UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA MATEMATICKÉ ANALÝZY A APLIKACÍ MATEMATIKY

# DIPLOMOVÁ PRÁCE

## Shluková analýza

Vedoucí diplomové práce:
**Mgr. Jaroslav Marek Ph.D.**
Rok odevzdání: 2010

Vypracovala:
**Bc. Kateřina Spohnerová**
AME, II. ročník

**Prohlášení**

Prohlašuji, že jsem diplomovou práci zpracovala samostatně pod vedením pana Mgr. Jaroslava Marka Ph.D. s použitím uvedené literatury a softwaru.

V Olomouci dne 8.4.2010

**Poděkování**

Na tomto místě bych chtěla poděkovat především svému vedoucímu diplomové práce panu Mgr. Jaroslavu Markovi Ph.D., za jeho trpělivost a čas, který mi věnoval při konzultacích, a také za cenné rady při práci s programem MATLAB. Poděkování si zaslouží i moje rodina a přátelé za podporu, kterou mi vyjadřovali po celou dobu studia.

# Obsah

**Abstrakt:**

Shluková analýza patří mezi velmi atraktivní vícerozměrné statistické metody. Hlavním cílem shlukové analýzy je jednak seskupit jedince do skupin na základě jejich podobností a rozdílností a dále také zhustit informace o jedincích redukcí individuálních popisů objektů do relativně menšího počtu shluků. V práci jsme se zaměřili na studium klasický metody hierarchické a nehierarchické analýzy. Dalším naším cílem je porovnat tyto metody s alternativními přístupy, např. neuronové sítě, genetické algoritmy, atd.

**Klíčová slova:** shluková analýza, hierarchické metody, nehierarchické metody, neuronové sítě, genetické algoritmy.

**Abstract:**

Cluster analysis is ranked among very attractive multivariate methods. The main goal of cluster analysis is the assignment of a set of observations into clusters so that observations in the same cluster are similar in some sense and the reduce information about observations into the smaller number of clusters. In the present paper we focused on the study of classic methods of hierarchical and non-hierarchical clustering. The next our goal is the compare this methods with alternative methods, e.g. neural networks, genetic algorithms, etc.

**Keywords:** cluster analysis, hierarchical methods, non-hierarchical methods, neural networks, genetic algorithm.

# Cíle práce

Cílem práce je nastudovat různé metody shlukové analýzy a jejich užití demonstrovat na příkladech. V práci jsem se nejdříve zaměřila na využití programu MATLAB a SAS při shlukování pomocí metod implementovaných v jejich knihovnách. Pomocí těchto programů jsem se věnovala aplikacím některých metod shlukové analýzy, metodě nejvzdálenějšího souseda, metodě nejbližšího souseda a centroidní metodě, které patří mezi hierarchické metody, dále metodě $k$ průměrů patřící mezi metody rozkladu. Užívání již naprogramovaných programových balíků s sebou nese i některé nevýhody. Např. MATLAB ani SAS nám neposkytují podrobný popis postupu při shlukování. Proto jsem si pomocí MATLABu vytvořila vlastní algoritmus hierarchického shlukování, který nám poskytuje podrobné informace o průběhu shlukování.

Jednotlivé metody jsem aplikovala na dvou příkladech. První příklad je věnován 11 hlasování 200 poslanců, která jsem získala z internetových stránek Poslanecké sněmovny. Druhý příklad využívá data z entomologického výzkumu věnovaného výskytu 123 druhů motýlů ve 113 lokalitách Bílých Karpat.

Dalším mým cílem práce je ukázat shlukování pomocí alternativních přístupů, které jsou vytvořeny na základě inspirace přírodou. Zaměřila jsem se na genetické algoritmy, jež jsou odvozeny z biologické genetiky a teorie evoluce. Konkrétně jsem zpracovala GA $k$ průměrů, který využívá operátou selekce, mutace a křížení. Vytvořený algoritmus jsem aplikovala na uvedeném příkladu výskytu motýlů.

Cílem práce je pomocí vytvořených algoritmů klasifikovat mnohorozměrná data na základě jejich podobnosti či nepodobnosti. Např. v první úloze z Poslanecké sněmovny lze určit, kteří poslanci hlasují shodně či rozdílně. V příkladu výskytu motýlů lze rozhodnout, kteří motýli vyhledávají stejné rostliny a obdobné lokality.

Doufám, že svou prací čtenáře pro shlukovou analýzu nadchnu a inspiruji k dalšímu studiu.

# 1. Úvod

V dnešní době se klade velký důraz na zobrazení mnohorozměrných datových souborů a jejich podrobnějšímu popisu. Rozsáhlé dimenze je v podstatě nemožné zobrazit a navíc některé z konvenčních metod a technik nefungují dobře pro velké množství pozorování, z tohoto důvodu je těžké interaktivně zobrazit rozsáhlé a komplexní datové soubory. Proto je dnes velké množství pozornosti věnováno objevování nových technik a softwarů v dané problematice.

Kromě metod shlukové analýzy lze k redukci dimenze a případně i k vizualizaci mnohorozměrných datových souborů použít i jiné metody: multidimension scalling (MDS), Metoda hlavních komponent (PCA), Fisher discriminant analysis (FDA), mv-plot(algoritmus), self-organized map (SOM).

V této práci se budeme věnovat metodám shlukové analýzy a to klasickým metodám rozkladu, metodám hierarchickým, dále se podíváme na alternativní metody z oblasti neuronových sítí a genetických algoritmů.

Smysl shlukování použijeme na *mv* algoritmu, který budeme v následující sekci aplikovat na příkladu Fisherova měření irisů.

Ve druhé kapitole se seznámíme se základními pojmy potřebné pro shlukovou analýzu. V současnosti se při shlukování využívá dvou základních odvětví metod. Mezi ně patří hierarchické metody, které jsou uvedeny ve třetí kapitole, a metody rozkladu (nehierarchické metody), kterým je věnována čtvrtá kapitola. Při shlukování se využívají i alternativní přístupy inspirované přírodou, se kterými čtenáře seznámíme v páté kapitole.

V současnosti je řešení shlukové analýzy poskytováno různými programy. V šesté kapitole se zaměříme na aplikaci základních shlukovacích metod programem MATLAB na příkladu hlasování poslanců. Dále si ukážeme řešení metod shlukování pomocí programu SAS na příkladu výskytu motýlů. Na závěr této kapitoly si ukážeme vlastní algoritmus metody nejbližšího souseda na příkladu z Poslanecké sněmovny a shlukování motýlů pomocí genetického algoritmu $k$ průměrů. K jejich vytvoření jsme použili program MATLAB.

## 1.1. Motivace

Základním cílem shlukové analýzy je redukce dat, když údaje o $n$ objektech z prostoru $R^m$ transformujeme do prostoru nižší dimenze $R^k$. Tuto transformaci do roviny ($k = 2$) budeme v této části demonstrovat pomocí $mv$-algoritmu ([11]) na příkladu měření irisů.

Metoda je navržena na zkoumání skrytých struktur z daného datového souboru. Je to způsob pro vizualizaci shlukování a klasifikace. $Mv$ algoritmus zakresluje body z $R^d$ do bodů v $R^2$ a je sestaven ze dvou souvislých odhadů $m$ a $v$. Máme pozorování $x \in R^d$ jako $\mathbf{x} = (x_1, \ldots, x_d)$, pak

$$m = \frac{1}{d} \sum_{j=1}^{d} |x_j| \,, \tag{1}$$

$$v = \sqrt{g(x, f(x))} = \left( \frac{1}{d} \sum_{j=1}^{d} |x_j - f(x)|^2 \right)^{\frac{1}{2}} . \tag{2}$$

Obecně různorodé datové soubory o $n$-pozorováních v $d$-dimenzionálním souboru označujeme $\mathbf{X} = \{x_{ij}, i = 1, \ldots, n; j = 1, \ldots, d\}$. V tomto procesu budou vytvořeny vektory $m$ a $v$, oba délky $n$, určené těmito vztahy:

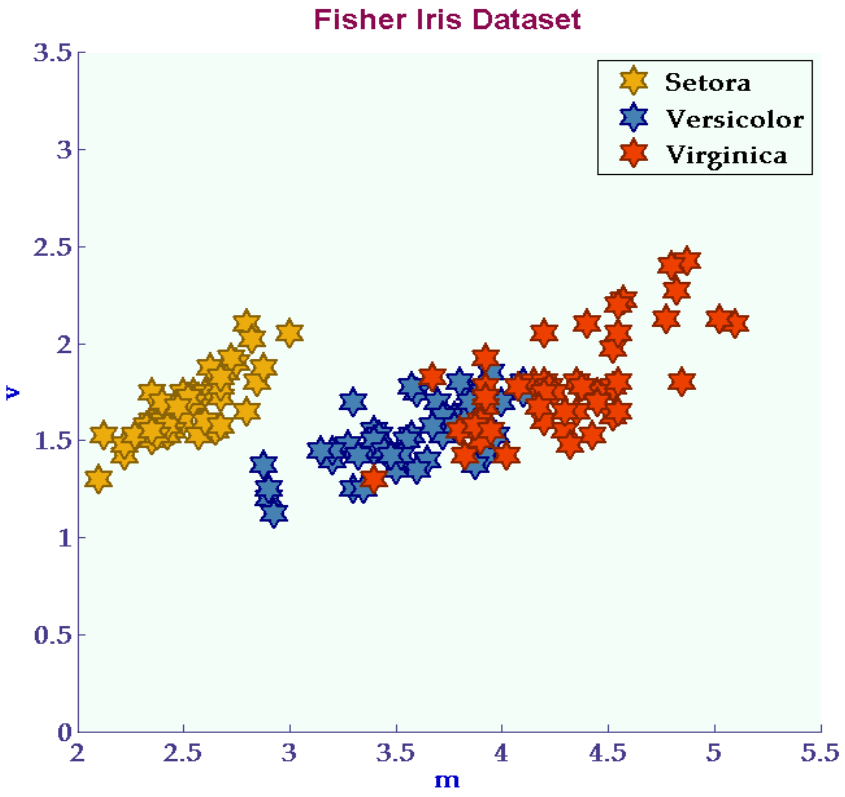
$$m_i = \frac{1}{d} \sum_{j=1}^{d} |x_{ij}| \,, \tag{3}$$

$$v_i = \left( \frac{1}{d} \sum_{j=1}^{d} (x_{ij} - m_i)^2 \right)^{\frac{1}{2}} . \tag{4}$$

Nyní budeme demonstrovat $mv$-algoritmu na příkladu měření irisů. Máme k dispozici data z měření délky a šířky kališního lístku a okvětního lístku irisů. Irisy patří do třídy Setora, Virginaca a Versicolor.

Po dosazení do vztahů (3) a (4) dostáváme dva vektory $m$ a $v$. V rovině vykreslíme body o souřadnicích $[m, v]$. Na základě daných měření se nám jednotlivé

irisy rozdělily do tří shluků podle třídy irisů. Irisy Setora tvoří samostatný shluk, kdežto Virginica a Versicolor tvoří dva překrývající shluky. Po podrobnějším prozkoumání by se mohlo zdát, že zde jsou vytvořeny čtyři samostatné shluky a to výše zmíněný shluk irisů Setora, dále dva malé shluky, které tvoří několik málo jedinců irisů Versicolor a Virginica. Zbývající shluk je tvořen irisy jak Virginica, tak Versicolor. Je to dáno velkou podobností těchto dvou druhů irisů.




