných klientov, ktorá predstavuje študentov, prípadne začínajúce rodiny a je pochopiteľné, že takéto produkty preferujú.

Záver

Zhluková analýza, ako štatistická disciplína, zahŕňa široké spektrum metód a algoritmov používaných pri zhlukovaní dát. S postupom času vznikajú stále novšie metódy, ktoré buď určitým spôsobom nadväzujú na tradičné algoritmy, teda vznikajú rôzne modifikácie, ktoré sú efektívnejšie a rýchlejšie, alebo vznikajú celkom nové algoritmy.

V tejto bakalárskej práci som sa zamerala na tradičnú metódu k-priemerov, ktorá je jednou z najbežnejších a najznámejších metód zhlukovej analýzy. Mojím cieľom bolo uviesť čitateľa do problematiky zhlukovej analýzy, ktorá je však omnoho obšírnejšia, než zahŕňa táto práca. Úvodné kapitoly by mali predstaviť rôzne postupy zhlukovania, ktoré sa v praxi používajú a umožňujú na základe teoretického podkladu lepšie porozumieť praktické aplikácie algoritmov zhlukovej analýzy. Pre úplné pochopenie práce algoritmu som zvolila aplikáciu na simulovaných dátach, ktoré sú pre grafické znázornenie najvhodnejšie. Vhodným prínosom sú aj ukážky kódu implementované v prostredí matematického softvéru Matlab, ktoré dopĺňajú zrozumiteľnosť popisovaných výsledkov.

V reálnom svete môže zhluková analýza dát odhaliť zaujímave pozorovania a nečakané súvislosti, ktoré bežným skúmaním nie sú pozorovateľné. Na finančných dátach som demonštrovala postup zhlukovania, ktorého výsledkom bolo niekoľko zaujímavých skutočností, ktoré sú interpretované a diskutované v predchádzajúcej kapitole.

Téma bakalárskej práce bola zaujímavá, pretože teoretické znalosti dopĺňa praktickou aplikáciou metódy, ktorá celej práci dodáva zrozumiteľnosť a pomáha pochopiť danú problematiku. So spracovávanou problematikou som bola nakoniec veľmi spokojná a myslím si, že výsledná práca predstavuje ucelený pohľad na metódu „k-means“ a jej aplikáciu v praxi.

Zoznam použitej literatúry

[1] Wunsch, D. C., Xu, R. *Clustering*. John Wiley and Sons, 2009. ISBN 0-470-27680-0.

[2] Späth, H. *Fallstudien Cluster - Analyse*. 1. Aufl. München, Wien: Oldenbourg, 1977. ISBN 3-486-20771-7.

[3] Aldenderfer, M. S., Blashfield, R. K. *Cluster analysis*. SAGE, 1995. ISBN 0-803-92376-7.

[4] Arthur, D., Vassilvitskii, S. *k-means++: The Advantages of Careful Seeding*.
http://www.stanford.edu/ darthur/kMeansPlusPlus.pdf, stav z 1. 8. 2011

[5] Jiang, M. F., Tseng, S. S., Cu, C. M. *Two-phase clustering process for outliers detection*.
http://sci2s.ugr.es/docencia/doctoM6/TwoPhaseClusteringDetection-Outliers.pdf, stav z 25. 4. 2000

[6] Řezanková, H., Húsek, D., Snášel, V. *Shluková analýza dat*. 1.vyd. Praha: Profesional Publishing, 2007. ISBN 978-80-86946-26-9

[7] Bassi, I., De Poi, P. *Measuring multifunctional (agritouristic) characterization of the territory*.
http://infoagro.net/shared/docs/a5/19.pdf, stav z 8. 11. 2010

[8] Laurent, S. *k-means++*.
http://www.mathworks.com/matlabcentral/fileexchange/28804-k-means++, stav z 24. 7. 2011

A. Teória grafov - základné pojmy

Graf ako matematickú štruktúru využíva v zhlukovej analýze množstvo aplikácií. V tejto kapitole zadefinujeme najdôležitejšie pojmy z teórie grafov, s ktorými sme sa stretli v predchádzajúcich kapitolách.

Definícia. *Grafom $G = (V, E)$ nazývame usporiadanú dvojicu množín $V = \{v_i\}$ a $E = \{e_i\}$, kde V je neprázdna konečná množina vrcholov v_i grafu a E je množina dvojprvkových podmnožín (v_i, v_j) množiny V, kde $i, j = 1, ..., n$ a $i \neq j$. Prvky množiny V nazývame* vrcholy grafu *a prvky množiny E nazývame* hrany *grafu.*

V zhlukovej analýze je množina $V = \{v_i\}$ najčastejšie reprezentovaná objektami zhlukovania. V našom prípade to budú centrá zhlukov.

Definícia. *Graf $G = (V, E)$ nazveme* orientovaný, *ak E je množina usporiadaných dvojíc typu (v_i, v_j), kde $v_i \neq v_j$, nazývaných* orientované hrany *grafu.*

Definícia. *Graf $G = (V, E)$ nazveme* neorientovaný, *ak E je množina neusporiadaných dvojíc typu $\{v_i, v_j\}$, kde $v_i \neq v_j$, nazývaných* hrany *grafu.*

Dôležitým druhom grafu je *strom*. Túto štruktúru využívajú mnohé zhlukovacie metódy. Stromy sa využívajú na reprezentáciu hierarchických databáz a v mnohých komunikačných, či distribučných sieťach za účelom zobrazenia stromovej štruktúry.

Definícia. *Strom $T = (V, E)$ je neprázdny súvislý graf bez kružníc.*

Definícia. *Les je graf, ktorého komponentmi sú stromy.*

Definícia. *Súvislým grafom nazývame (neorientovaný) graf, v ktorom platí, že pre každé dva vrcholy v_i, v_j existuje aspoň jedna cesta z v_i do v_j.*

Definícia. *Kružnica je graf $C_n = (V, E)$, kde $V = \{v_1, ..., v_n\}$ a $E = \{e_1, ..., e_n\}$ a platí:*

- *orientovaný graf*
  $e_i = (v_i, v_{i+1}), i = 1, ..., n-1$ *a* $e_n = (v_n, v_1)$
  každý vrchol orientovanej kružnice má vstupný i výstupný stupeň rovný 1
- *neorientovaný graf*
  $e_i = \{v_i, v_{i+1}\}, i = 1, ..., n-1$ *a* $e_n = \{v_n, v_1\}$
  každý vrchol neorientovanej kružnice má stupeň 2

Definícia. *Graf neobsahujúci žiadnu kružnicu nazývame* acyklickým *grafom.*

Definícia. *Graf G sa nazýva* ohodnotený, *ak každej hrane e_i je priradené nejaké číslo $w_i \in \mathbb{R}$*

Definícia. *Úplný graf označuje taký neorientovaný graf, v ktorom sú každé dva vrcholy spojené hranou.*

Definícia. *Kostra grafu $G = (V, E)$ je taký jeho podgraf H, ktorý je stromom a zároveň množina vrcholov grafu H je totožná s množinou vrcholov grafu G a $V(H) = V(G)$. V grafe G = (V,E) existuje kostra práve vtedy, ak G je súvislý.*

Definícia. *Minimálna kostra grafu je kostra ohodnoteného grafu, ktorá má najmenšie ohodnotenie hrán spomedzi všetkých kostier grafu.*