Obrázek 1: Irisy a mv - algoritmus

# 2. Základní principy shlukové analýzy

Základním cílem shlukové analýzy je zařadit objekty do skupin (shluků) a to především tak, aby dva objekty stejného shluku si byly více podobné, než dva objekty z různých shluků.

Můžeme rozlišovat dva typy slukování a to *konvenční* a *konceptuální*. Konvenční je založeno na měření podobnosti, kde podobnost dvou znaků je funkcí jejich vlastností. U konceptuálního shlukování jsou shluky založeny na konceptuální soudržnosti, která je funkcí jednak vlastností objektů, tak popisného jazyka a okolí. Popisný jazyk je způsob, jakým jsou popsány třídy objektů, a okolí je množina sousedících vzorů. Konceptuální shlukování předpokládá, že známe charakteristiku shluků, do kterých mohou být objekty zařazeny.

Základní princip shlukové analýzy je, že každý objekt je jednoznačně zařazen do jednoho shluku. Takové shlukování nazýváme *pevné (disjunktní)*. Objekt však může být zařazen do více shluků, tehdy mluvíme o *překrývajícím se* shlukování.

Význam shlukové analýzy spočívá v usnadnění vyhledávání informací např. ve vědeckém výzkumu, ve výuce, ale i v běžném životě jako cestování, nakupování atd.

Dále se budeme věnovat základním předpokladům pro shlukovou analýzu, které jsou také popsány v literatuře [3], [4], [5], [8], [15] a [16].

## 2.1. Prvky shlukování

Z hlediska analýzy dat je vstupem pro shlukování datová matice, výstupem je identifikace shluků. Při shlukové analýze se zkoumá podobnost objektů, k čemuž slouží míry podobnosti. K dalším úkolům shlukování patří stanovení počtu shluků a zejména interpretace výsledků.

### 2.1.1. Vstupní datová matice

Základem vícerozměrné statistické analýzy jsou $m$-rozměrná pozorování objektů, počet objektů je $n$. Při odvození některých statistických metod se vychází

ze vstupní matice, v níž řádky představují vektory údajů o jednotlivých objektech a sloupce odpovídají jednotlivým proměnným ([8], [15], [16]). Vstupní matice je rozměru $n \times m$. Dále ji budeme označovat $\mathbf{X}$ a její prvky $x_{il}$, kde $i = 1, 2, \ldots, n$ a $l = 1, 2, \ldots, m$.

Jiným typem vstupní datové matice pro dvě kategoriální proměnné je dvourozměrná tabulka sdružených četností (kontingenční tabulka) . V tomto případě bude rozměr vstupní matice $K_k \times K_l$, kde $K_k$ je počet kategorií $k$-té (řádkové) a $K_l$ počet kategorií $l$-té (sloupcové) proměnné. Prvky této tabulky budeme označovat $n_{rs}$, kde $r = 1, 2, \ldots, K_k$ a $s = 1, 2, \ldots, K_l$.

Některé statistické metody vycházejí z matice vzdáleností, jejímiž prvky jsou hodnoty charakterizující vztahy mezi všemi dvojicemi objektů. Takováto matice může být přímo k dispozici nebo jsou vzdálenosti vypočítány na základě původní vstupní matice. Předmětem shlukování nemusí být jen objekty, ale také kategorie.

### 2.1.2. Shluky

Obvykle se pojem shlukování spojuje se shlukováním objektů na základě podobnost vektorů. Při analýze dat mohou být hledány také shluky proměnných nebo současné shlukování objektů i proměnných.

Jiným hlediskem pro klasifikaci shluků může být rozdělení:

- *disjunktní* (=pevné, jednoznačné přiřazení objektu do skupin),
- *překrývající se* (někdy označováno jako chumáčování).

Většina postupů ve shlukové analýze se zaměřuje na vytváření disjunktních shluků.

Počet shluků budeme dále označovat $k$ a jednotlivé shluky jako $C_g$ pro $g = 1, 2, \ldots, k$.

Pro stanovení cíle shlukování, tj. jakých typů shluků chceme dosáhnout, je výběr shlukovacího algoritmu. V tomto smyslu rozlišujeme shlukování:

- *nehierarchické*,
- *hierarchické*.

V případě, že je každému objektu přiřazena míra příslušnosti, pak mluvíme o fuzzy shlukování.

Z hlediska postupu rozlišujeme metody, které vycházejí z matice vzdáleností, a metody, které je nevyužívají a vycházejí přímo ze vstupní datové matice ([8], [15]). Tyto metody nazýváme:

- *metody založené na vzdálenostech* (hierarchické metody),
- *metody vektorového prostoru* (nehierarchické metody = metody rozkladu).

Předpokladem metod nehierarchického shlukování je stanovení počtu shluků. Možností je provádět shlukování pro různé počty shluků a ze získaných výsledků zjistit optimální počet.

## 2.2. Předzpracování datového souboru

Předzpracování datového souboru spočívá zejména v proměnných, posouzení zda mají být zahrnuty všechny nebo jen některé. Výběrem proměnných se budeme zabývat v následující podkapilole, a to jednotlivým typům, jejich významu pro analýzu. Předmětem studia budou také vhodné transformace dat. Vycházíme z literatury [3], [5], [8] a [15].

### 2.2.1. Výběr proměnných

Vhodnost proměnných pro analýzu může být hodnocena jednak z věcného, tak statistického hlediska. Je třeba stanovit, které proměnné jsou významné z hlediska posouzení podobnosti objektů (objektivní, subjektivní).

Ze statistického hlediska je vhodné, aby v souboru zůstaly pouze proměnné statisticky nezávislé. Někdy však datové soubory mohlou být malého počtu, proto řešením je ponechání takových proměnných, mezi nimiž není silná závislost. Dále se budeme zabývat testováním závislosti a míry intenzity závislosti pro dvojice proměnných v závislosti na jejich typech. Míry intenzity závislosti se používají jako míry podobnosti proměnných.

Podle typu škály měření rozlišujeme proměnné:

- *nominální* - hodnoty jsou různé, nemůže stanovi jejich pořadí,
- *ordinální* - můžeme stanovit pořadí, ale nemůžeme urči, o kolik je jedna hodnota větší než druhá
- *kvantitativní*:
  - *intervalové* - můžeme určit o kolik je daná hodnota větší,
  - *poměrové* - můžeme určit o kolik i kolikrát je daná hodnota větší,
- *dichotomická* (alternativní - nabývá pouze dvou hodnot):
  - *symetrické* - obě kategorie stejné důležitosti,
  - *asymetrické* - jedna kategorie důležitější.

Kvantitativní proměnné můžeme dělit na :

- *diskrétní* - nabývají celočíselných hodnot,
- *spojité* - mohou nabývat libovolných hodnot z určitého intervalu.

Nominální, ordinální a kvantitativní diskrétní proměnné s malým počtem variant hodnot můžeme souhrně označit jako *kategoriální*. Základem posouzení závislosti kategoriálních proměnných je *chí-kvadrát test o nezávislosti*, založený na četnostech v kontingenční tabulce. Používá se přitom *chí-kvadrát statistika*. Koeficienty závislosti nabývají hodnot z intervalu $\langle 0; 1\rangle$ a $\langle -1; 1\rangle$, přičemž hodnota 0 znamená nezávislost.

### 2.2.2. Transformace dat

V případě nominálních proměnných se bere v úvahu, zda jsou všechny nominální nebo jen některé. Ty pak převádíme na skupinu pomocných proměnných (binárních), pro které lze použít stejné míry vzdáleností jako pro kvantitativní data. U binární proměnné je aritmetický průměr vypočítaný z daných hodnot shodný s relativní četností jedniček a výběrový korelační koeficient je shodný s koeficientem asociace, používaný jako výběrová míra zavislosti dvou dichotomických znaků. I ordinálních proměných můžeme převést na proměnné binární.

Pokud hodnoty určité proměnné vyjadřují pořadí a kódování začíná hodnotou jedna, pak je doporučována transformace do intervalu $\langle 0; 1\rangle$. Je-li kódování jiné, je třeba každou hodnotu $x_{il}$ nahradit jejím pořadím $o_{il}$ a teprve pak může být pořadí transformováno do intervalu $\langle 0; 1\rangle$. Poté můžeme použít míry pro kvantitativní data.

I pro kvantitativní data existují postupy pro jejich transformaci. Pro data měřená v poměrové škále máme dva přístupy, a to zlogaritmování hodnot a postup pro ordinální data. Pro proměnné měřené na intervalové škále lze použít standardizaci.

Pří výpočtu normovaných proměnných je od každé $i$-té hodnoty $l$-té proměnné odečten aritmetický průměr hodnot této proměnné a výsledek je dělen výběrovou směrodatnou odchylkou, tedy

$$z_{il} = \frac{x_{il} - \bar{x}_l}{s_l}\,, \tag{5}$$

$\bar{x}_l$ je aritmetický průměr, $l = 1, \ldots, m$, $i = 1, \ldots, n$,

$$\bar{x}_l = \frac{\sum x_{il}}{n}$$

a $s_l$ je výběrová směrodatná odchylka, tj.

$$s_l = \sqrt{\frac{\sum_{i=1}^{n}(x_{il} - \bar{x}_l)^2}{n-1}}\,. \tag{6}$$

Nová proměnná $z_{il}$ má střední hodnotu 0 a směrodatnou odchylku 1.

Některé metody jsou založeny na předpokladu, že proměnné mají nulové střední hodnoty. Toho docílíme transformací, kde od každé $i$-té hodnoty $l$-té proměnné odečteme aritmetický průměr hodnot této proměnné, tj.

$$z_{il} = x_{il} - \bar{x}_l\,, \tag{7}$$

pro $i = 1, \ldots, n$, $l = 1, \ldots, m$.

Další možnosti pro transformaci dat jsou uvedeny v literatuře [8], [15].

### 2.2.3. Chybějící údaje a odlehlé objekty

V reálných datových souborech mohou některé údaje chybět z různých příčin, např. nebylo možné je změřit, získané hodnoty byly nesmyslné, případně chybou při vstupu dat.

Ve shlukové analýze existují tři základní přístupy při chybějících údajích:

1. nahrazení chybějící hodnoty
2. vynechání objektu
3. použití speciální míry.

V případě nahrazení chybějící hodnoty můžeme např.:

a) *nahradit výběrovou mírou polohy* příslušné proměnné, např. aritmetický průměr

b) *nahradit podmíněnou (skupinovou) mírou polohy*, kdy jsou objekty zařazeny do skupin vytvořených podle hodnot vybrané proměnné

c) *nahradit hodnotou vyskytující se u jiného objektu.*

Odlehlé objekty tvoří při použití základních algoritmů samostatné shluky. Jsou-li odlehlé objekty identifikovány, pak by měly být ve vstupní matici vynechány, což je shodné s jedním z přístupů u chybějících údajů.

## 2.3. Velké datové soubory

Základním problémem velkých souborů dat je, že analýza vycházející z matice vzdáleností vypočtených pro všechny dvojice objektů je velmi náročná. Dalším problémem může být velký počet proměnných, neboť shlukovací algoritmy porovnávající objekty na základě měr podobnosti fungují efektivně do 16 proměnných. Snížení rozměru úlohy se provádí buď na základě analýzy hlavních komponent nebo zjistíme skupiny podobných proměnných a z každé skupiny ponecháme pouze jednu, která tuto skupinu reprezentuje. Skupiny podobných proměnných se stanoví např. pomocí shlukové analýzy nebo vícerozměrného škálování.

## 2.4. Snižování rozměru úlohy

Základním principem redukce dimenze je zmenšení počtu původních proměnných $m$. Je možné vybrat z původních proměnných jen některé nebo z původních vytvořit nové. Hlavním úkolem při analýze rozsáhlých souborů dat je nalezení lepší reprezentace dat pomocí vhodné transformace. Základním přístupem, jak snížit dimenzi dat, je metoda hlavních komponent, viz [5], [8], [15], [16].

### 2.4.1. Analýza hlavních komponent

Cílem analýzy hlaních komponent (PCA - Principal Component Analysis) je nalézt na základě rozsáhlého počtu proměnných menší množinu nových proměnných, která by poskytovala nejlepší možnou reprezentaci dat.

Za hlavní komponenty považujeme množinu proměnných $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_L$, z nichž každá je zapsána jako vhodná lineární kombinace výchozích proměnných $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$, přičemž $L < m$ ([8], [15]).

Předpokládejme, že pozorování $\mathbf{x}_t$, $t = 1, 2, \ldots, n$ ,pocházejí z $m$-rozměrného základního souboru s nulovými středními hodnotami a kovarianční maticí $\mathbf{\Sigma}_X$. První hlavní komponentou základního souboru nazveme normovanou lineární kombinaci $m$ náhodných veličin $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$, tj.

$$\mathbf{Y}_1 = \sum_{i=1}^{m} l_{1i}\mathbf{X}_i = \mathbf{X}l_1 \,,$$

která má mezi všemi ostatními normovanými lineárními kombinacemi výchozích proměnných největší rozptyl a pro kterou platí

$$\sum_{i=1}^{m} l_{1i}^2 = 1 \,.$$

obecně za $j$-tou hlavní komponentu základního souboru budeme považovat takovou normovanou lineární kombinaci

$$\mathbf{Y}_j = \sum_{i=1}^{m} l_{ji}\mathbf{X}_i = \mathbf{X}l_j \,,$$

která má mezi ostaními kombinacemi, nekorelovanými mezi předcházejícími hlavními komponentami $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_{j-1}$, největší rozptyl.

Hlavní komponenty $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_m$ jsou očíslovány podle zmenšujícího se rozptylu, tj. $\sigma^2_{\mathbf{Y}_1} \geq \sigma^2_{\mathbf{Y}_2} \geq \ldots \sigma^2_{\mathbf{Y}_m}$. Vektor $l_j$ je tzv. vlastní vektor kovarianční matice, tj. jeho složky $l_{j1}, l_{j2}, \ldots, l_{jm}$ se určují jako normované řešení soustavy rovnic

$$(\mathbf{\Sigma_Y} - \lambda_j \mathbf{I}) l_j = 0\,, \tag{8}$$

kde $\mathbf{\Sigma_Y}$ je kovarianční matice hlavních komponent a $\lambda_j$ je kořen rovnice

$$|\mathbf{\Sigma_Y} - \lambda \mathbf{I}| = 0\,, \tag{9}$$

kde $\mathbf{I}$ je jednotková matice a $\lambda$ je neznámé číslo.

Z předchozích vztahů plyne, že

$$\sigma^2_{\mathbf{Y}_j} = \lambda_j\,. \tag{10}$$

Kovarianční matice $\mathbf{\Sigma_Y}$ hlavních komponent má tvar

$$\mathbf{\Sigma_Y} = \begin{bmatrix} \lambda_1 & 0 & 0 \ldots & 0 \\ 0 & \lambda_2 & 0 \ldots & 0 \\ & & \vdots & \\ 0 & 0 & 0 \ldots & \lambda_m \end{bmatrix}\,. \tag{11}$$

Matice $\mathbf{L}$ je ortogonální, proto lze snadno výchozí veličiny $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$ vyjádřit pomocí hlavních komponent, tj. $\mathbf{X} = \mathbf{Y}\mathbf{L}^T$.

V konkrétních případech je přesná znalost kovarianční matice základního souboru $\mathbf{\Sigma_X}$ spíše výjimkou. Neznáme-li kovarianční matici $\mathbf{\Sigma_X}$, použijeme výběrovou kovarianční matice $\hat{\mathbf{\Sigma}}_\mathbf{Y}$, jejíž prvky $\hat{\sigma}^2_{ij}$ se určí podle vztahu

$$\hat{\sigma}^2_{ij} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ti} - \bar{x_i})(x_{tj} - \bar{x_j})\,, \tag{12}$$

kde $x_{it}$ je hodnota $i$-té proměnné zjištěna u $t$-tého objektu a $\bar{x_i}$ je průměr $i$-té proměnné za všechny zkoumané objekty.

Využití hlavních komponent vypočtených z kovariančních matic je nejúčelnější v situacích, kdy všechny sledované proměnné mají stejný fyzikální charakter a jsou vyjádřeny ve stejných měrových jednotkách, viz [8], [15], [16].

## 2.5. Míry podobnosti a nepodobnosti

Pro shlukování je velmi důležité určování míry podobnosti, častěji míry nepodobnosti. Uvažujeme dva objekty $\mathbf{x}_i$ a $\mathbf{x}_j$ potom *podobnost* zapisujeme ve tvaru $S(\mathbf{x}_i, \mathbf{x}_j)$, zkráceně $S_{ij}$. Zároveň platí $S_{ij} = S_{ji}$ a nabývá hodnot z intervalu $\langle 0; 1\rangle$, tj.$S_{ii} = 1$.

*Míru nepodobnosti* zapisujeme ve tvaru $D(\mathbf{x}_i, \mathbf{x}_j)$, zkráceně $D_{ij}$, a platí:

1. $D_{ij} \geq 0$ ,
2. $D_{ii} = 0$ ,
3. $D_{ij} = D_{ji}$ .

Pro kvantitativní data se pro vyjádření míry nepodobnosti používají *míry vzdáleností* a je-li splněna trojúhelníková nerovnost

$$D_{ij} + D_{jy} \geq D_{iy} \ (i, j = 1, \ldots, n) \tag{13}$$

mluvíme o *metrice*.

Mezi nejznámější typy vzdáleností patří *euklidovská* $D_E$, *vážená euklidovská vzdálenost* $D_{EW}$, *čtvercová euklidovská vzdálenost* $D_{ES}$, *Čebyševova* $D_C$, *Minkowského* $D_M$, *Manhattanská* $D_B$, *Lanceyova-Williamsova* $D_{LW}$, *Hammingova* $D_H$.

$$D_E(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{m} (x_{il} - x_{jl})^2} = \|\mathbf{x}_i - \mathbf{x}_j\| \,, \tag{14}$$

$$D_{EW}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{m} w_l^2 (x_{il} - x_{jl})^2} \,, \tag{15}$$

$$D_{ES}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{m} (x_{il} - x_{jl})^2 \,, \tag{16}$$

$$D_C(\mathbf{x}_i, \mathbf{x}_j) = \max_l(|x_{il} - x_{jl}|) \,, \tag{17}$$

$$D_M(\mathbf{x}_i, \mathbf{x}_j) = \sqrt[q]{\sum_{l=1}^{m} |x_{ij} - x_{jl}|^q}\,, \tag{18}$$

$$D_B(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{m} |x_{il} - x_{jl}| = |\mathbf{x}_i - \mathbf{x}_j|\,, \tag{19}$$

$$D_{LW}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{m} \frac{|\mathbf{x}_{il} - \mathbf{x}_{jl}|}{|\mathbf{x}_{il}| + |\mathbf{x}_{jl}|}\,, \tag{20}$$

$$D_H(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{m} |\mathbf{x}_i - \mathbf{x}_j|}\,. \tag{21}$$

Vedle vzdáleností se používají *míry podobnosti*, které jsou vhodné pro porovnávání vektorů nezáporných hodnot. Mezi tyto míry patří např. *Kosínova míra* $S_K$, *Jaccardův* $S_J$, *Diceův* $S_D$, *Czekanowského* $S_C$ koeficient ve tvaru:

$$S_K(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{l=1}^{m} x_{il}x_{jl}}{\sqrt{\sum_{l=1}^{m} x_{il}^2 \sum_{l=1}^{m} x_{jl}^2}}\,, \tag{22}$$

$$S_J(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{l=1}^{m} x_{il}x_{jl}}{\sum_{l=1}^{m} x_{il}^2 + \sum_{l=1}^{m} x_{jl}^2 - \sum_{l=1}^{m} x_{il}x_{jl}}\,, \tag{23}$$

$$S_D(\mathbf{x}_i, \mathbf{x}_j) = \frac{2\sum_{l=1}^{m} x_{il}x_{jl}}{\sum_{l=1}^{m} x_{il}^2 + \sum_{l=1}^{m} x_{jl}^2}\,, \tag{24}$$

$$S_C(\mathbf{x}_i, \mathbf{x}_j) = \frac{2\sum_{l=1}^{m} \min\{x_{il}, x_{jl}\}}{\sum_{l=1}^{m} (x_{il} + x_{jl})}\,. \tag{25}$$

V praxi se více využívají keoficienty nepodobnosti,proto je výhodnější míry podobnosti převést na míry nepodobnosti vztahem $D_{ij} = 1 - S_{ij}$. Vzorce pro výpočet měr podobnosti a nepodobnosti jsme převzali z literatury [8], [15] a [16].

# 3. Hierarchické shlukování

Hierarchické shlukování patří mezi základní metody shlukové analýzy. Základním principem těchto metod je postupné přiřazení objektů do jednoho shluku na základě podobnosti.

Rozlišujeme dva přístupy shlukování:

a) *monotetické*, kde se shluky vytvářejí na základě jedné poměnné,

b) *polytetické*, zde bere v úvahu všechny proměnné.

## 3.1. Monotetické shlukování

Monotetická analýza je speciální typ divizioního shlukování pro binární data. Vycházíme z jednoho shluku, který rozdělíme na dva. Pokud máme $m$ proměnných, pak existuje $m$ potencionálních rozdělení objektů do dvou shluků. Dále máme k dispozici $m-1$ možností. Kritériu pro dělení je založeno na závislosti dvou proměnných. Pro sestavení intenzit závislosti mezi $k$-tou a $l$-tou proměnnou se např. používá míra:

$$q_{kl} = |a_{kl}d_{kl} - b_{kl}c_{kl}|\,, \tag{26}$$

kde $a_{kl}$,$b_{kl}$,$c_{kl}$,$d_{kl}$ jsou četnosti v kontingenční tabulce pro proměnné $k$ a $l$. Pro každou $l$-tou proměnnou spočítáme hodnotu

$$q_l = \sum_{k \neq l} q_{kl}, k = 1, 2, \ldots, m. \tag{27}$$

Zařazení objektů je provedeno podle proměnné, u které je dosaženo maxima z daných hodnot, tj. $\max_l(q_l)$. U monotetického shlukování můžeme po provedení analýzy přidat nový objekt, který nebyl v původní analýze. Monotetická analýza je označována za jeden z přístupů konceptuálního shlukování, viz [15].

## 3.2. Polytetické shlukování

U polytetického hierarchického shlukování rozlišujeme dva přístupy a to *aglomerativní* a *divizioní*. U *divizioního* přístupu předpokládáme, že všechny objekty tvoří jeden shluk, ten je pak dělen do více shluků až každý objekt tvoří samostatný shluk. My se dále budeme zabývat *aglomerativním* přístupem.

Na začátku každý objekt tvoří jednoprvkový shluk. Tento počáteční rozklad na $n$ jednoprvkových shluků si označíme jako nultý rozklad $\Omega_0$ množiny objektů.

Dále si zvolime kvantitativní hodnocení podobností vztahů mezi objekty. V prvním kroku vybereme dva shluky, které si jsou nejvíce podobné. Tyto dva shluky spojíme a tak vznikne nový shluk. Nově vytvořený shluk a shluky zbývající tvoří rozklad $\Omega_1$. Další rozklady jsou tvořeny následovně: $\Omega_s$ je $s$-tý rozklad množiny objektů, tedy obsahuje $n-s$ shluků. Tento rozklad se skládá z nového shluku složeného sloučením dvou vzájemně nejpodobnějších shluků předcházejícího $(s-1)$-ního rozkladu $\Omega_{s-1}$ a ostatních nezměněných shluků téhož rozkladu. Nyní stačí spočítat hodnoty podobnosti nového shluku s ostatními shluky rozkladu. Zbývající hodnoty se nemění. Porovnáním hodnot vzájemných podobností u $(n-s)$ shluků z rozkladu $\Omega_s$ najdeme opět dva shluky, které jsou si nejpodobnější, a sloučíme je v jeden. Spolu s ostatními pak tvoří $(s+1)$-ní rozklad $\Omega_{s+1}$ množiny objektů. Tento postup opakujeme dokud nedospějeme k poslednímu rozkladu $\Omega_{n-1}$ o jednom shluku obsahující všechny objekty dané množiny.

Nyní si v několika následujících krocích naznačíme algoritmus hierarchického shlukování:

1. Uvažujeme $n$ objektů, kde každý bod tvoří právě jeden shluk.
2. Vypočítáme vzdálenosti mezi jednotlivými shluky dle vzorečků z kapitoly 3.
3. Najdeme nejmenší vzdálenost mezi dvěmi shluky, dle
$$d[(r),(s)] = \min d[(i),(j)]\,.$$
4. Spojíme vybrané dva shluky do jednoho.
5. Vypočítáme vzdálenost mezi novým shlukem a všemi zbývajícími shluky. Odtud dostaneme novou matici vzdáleností, kde jsme vynechali řádky a sloupce shluků, které jsme spojili, a přidáme řádek a sloupec nového shluku.
6. Blízkost mezi novým shlukem $(r,s)$ a starými shluky můžeme definovat jako:
$$d[(k),(r,s)] = \min\{d[(k),(r)], d[(k),(s)]\}\,.$$
7. Jestliže jsou všechny objekty v jednom shluku, končíme. Jinak se vrátíme na krok 3.

Tento algoritmus je popsán ve zdrojích [1], [8], [15], [16].

U hierarchické metody pro krok č. 3 existuje mnoho přístupů pro určení podobnosti a nepodobnosti shluků, kterými se bude zabývat v následujících podkapitolách, kde jsme základní vzorce převzali z literatury [8] a [15]. Pro lepší znázornění algoritmu jsme použili metodu nejbližšího souseda.

Dále rozlišujeme několik metod pro spojení dvou vybraných shluků. Mezi ně patří *jednoduché propojení*, *úplné propojení* a *průměrné propojení*. U jednoduchého propojení považujeme vzdálenost mezi jedním shlukem a druhým shlukem za nejkratší vzdálenost od jednoho z těchto shluků k ostatním. Úplné propojení shlukování postupuje stejně jako jednoduché propojení s rozdílem, že uvažujeme největší vzdálenost. U průměrného propojení vycházíme z průměrné vzdálenosti daných dvou shluků od ostatních.

Postup shlukování se dá nejlépe znázornit na speciálním grafu: *dendrogramu*. Je to stromový graf, který znázorňuje postupné shlukování jednotlivých objektů a shluků vytvořených v předcházejících krocích. Dendrogram může být jak v horizontální (objekty na ose $Y$), tak vertikální podobě (objekty na ose $X$).

### 3.2.1. Metoda nejbližšího souseda

Nechť $d$ je libovolný koeficient nepodobnosti objektů a $\mathcal{A}$, $\mathcal{B}$ jsou jednotlivé shluky rozkladu $\Omega$. Pak

$$D(\mathcal{A}, \mathcal{B}) = \min_{O_i \in \mathcal{A}, O_j \in \mathcal{B}} \{d(O_i, O_j)\} \tag{28}$$

je koeficient nepodobnosi shluků definovaný na základě *metody nejbližšího souseda* (nearest neighbour), viz [8], [15], [16]. Tato metoda splňuje všechny podmínky pro výpočet vzdáleností.

### 3.2.2. Metoda nejvzdálenějšího souseda

U této metody je určující maximální vzdálenost objektů. Stejně jako u metody nejbližšího souseda, zde používáme koeficient nepodobnosti $d$, shluky $\mathcal{A}$, $\mathcal{B}$

rozkladu $\Omega$. Pak pro $\mathcal{A} \neq \mathcal{B}$ platí:

$$D(\mathcal{A}, \mathcal{B}) = \max_{O_i \in \mathcal{A}, O_j \in \mathcal{B}} \{d(O_i, O_j)\}\,, \tag{29}$$

$$D(\mathcal{A}, \mathcal{A}) = 0. \tag{30}$$

Tato metoda je známa jako *furthest neighbour* ([8], [15], [16]).

### 3.2.3. Centroidní metoda

Pro centroidní metodu se využívá výpočet koeficientu nepodobnosti jako čtvercové euklidovské vzdálenosti $d_{ES}$, vzorec (16), neboli $d_E^2$ vzorec (14). Je to výhodnější než samotný výpočet vzdálenosti $d_E$ a to pro zjednodušení výpočtu hodnot nepodobnosti shluků a jednak pro odvození rekurzivního schématu výpočtu těchto hodnot. Nyní nadefinujeme koeficient nepodobnosti centroidní metodou:

Nechť $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$, $\mathcal{B} = \{B_1, B_2, \ldots, B_l\}$ jsou shluky rozkladu $\Omega$, $A_i = (a_{i1}, a_{i2}, \ldots, a_{ip})$, $i = 1, 2, \ldots, k$ jsou objekty shluku $\mathcal{A}$, $B_i = (b_{i1}, b_{i2}, \ldots, b_{ip})$, $i = 1, 2, \ldots, l$ jsou objekty shluku $\mathcal{B}$.

Nechť $\bar{A}$ a $\bar{B}$ jsou těžiště shluků $\mathcal{A}$ a $\mathcal{B}$, tj.:

$$\bar{A} = (\bar{a_1}, \bar{a_2}, \ldots, \bar{a_p})\ , \bar{a_j} = \frac{1}{k} \sum_{i=1}^{k} a_{ij}\,, \tag{31}$$

$$\bar{B} = (\bar{b_1}, \bar{b_2}, \ldots, \bar{b_p})\ , \bar{b_j} = \frac{1}{k} \sum_{i=1}^{k} b_{ij}\,, \tag{32}$$

pro $j = 1, 2, \ldots, p$.

Potom

$$D = (\mathcal{A}, \mathcal{B}) = d_E^2(\bar{A}, \bar{B}) \tag{33}$$

je koeficient nepodobnosti shluků pro centroidní metodu ([8], [15]).

Kromě této metriky můžeme použít i jiné metriky jako míry vzdálenosti těžiště.

### 3.2.4. Metoda průměrné nepodobnosti objektů

Uvažujeme shluky $\mathcal{A} = A_1, A_2, \ldots, A_k$, $\mathcal{B} = B_1, B_2, \ldots, B_l$ rozkladu $\Omega$, $k = ||\mathcal{A}||$, $l = ||\mathcal{B}||$ jsou počty objektů shluků $\mathcal{A}$, $\mathcal{B}$. Potom pro $\mathcal{A} \neq \mathcal{B}$ je

$$D(\mathcal{A}, \mathcal{B}) = \frac{1}{kl} \sum_{(i,j)}^{k*l} d(A_i, B_j)\,, \tag{34}$$

a

$$D(\mathcal{A}, \mathcal{A}) = 0 \tag{35}$$

koeficient nepodobnosti objektů pro metodu průměrné vazby (nepodobnosti objektů), viz [15].

### 3.2.5. Mediánová metoda

Tato metoda byla vytvořena na základě centroidní metody, neboť pan Gower (zakladatel mediánové metody) přišel s názorem, že rozdílné počty objektů shluků způsobují rozdílné váhy prvních dvou složek předpisu centroidní metody, tím se vlastnosti malých shluků v konečném sjednocení ztrácejí. Tato metoda je nazvána mediánová, neboť spojnice těžiště shluku $\mathcal{U}$ s těžištěm shluku $\mathcal{R} = \mathcal{P} \cup \mathcal{L}$ leží na střední příčce trojúhelníku, jehož vrcholy jsou těžiště shluků $\mathcal{L}, \mathcal{P}, \mathcal{U}$.

Potom koeficient nepodobnosti $d$ pro mediánovou metodu ([15])je dán vztahem:

$$D(O_i, O_j) = d_E^2(O_i, O_j), \tag{36}$$

$$D(\mathcal{U}, \mathcal{R}) = \frac{1}{2} D(\mathcal{U}, \mathcal{P}) + \frac{1}{2} D(\mathcal{U}, \mathcal{L}) - \frac{1}{4} D(\mathcal{P}, \mathcal{L}). \tag{37}$$

# 4. Metody rozkladu

Podstatou metod rozkladu neboli nehierarchického shlukování je přiřazení objektů do předem stanoveného počtu disjunktních shluků. V současné době existuje mnoho pohledů na klasifikaci metodami rozkladu jako metody matematického programování, optimalizační metody (iterativní relokační algoritmy), hybridní

klasifikaci a další. Nehierarchické metody můžeme rozdělit na *metody pevného shlukování* (objekt je přiřezen do shluku či ne) a na *fuzzy analýzu* (každému objektu přiřazena míra příslušnosti ke každému shluku). Dále můžeme dané metody dělit na *metody založené na vzdálenostech* (fuzzy shlukování, metoda $k$-medoidů) a *metody vektorového prostoru* (metoda $k$-průměrů).

Počet shluků můžeme stanovit na základě zkušenosti s předchozími daty. Využívá se expertní zadání nebo výpočet optimálního počtu shluků, kterému se budeme věnovat později. V následujících podkapitolách vycházíme z literatury [1], [5], [8], [15] a [16].

## 4.1. Metoda $k$ průměrů

Metoda $k$ průměrů je jeden z nejjednodušších algoritmů, který řeší problém shlukování. Procedura podává jednoduchý a snadný způsob klasifikace daného datového souboru prostřednictvím daného počtu shluků. Hlavní myšlenka je definovat $k$ centroidů, jeden pro každý shluk. Tyto centroidy by měly být umístěny na vhodné pozici, jelikož různá umístění nám mohou dát různé výsledky. Další krokem je vzít každý bod z datového souboru a přiřadit jej k nejbližšímu centroidu. Následně přepočítáme $k$ nových centroidů jako barycentrum shluků ($m$-rozměrný vektor průměrných hodnot jednotlivých proměnných odpovídající dané skupině shluků). Po výpočtu nových centroidů uděláme opět spojení mezi body datového souboru s danými cetroidy. Tento proces budeme opakovat, dokud bude docházet ke změnám ve shlucích, tj. dokud se bude měnit umístění centroidů.

Algoritmus usiluje o minimalizaci funkce, v tomto případě čtvercové chyby:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^{(j)} - c_j||^2 \,, \tag{38}$$

kde $|x_i^{(j)} - c_j|^2$ je volba vzdálenosti mezi bodem z datového souboru $x_i^{(j)}$ a středem shluku $c_j$. $J$ je ukazatel vzdálenosti pro $n$ bodů od jejich příslušného středu shluku, viz [1], [15].

Nyní si v několika krocích popíšeme algoritmus $k$ průměrů:

1. Umístíme $k$ bodů do prostoru reprezentující objekty, které jsou shlukovány. Tyto body reprezentují počáteční skupinu centroidů.
2. Každý objekt přiřadíme do skupiny, která obsahuje nejbližší centroid vzhledem k danému bodu.
3. Pokud jsou všechny objekty přiřazeny, přepočítáme všechny pozice $k$ cetroidů.
4. Vrátíme se na krok 2 a 3 dokud se centroidy budou pohybovat.

Ačkoliv se může prokázat, že procedura bude ukončena, algoritmus $k$ průměrů nemusí nezbytně najít nejoptimálnější řešení odpovídající minimu dané funkce. Algoritmus je také citlivý na počáteční volbu středů shluků (centroidů). Tento algoritmus může být viděný jako nenasytný pro dělení $n$ reprezentantů do $k$ shluků jako minimalzování součtu vzdálenosti čtveců od středu shluku. Metoda je vhodná pro velké datové soubory neboť není založena na matici vzdáleností.

Algoritmus a jeho slabosti:

* Způsob spouštěná algoritmu není nijak specifikovaný. Obvykle se na počátku volí $k$ bodů z daného souboru jako $k$ centroidů.
* Výsledek procedury závisí na počáteční volbě centroidů. Standardní řešení dostáváme, pokud zvolíme několik rozdílných počátečních řešení.
* Výsledek závisí na hodnotě $k$.

## 4.2. Metoda $k$ medoidů

Algoritmus je určen pro kvantitativní data a vychází z počátečního rozdělené do $k$ shluků jako předchozí metoda. Každému shluku je přiřazen *medoid*, což je konkrétní objekt z daného shluku. Počáteční medoid je určen součtem vzdáleností jednotlivých objektů ve shluku od vybraného objektu tak, aby vzdálenost byla minimální, viz [8], [15], [16].

Dále zkoumáme všechny objekty. Je-li vzdálenost objektu od svého medoidu nejmenší, pak je v tomto shluku ponechán. V opačném případě se přiřadí

do shluku, ve kterém má objekt nejblíže k příslušnému medoidu. Tzn. objekt $\mathbf{x}_i$ je přiřazen do shluku $C_g$ s medoidem $m_g$ platí-li:

$$D(\mathbf{x}_i, m_g) \leq D(\mathbf{x}_i, m_u) \ , u = 1, 2, \ldots, k \ , \tag{39}$$

kde $m_u$ je libovolný jiný medoid.

V dalším kroku jsou medoidy stanoveny minimalizací funkce, která je dána součtem vzdáleností jednotlivých objektů od medoidu v příslušném shluku. Je třeba, aby bylo dosaženo minima funkce

$$f = \sum_{i=1}^{n} D(\mathbf{x}_i, m_{g,i}) \, , \tag{40}$$

kde $m_{g,i}$ je medoid v $g$-tém shluku, kterému je přiřazen objekt $\mathbf{x}_i$. Postup opakujeme dokud klesá hodnota funkce.

## 4.3. Fuzzy c-means shlukování

Fuzzy $c$-means je metoda, která dovoluje jednomu druhu dat patřit do dvou nebo více shluků. Metoda je založena na minimalizaci funkce:

$$J_d = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^d ||\mathbf{x}_i - c_j||^2 \ , 1 \leq d < \infty \, , \tag{41}$$

kde $d$ je reálné číslo větší jak 1,
$u_{ij}$ je míra příslušnosti pro každý objekt $\mathbf{x}_i$ a $j$-tý shluk,
$u_{ij} \in \langle 0; 1 \rangle$,
$\mathbf{x}_i$ je $i$-tý $m$-rozměrný měřený prvek,
$c_j$ je $m$-rozměrný střed shluku,
$|| * ||$ je norma vyjadřující podobnost mezi měřenými daty a středy shluků.

Fuzzy rozdělení je uskutečňováno skrz iterační optimalizaci funkce $J_d$ s aktualizací stupně příslušnosti $u_{ij}$ a středu shluku $c_j$ pomocí vztahů:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} (\frac{||\mathbf{x}_i - c_j||}{||\mathbf{x}_i - c_k||})^{\frac{2}{d-1}}} \, , \tag{42}$$

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}^d \mathbf{x}_i}{\sum_{i=1}^{n} u_{ij}^d} \,. \tag{43}$$

Míra příslušnosti musí splňovat následující podmínky:

$$0 \leq u_{ij} \leq 1 \ , i = 1, \ldots, n \ , j = 1, \ldots, c$$

$$\sum_{j=1}^{c} u_{ij} = 1 \ , i = 1, \ldots, n \,.$$

Tato iterace bude ukončena pokud $\max_{ij}\{|u_{ij}^{k+1} - u_{ij}^{k}|\} < \epsilon$, kde $\epsilon$ je ukončovací kritérium, $\epsilon \in (0; 1)$, a $k$ jsou kroky iterace. Tato procedura konverguje k lokálnímu minimu nebo sedlovému bodu $J_d$.

Algoritmus je složen z následujících kroků:

1. Zahájení procesu $U = [u_{ij}]$, počáteční hodnotu $U^{(0)}$.
2. V $k$-tém kroku: vypočítáme středové vektory $C^{(k)} = [c_j]$ s $U^{(k)}$ dle (43).
3. Aktualizujeme $U^{(k)}$ a $U^{(k+1)}$ pomocí vzorce (42).
4. Jestliže $||U^{(k+1)} - U^{(k)}|| < \epsilon$, pak končíme. Jinak se vrátíme na krok 2.

Na začátku sestavíme vhodnou matici $U$, jejíž složky nabývají hodnoty z intervalu $\langle 0; 1\rangle$, reprezentující stupeň příslušnosti mezi daty a středy shluků, viz [1], [8], [15].

V následující kapitole si naznačíme metodu matematického programování, která patří mezi metody rozkladu, ale je méně využívána.

## 4.4. Aplikace matematického programování

Matematické lineární programování využíváme pro pevné shlukování definované pomocí binárních dat. Zde si uvedeme dva přístupy ([15]). První vychází ze dvou množin proměnných:

* proměnných $u_{ih}$, které vyjadřují přiřazení $i$-tého objektu k $h$-tému shluk ($u_{ih} = 1$ pokud $i$-tý objekt patří do $h$-tého shluku a $u_{ih} = 0$ pokud nepatří),

* a $y_{ij}$, které udávají, zda $i$-tý a $j$-tý objekt patří do stejného shluku ($y_{ij} = 1$ pokud patří a $y_{ij} = 0$ pokud nepatří do stejného shluku).

Úkolem je najít takové dvě množiny binárních proměnných, které tvoří optimální rozklad. Tento úkol specifikujeme jako úlohu matematického programování, tedy minimalizujeme součet vnitroskupinových vzdáleností. Vše vyjádříme pomocí minimalizace funkce:

$$f = \sum_{i=2}^{n} \sum_{j=1}^{i-1} D_{ij} y_{ij} \ , \tag{44}$$

za podmínek

$$y_{ij} \geq u_{ij} + u_{jh} - 1 \ (1 \leq j < i \leq n; h = 1, \ldots, k),$$

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

$$u_{ih} \in \{0; 1\} \ (i = 1, \ldots, n; h = 1, \ldots, k) \, ,$$

$$y_{ij} \geq 0 \ (1 \leq j < i \leq n) \, .$$

U druhého přístupu se využívá toho, že $j$-tý objekt je medoidem určitého shluku. Proměnné $u_{ij}$ představují příslušnost $i$-tého objektu ke shluku, jehož medoidem je $j$-tý objekt, tzn. $u_{jj} = 1$, pokud je $j$-tý objekt vybrán jako jeden z medoidů, jinak $u_{jj} = 0$. Minimalizuje se funkce:

$$f = \sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} u_{ij} \, , \tag{45}$$

za podmínek

$$u_{ij} \leq u_{jj} \ (i, j = 1, \ldots, n) \, ,$$

$$\sum_{j=1}^{n} u_{ij} = 1 \ (i = 1, \ldots, n) \, ,$$

$$\sum_{j=1}^{n} u_{jj} = k \, ,$$

$$u_{ij} \in \{0; 1\} \ (i, j = 1, \ldots, n) \, .$$

Problém patří k úlohám celočíselného programování.

## 4.5. Optimální počet shluků

Velkým problémem může být určení optimálního počtu shluků, do kterých mají být příslušné objekty přiřazeny. Existuje několik přístupu. Jedním z nich jsou složité matematické výpočty. O jednom z nich se můžeme blíže dočíst např. v literatuře [8], [15], který je zde podrobně popsán.

Dalším způsobem může být metoda pokusu, kdy postupně zkoušíme objekty přiřadit mezi různý počet shluků. Zde je nutné výpočty několikrát opakovat, jelikož u většiny metod rozkladu jsou na začátku voleni reprezentanti shluků náhodně. Na základě těchto výpočtů se rozhodneme, jaký počet shluků je pro daná data optimální.

# 5. Alternativní přístupy

V této kapitole se budeme zabývat biologicky inspirovanými algoritmy, které se používají při shlukové analýze. Tyto metody mají inspiraci v přírodě a jsou alternativou matematických a heuristických metod. Dále se zaměříme na *umělé neuronové sítě* a *genetické algoritmy*.

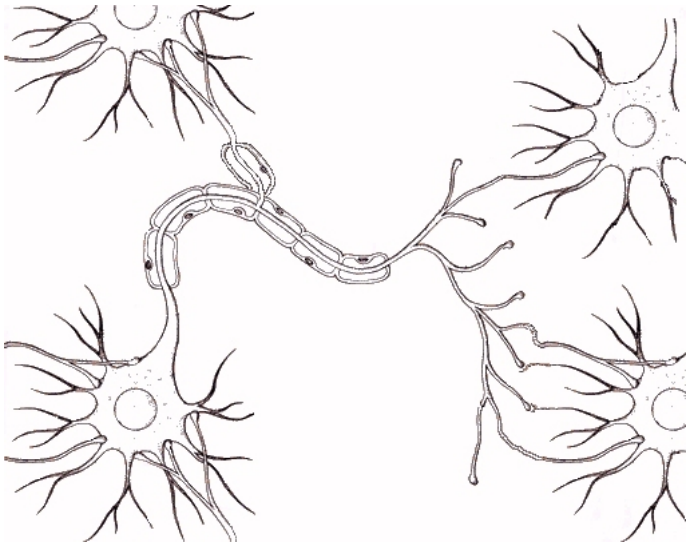
## 5.1. Umělé neuronové sítě

Neuronové sítě se využívají v oblastech, kde se vyskytuje problém predikce a klasifikace, který je těžko řešitelný pomocí statistických metody. V současné době se využívá v mnoha oborech jako finance, medicína, biologie a další. Velkou výhodou neuronových sítí je jednoduchá adaptace na danou situaci. Existuje mnoho modelů neuronových sítí, jako např. *Kohonenův model samoorganizujících se map*, *Hopfieldův model*, *adaptivní rezonanční teorie*, atd. Některé z nich prozkoumáme podrobněji.

**Biologická inspirace**

Umělé neuronové sítě vycházejí z výzkumu umělé inteligence, napodobení živé inteligence. Základním prvkem je neuron, což je specifikovaná buňka schopna zpracovávat signály. Lze jej rozložit na dendrity (vstupy), tělo a axon (výstup). Jeho model se analogicky skládá z několika vstupů ($X$), které mají přiřazenou číselnou váhu ($w$), udávající tzv. sílu daného vstupu. Vstupy jsou v modelu složeny (nejčastěji součtem vážených hodnot vstupů) a výsledek je jako argument předán nelineární prahové funkci. Proces učení neuronové sítě spočívá ve vytvoření neuronů, jejich vzájemnému zapojení a vhodnému nastavení všech vah.

Umělý neuron pracuje podobně jako biologický, ale pouze pokud celkový počet všech signálů překročí určitou úroveň (aktivační práh).



Obrázek 2: Biologický neuron

Základní myšlenkou řešení problémů umělé inteligence je konstrukce modelů neuronového systému živých organizmů, složených propojením umělých neuronů.

**Formální neuron**

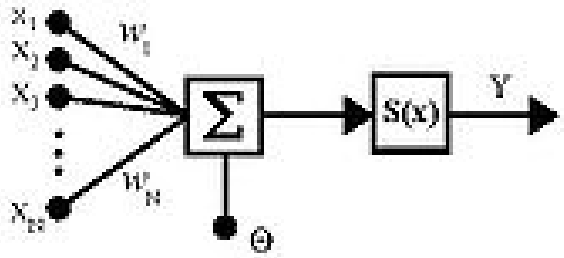
*Formální neuron* je zjednodušený matematický model neuronu. Biologický neuron přijímá informaci od ostatních neuronů přes synaptické vazby a dál je předává velkému množství neuronů. Signály vstupující do neuronů se sčítají. Takovýto součet tvoří postsynaptický potenciál dosahující prahové úrovně, tzn. na výstupu z neuronu se objeví jednotkový signál ($> 0$). Tento signál se šíří

k neuronům, se kterými má synaptickou vazbu. Model neuronu se skládá ze dvou funkčních bloků: *sumátoru* a *bloku realizující prahovou funkci.*

Do sumátoru vstupují signály od ostatních neuronů $X_i, (i = 1, \ldots, n)$. Síla vazby je dána vahami $w_i$, součet vážených signálů je tzv. postsynaptický potenciál $\xi$, což je vstupní veličina prahové funkce $f$, jejímž parametrem je práh $\vartheta$. Prahová fukce nabývá hodnot:

$$f = \begin{cases} 1\,, & \xi > \vartheta \\ -1\,, & \xi < \vartheta\,. \end{cases} \tag{46}$$

Formální neuron realizuje zobrazení $y = f_\vartheta(w_i, x_i)$, kde $x_i$ je ohodnocení neuronu $X_i$, $y$ je ohodnocení neuronu $Y$. Jedná se tedy o diskrétní model, kde vstupy do neuronu $Y$ nabývají hodnot $x_i \in \{-1; 1\}$ a i váhy nabývají celočíselných hodnot.




Obrázek 3: Formální neuron

**Umělá neuronová síť**

Neuronovou síť, složenou z formálních neuronů, popisujeme pomocí dynamiky orientovaného grafu s ohodnocenými hranami a uzly. Uzly grafu reprezentující neurony tvoří množinu $\{X_1, \ldots, X_n\}$, která s množinou hran udává topologii sítě. K $i$-tému neuronu přísluší ohodnocení $x_i$ (vnitřní potenciál) a $\xi_i$ (stav, výstup neuronu) připisující ke všem hranám $(i, j)$ vystupující z $i$-tého uzlu a vstupující do $j$-tého uzlu, $\alpha_i$ (vstup, stimul).

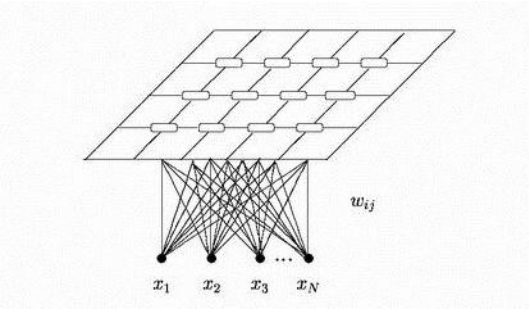
V problematice umělých neuronových sítí rozlišujeme algoritmy s učením *bez učitele*, *s učitelem*, *upevňované učení*, *kombinované*. My se dále budeme zabývat algoritmy s učením bez učitele, kdy jsou dána data $X$, cenová funkce, což je libovolná funkce dat $X$ a výstupu $Y$, která má být minimalizovaná. O jednotlivých

kategorií se můžeme více dočíst v literatuře [6], [7], [15] a [17].

### 5.1.1. Kohonenovy mapy

Kohonenovy mapy se nazývají také samoorganizující mapy SOM (Self - Organizing Maps). Tato metoda se převážně aplikuje na vyhledávání shluků v textových dokumentech a také na detekci nových statistických souborů. To je možné díky tomu, že SOM v průběhu vytváří reprezentanty shluků a také příslušné váhy. Jedním z použití SOM je explorační analýza dat, kdy neuronová síť je schopna rozpoznávat shluky dat a také přiřadit k sobě podobné třídy. Sítě SOM se mohou používat pro klasifikaci, pokud je dopředu znám počet shluků. Model SOM transformuje vstupní stavový vektor $\mathbb{R}^n$ do libovolné dimenze, většinou $\mathbb{R}^n \rightarrow \mathbb{R}^2$.

Síť SOM je tvořena 2 vrstvy, a to vstupní a výstupní jednotky. Kohonenova vrstva (vrstva topologické mapy) je tvořena neurony, které jsou uspořádány do topologické struktury (obvykle dvourozměrné mřížky). Topologická struktura nám určuje, které neurony se sebou sousedí. Každému neuronu přísluší vektor vah o stejné dimenzi ([6], [7], [15], [17]).




Způsob zjišťování pozic odezvy v SOM nazýváme soutěžení. Výsledkem soutěžení v každém kroku dostáváme vítězný neuron, který nejvíce reaguje na vstup $\mathbf{x}$. Maximum výstupu lineárního neuronu je možné najít pomocí $i^* = arg\max_i(\mathbf{w}_i^T\mathbf{x})$, $i^*$ je index vítězného neuronu. Zde se využívá výpočtu euklidovské vzdálenosti, vzorec (14). Po předložení vstupního vektoru od vstupní vrstvy nejdříve spočítáme vzdálenost tohoto vektoru od všech ostatních neuronů dle:

$$i^* = arg\min_i \|\mathbf{x} - \mathbf{w}_i\|, \tag{47}$$

kde norma $(\mathbf{x}-\mathbf{w}_i)$ je minimální, $i^*$ je index. Další vztahy převezmeme z literatury [3].

Jakmile najdeme vítěze, je potřeba adaptovat váhy. Algoritmus zaručí posun vahového vektoru vítězného vektoru a jeho topologických sousedů k aktuálnímu vstupu dle vztahu:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t)h(i^*, i)[\mathbf{x}(t) - \mathbf{w}_i(t)]\,, \tag{48}$$

kde funkce $\alpha(t) \in \langle 0; 1\rangle$ je učící parametr, jehož hodnota klesá v čase, čímž se ukončí proces ([16]). Funkce okolí $h(i^*, i)$ definuje rozsah spolupráce mezi neurony, nejjednodušší způsob je:

$$h(i^*, i) = \begin{cases} 1\,, & \text{pokud } D_M(i*, i) \leq \alpha(t) \\ 0\,, & \text{jinak}\,, \end{cases} \tag{49}$$

kde $D_M(i*, i)$ je manhattanská vzdálenost, vzorec (19), mezi neurony $i^*$ a $i$ v mřížce map ([15]).

V několika krocích si naznačíme algoritmus SOM:

**Vstup:** trénovací množina $\mathbf{x}_i$, $i = 1, 2, \ldots, n$.

**Výstup:** vektory vah $\mathbf{w}_j$, $j = 1, 2, \ldots, K$.

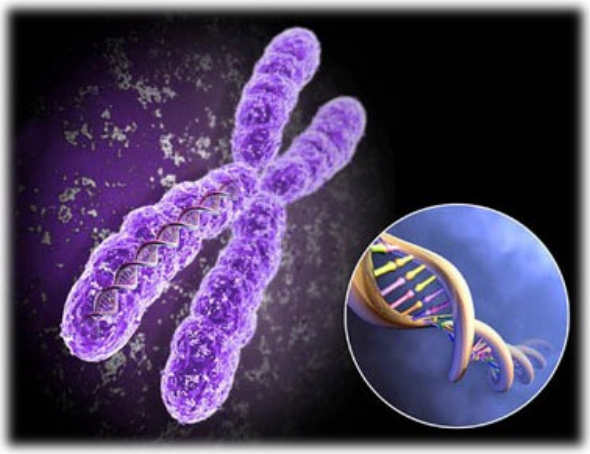
1. inicializuj náhodně $\mathbf{w}_j$, $i = 0$,
2. vyber $\mathbf{x}_i$ a najdi nejbližší neuron $i^*$ ,
3. $\|\mathbf{x}_i - \mathbf{w}_{i^*}\| = \min_j \|\mathbf{x}_i - \mathbf{w}_j\|$,
4. uprav všechny vahové vektory (pro všechna $j$): $\mathbf{w}_i = \mathbf{w}_j + \alpha h_{i^*j}(\mathbf{x}_i - \mathbf{w}_j)$, kde $\alpha$ je koeficient útlumu, $h_{i^*j}$ funkce okolí; pak $i = i + 1$,
5. opakuj kroky 3 až 5 dokud není splněno kritérium zastavení.

## 5.2. Genetické algoritmy

Genetické algoritmy (GA) patří mezi stochastické optimalizační metody. Jsou odvozeny na základě biologické genetiky a teorie evoluce, napodobují přirozený výběr, křížení a mutaci živých organismů. GA se snaží najít řešení složitých problémů, pro které neexistují vhodné exaktní algoritmy.

Hlavním principem GA je postupná tvorba nových generací. V iteracích vzniká tzv. populace, jejíž každý jedinec představuje jedno řešení. V průběhu algoritmu se pro každého jedince vyčísluje hodnota cenového funkcionálu, který chceme minimalizovat. V GA se tento funkcionál nazývá fitness funkce.

Jedinec se nazývá fenotyp. Jedince reprezentujeme chromozomem, který se dělí na geny. Geny nabývají různých hodnot (tzv. alely). Alela je obvykle reprezentována binárním číslem. Gen je pak řetězec jedniček a nul. Při softwarovém zpracování se pro uložení fenotypů využívá metoda stromu, matice, pole atd. V každé iteraci jsou jedinci následně modifikování a vzniká nová populace. Do nové populace jsou vybíráni nejlepší jedinci na základě hodnoty jejich fitness funkce. Většinou se za fitness funkci používá počet jedniček v chromozomu.



Obrázek 4: Chromozom, viz [2]

U GA používáme následující evoluční procesy:

**Selekce**

- proces, kde jsou chromozomy kopírovány do nové populace podle jejich účelové funkce; tj. chromozomy s větší hodnotou účelové funkce jsou převzaty s větší pravděpodobností,
- formálně vyjádřeno:

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j},$$

kde $p_i$ je pravděpodobnost reprodukce $i$-tého chromozomů, $f_i$ je hodnota fitness funkce $i$-tého chromozomu, $\sum f_j$ je celkové ohodnocení všech chromozomů.

**Křížení**

- nejdříve jsou náhodně vytvořeny páry z vybraných chromozomů,
- následně probíhá výměna informace mezi páry chromozomů, tj. náhodně zvolíme gen v chromozomu, od kterého dochází k výměně zbylé části,
- pro lepší představu si vše ukážeme na jednoduchém příkladu:
máme dané 4 chromozomy a jejich ohodnocení (počet jedniček):

| číslo | chromozom | ohodnocení |
|---|---|---|
| 1 | (1,1,0,1,0,0,1,0) | 4 |
| 2 | (1,1,1,1,0,0,1,1) | 6 |
| 3 | (0,0,1,0,1,1,0,0) | 3 |
| 4 | (1,0,0,0,1,0,0,0) | 2 |

náhodně vybereme páry chromozomů a následně provedeme výměnu části chromozomů:

| číslo | chromozom | křížení | nový chromozom |
|---|---|---|---|
| 2 | $(1,1,1\|1,0,0,1,1)$ | $\rightarrow$ | $(1,1,1,1,0,0,1,0)$ |
| 1 | $(1,1,0\|1,0,0,1,0)$ | $\rightarrow$ | $(1,1,0,1,0,0,1,1)$ |
| 2 | $(1,1,1,1\|0,0,1,1)$ | $\rightarrow$ | $(1,1,1,1,1,1,0,0)$ |
| 3 | $(0,0,1,0\|1,1,0,0)$ | $\rightarrow$ | $(0,0,1,0,0,0,1,1)$ |

- křížení necháváme proběhnout obvykle s pravděpodobností $0.75 - 0.95$.

**Mutace**

- operátor, který v daném procesu prochází jednotlivé chromozomy a s danou pravděpodobností mění jejich geny,
- mutace nastává s pravděpodobností $0.001 - 0.05$,
- v našem případě se změnila čtvrtá hodnota u prvního chromozomu a druhá hodnota u čtvrtého chromozomu:

| | | |
|---|---|---|
| $(1,1,1,1,0,0,1,0)$ | $\rightarrow$ | $(1,1,1,\mathbf{0},0,0,1,0)$ |
| $(1,1,0,1,0,0,1,1)$ | $\rightarrow$ | $(1,1,0,1,0,0,1,1)$ |
| $(1,1,1,1,1,1,0,0)$ | $\rightarrow$ | $(1,1,1,1,1,1,0,0)$ |
| $(0,0,1,0,0,0,1,1)$ | $\rightarrow$ | $(0,\mathbf{1},1,0,0,0,1,1)$ |

Základní informace o GA jsou popsány v literatuře [7], [9], [15], [17].

GA budeme využívat na shlukování. GA lze využít pro optimální rozdělení dat do shluků. Ve shlukování pomocí GA vycházíme z metody $k$ průměrů, kde máme $n$ jedinců, které chceme přiřadit do $k$ shluků. Snažíme se zařadit jedince do shluků tak, aby byla minimalizována vnitřní variabilita shluků. Vycházíme z následujících vzorců, převzatých z literatury [5].

Máme danou množinu $n$ objektů $\{\mathbf{x}_i; i = 1, \ldots, n\}$, kde $x_{ij}$, $j = 1, \ldots, m$ je hodnota $j$-té proměnné $i$-tého objektu. Dále uvažujeme váhy

$$w_{ih} = \begin{cases} 1\,, & \text{i-tý objekt součástí h-tého shluku} \\ 0\,, & \text{jinak}\,, \end{cases} \tag{50}$$

pro $i = 1, \ldots, n$ a $h = 1, \ldots, k$, kde $\mathbf{W} = [w_{ih}]$ je matice vah s vlastnostmi $w_{ih} = \{0, 1\}$ a $\sum_{h=1}^{k} w_{ih} = 1$.

Centroid (střed) $h$-tého shluku je dán vzorcem:

$$c_{hl}(\mathbf{W}) = \frac{\sum_{i=1}^{n} w_{ih} x_{il}}{\sum_{i=1}^{n} w_{ih}}\,. \tag{51}$$

Vnitřní variabilita $h$-tého shluku je dána:

$$S^h(\mathbf{W}) = \sum_{i=1}^{n} w_{ih} \sum_{l=1}^{m} (x_{il} - c_{hl})^2 \tag{52}$$

a celková vnitroshluková variabilita shluků:

$$S(\mathbf{W}) = \sum_{h=1}^{k} S^h(W)\,. \tag{53}$$

Vzdálenost mezi centroidem a daným objektem se počítá pomocí euklidovské vzdálenosti (14). Cílem je najít matici $\mathbf{W}^*$ minimalizující součet čtverců vzdáleností objektů ve shlucích od centroidu. V následující podkapitole se více seznámíme s konkrétní metodou genetického algoritmu $k$ průměrů.

### 5.2.1. Genetický algoritmus $k$-průměrů

GA $k$ průměrů vychází z metody rozkladu $k$ průměrů, ale využívá genetické operátory selekce, mutace založené na vzdálenosti a operátoru $k$ průměrů zajišťující rychlou konvergenci celého procesu. GA vychází z populace zakódovaných řešení, jež je inicializována náhodně. Nyní se blíže seznámíme s jednotlivými částmi GA.

#### Kódování

Stavový prostor dané úlohy je tvořen všemi maticemi $\mathbf{W}$ splňující podmínku $w_{ih} = \{0, 1\}$ a $\sum_{h=1}^{k} w_{ih} = 1$. Každá alela koresponduje s určitým objektem a její hodnota odpovídá pořadovému číslu shluku, do něhož objekt náleží. Vycházíme z (50).

#### Inicializace

Každého jedine náhodně přiřadíme k číslu shluku (diskrétní rovnoměrné rozdělení od 0 do $k$).

#### Operátor selekce

Operátor náhodně vybírá chromozomy z předešlé populace pomocí distribuce

$$p(s_i) = \frac{F(s_i)}{\sum_{j=1}^{n} F(s_j)} \ , \tag{54}$$

kde $F(s_i)$ udává hodnotu účelové funkce pro dané řetězce v populaci.

Je třeba, aby každé řešení (jedinec) v populaci bylo ohodnoceno. Toto ohodnocení udává jeho schopnost přežít nebo přispět ke zlepšení životaschopnosti příští generace. Hodnota fitness funkce jedince zakódovaného řetězcem $s_{\mathbf{W}}$ závisí na celkové vnitřní shlukové variabilitě $S(\mathbf{W})$. Cílem je minimalizovat $S(\mathbf{W})$, tedy řešení s relativně menším součtem čtverců odchylek uvnitř shluku musí mít relativně větší hodnotu účelové funkce.

My zde budeme dále používat mechanismus $\delta$ oříznutí převzatý z literatury [15]:

Nechť

$$f(s_{\mathbf{W}}) = -S^h(\mathbf{W}) \ ,$$

$$g(s_{\mathbf{W}}) = f(s_{\mathbf{W}}) - (\bar{f} - c\delta) \ ,$$

kde $\bar{f}$ je střední hodnota a $\delta$ směrodatná odchylka funkce $f(s_{\mathbf{W}})$ ve stávající populaci a $c$ je konstanta z intervalu $\langle 1; 3\rangle$. Potom je hodnota účelové funkce jedince $F(s_{\mathbf{W}})$ dána vztahem

$$F(s_{\mathbf{W}}) = \begin{cases} g(s_{\mathbf{W}}) \,, & \text{jestliže } g(s_{\mathbf{W}}) \geq 0, \\ 0 \,, & \text{jinak} \,. \end{cases} \tag{55}$$

**Operátor mutace**

Mutace mění hodnotu nějaké alely v závislosti na vzdálenosti centroidu shluku od odpovídajícího objektu. Operátor je definován tak, že pravděpodobnost změny hodnoty alely na shluk je vyšší, pokud je centroid korespondujícího shluku blíže k objektu. Zde budeme dále využívat euklidovskou vzdálenost mezi objektem $\mathbf{x}_i$ a centroidem $\mathbf{c}_h$. Pak je ale zaměněná hodnotou náhodně vybranou z distribuce:

$$p_j = P\{s_{\mathbf{W}}(i) = j\} = \frac{c_m D_{\max} - D_j}{\sum_{h=1}^{k} (c_m D_{\max} - D_h)} \ , \tag{56}$$

kde $c_m$ je s hodnotou mezi 1 a $D_{\max} = \max_h\{D_h\}$. Čím menší bude počet shluků, tím větší bude hodnota čtvercové euklidovské míry. Díky tomu bude zabráněno vzniku prázdných shluků.

**Operátor $k$ průměrů**

Pro zrychlení výpočtu se používá jednokrokový operátor $k$ průměrů. Nechť $s_{\mathbf{W}}$ je řetězec, pak operátor $k$ průměrů je tvořen dvěma kroky, které po aplikaci na $s_{\mathbf{W}}$ dávají lepší $s_{\mathbf{W}^*}$:

1. spočítáme středy shluků dle (51) pro danou matici $\mathbf{W}$,
2. přeuspořádáme objekty, aby patřili do shluků, jejichž vzdálenost od středu shluku je nejmenší, následně vypočítáme novou matici $\mathbf{W}^*$.

Nyní si v několika krocích naznačíme GA $k$ průměrů:

**Vstup:** pravděpodobnost mutace $p_m$, počet jedinců v populaci $n$, maximální počet generací MAXGEN.

**Výstup:** výsledné řešení - řetězec $s$.

{Inicializuj populaci $P_0$;
$geno = \text{MAXGEN}$;
$s = P_1$; ($P_i$ je $i$-tý řetězec v $P_0$);
dokud ($geno > 0$)
{Vypočti hodnoty fitness řetězců v $P_0$;
$P^*$=Selekce($P$);
pro $i = 1$ do $n$; $P_i = \text{Mutace}(P_i^*)$;
pro $i = 1$ do $n$; $k$ průměrů ($P_i$);
$s =$ řetězec v $P$, k němu odpovídající vahová matice $\mathbf{W}_s$
generuje minimální průměrnou čtvercovou chybu;
jestliže $S(\mathbf{W}_s^*) > S(\mathbf{W}_s)$, $s^* = s$;
$geno = geno - 1$;}
výstup $s^*$;}

GA se dají lehce interpretovat na řešení nejrůznějších úkolů, vychází to z jejich obecnosti. Jejich hlavní nevýhodou je nutnost nastavení mnoha parametrů, jako je reprezentace dat, definice mutace a křížení. V současné době se GA stále více používají, i když nejsou příliš propracované. Každý si je musí upravit pro svá data, je to spíše prováděno formou pokus - omyl.

O podrobnějších informacích o GA se mužeme dočíst např. v literatuře [9], [12], [15] a [17].

# 6. Aplikace

## 6.1. Metody shlukové analýzy implementované v Matlabu

V následující kapitole se budeme zabývat aplikacemi jednotlivých metod na praktickém příkladě v programu MATLAB. Podíváme se na řešení hierarchických metod a metody rozkladu (metoda $k$ průměrů) pomocí zabudovaných příkazů. Nyní se blíže seznámíme s příkladem.

Pro praktický příklad jsme zvolili hlasování poslanců. Data jsme získali z internetových stránek Poslanecké sněmovny [14]. Jedná se o 11 vybraných hlasování všech poslanců za období červen - říjen 2008. Vybrané období je kratší kvůli časté změně v Poslanecké sněmovně, jelikož někteří poslanci odstupují ze své funkce v průběru zvoleného období a jsou nahrazeni jinými z příslušné politické strany.

Vybraná hlasování se týkají následujících novel: o registrovaném partnerství, o zaměstnanosti, o vzdělání, o veřejných zakázkách, o cestovních dokladech, o elektronických komunikacích, na ochranu zvířat proti týrání - EU, o zajištění jakosti a bezpečnosti lidského tkaní, o námořní plavbě, o správě daní a poplatků, o omezení plateb v hotovosti.

Přehled o hlasování všech poslanců v jednotlivých novelách zákonů je uveden v souboru POSLANCI200.xls, zde uvedeme jen ukázku:

| jméno | p.str. | poř. | h1 | h2 | h3 | h4 | h5 | h6 | h7 | h8 | h9 | h10 | h11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| W.Bartoš | ODS | 1 | A | A | A | Z | Z | A | Z | A | A | N | A |
| J.Bauer | ODS | 2 | A | A | A | A | A | A | A | O | O | O | O |
| ⋮ | ⋮ | ⋮ | | | | | | | | | | | |
| J.Paroubek | ČSSD | 121 | A | N | N | A | Z | O | A | O | A | A | Z |
| B.Petr | ČSSD | 122 | A | N | N | A | O | O | Z | O | O | A | Z |
| ⋮ | ⋮ | ⋮ | | | | | | | | | | | |
| M.Pohanka | NZ | 200 | O | O | Z | A | Z | O | Z | O | O | N | A |

Pro zjednodušení výpočtu jsme si převedli kvalitativní hodnoty jednotlivých hlasování na číselné jednotky a to následujícím způsobem:

[A] - ano ..... 1,

[N] - ne ..... -1,

[Z] - zdržel se ..... 0,

[O] - nepřítomen ..... 0,

[M] - omluven ..... 0 .

Budeme zkoumat, kteří poslanci a také které politické strany mají k sobě nejblíže, nebo naopak se nejvíce liší ve svých názorech.

Budoucí čtenáři této práce by nemuseli být seznámeni se situací ve sněmovně ve zkoumaném období červen – říjen 2008. Pravicová koalice byla tvořena ODS, KDU–ČSL, SZ, NZ. K opozičním stranám patřila ČSSD, KSČM.

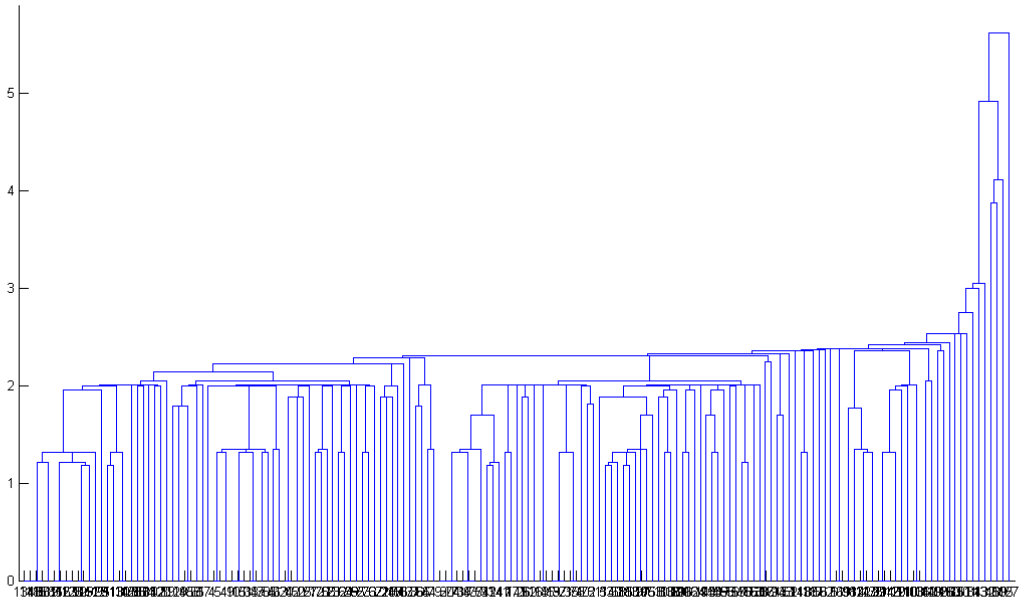
### 6.1.1. Hierarchické metody

Nejdříve se podíváme na řešení daného příkadu pomocí zabudovaných příkazů programu MATLAB a na dendrogram. Tento program nám poskytuje rychlé a snadné řešení daného úkolu. Pro první ukázku jsme zvolili hierarchickou metodu nejbližšího souseda (28) s výpočtem euklidovské vzdálenosti (14). Využili jsme základní příkazy:

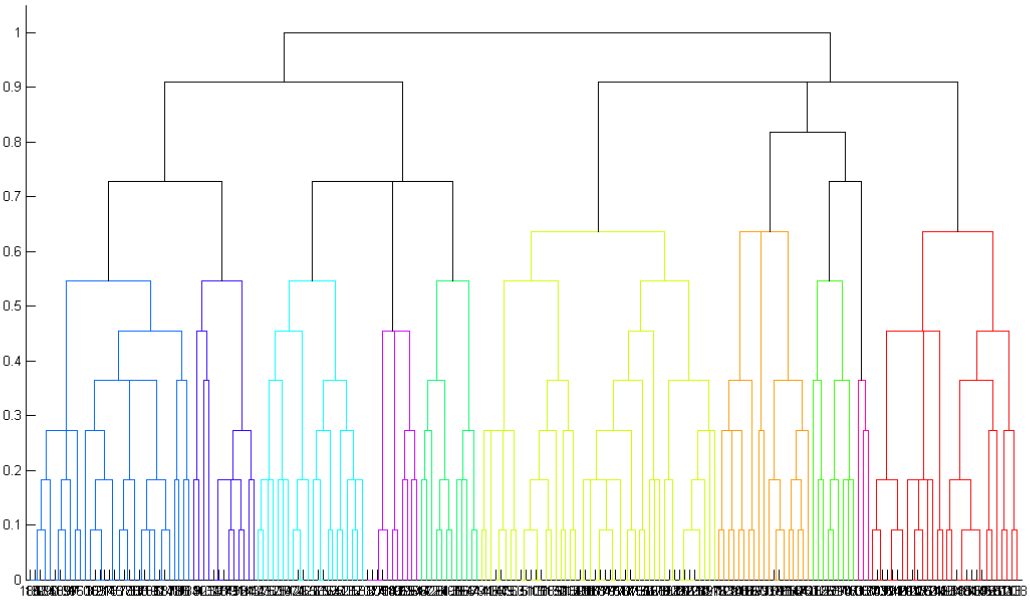
```
⋮
Y = pdist(X,'seuclidean')
squareform(Y);
Z=linkage(Y,'single');
[H,T]=dendrogram(Z,200,'colorthreshold','defaul');
```

kde $X$ je vstupní matice hlasování, *dpist* je funkce pro výpočet vzdáleností, *squareform* vrací symetrickou matici vzdáleností typu $n \times n$ (v našem případě $200 \times 200$) a funkce *linkage* zajišťuje shlukování.

Na obrázku 5 si můžeme všimnout, že se nám netvoří menší podshluky, ale jednotlivé objekty se postupně shlukují do jednoho shluku. Dále si ukážeme, jak to dopadne, pokud použijeme hierarchickou metodu nejvzdálenějšího souseda (29) s využitím Hammingovy vzdálenosti (21). Na šestém obrázku můžeme vidět vytvoření menších podshluků, které se následně shlukly do jednoho. Podshluky jsou pro přehlednost barevně rozlišeny.

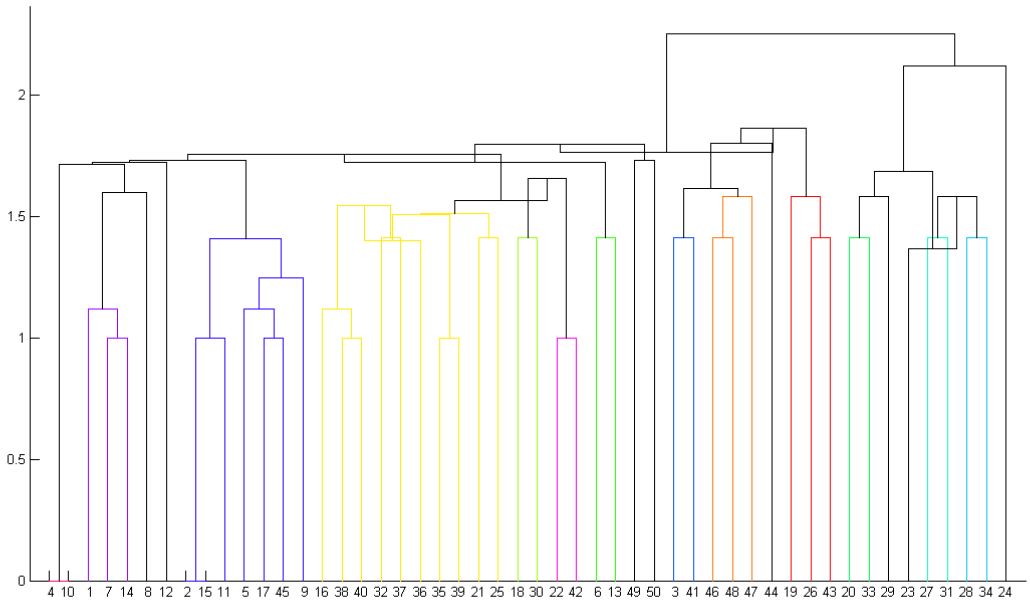



Obrázek 5: Metoda nejbližšího souseda




Obrázek 6: Metoda nejvzdálenějšího souseda

Jednotlivé dendrogramy jsou pro velký počet objektů nepřehledné. Proto si pro lepší přehled ukážeme dendrogram pro vybraných 50 poslanců a jejich 11 hlasování. Jsou vybráni zástupci ze všech politických stran. Zde uvažujeme hierarchickou centroidní metodu (33), u které se využívá pouze výpočet euklidovské vzdálenosti (14).




Obrázek 7: Centroidní metoda pro vybraných 50 poslanců

Na tomto obrázku vidíme postupné vytváření podshluků, dokud se nám všechny objekty neshlukly do jednoho shluku. Nejdříve se nám vytvořily podshluky:

- č. 4 M. Hrbala (ODS) + č. 10 P. Nečas (ODS),
- č. 1 W. Bartoš (ODS) + č. 7 J. Ježek (ODS) + č 14 L. Talmanová (ODS),
- č. 2 J. Bauer (ODS) + č. 5 P. Hrnčíř (ODS) + č. 9 I. Langer (ODS) + č. 11 M. Němcová (ODS) + č. 15 P. Tluchoř (ODS) + č.17 M. Topolánek (ODS) + č. 45 V. Parkanová (KDU-ČSL),
- č. 16 V. Tlustý (ODS) + č. 21 M. Hašek (ČSSD) + č. 25 J. Krátký (ČSSD) + č. 32 D. Rath (ČSSD) + č. 35 M. Bayerová (KSČM) + č. 36 Z.B. Rujbrová

(KSČM) + č. 37 M. Bičík (KSČM) + č. 38 S. Marková (KSČM) + č. 39 Z. Maršíček (KSČM) + č. 40 L. Mlčák (KSČM),

- č. 18 V. Aubrecht (ČSSD) + č. 30 P. Ploc (ČSSD),
- č. 6 R. Chytka (ODS) + č. 13 D. Riesiegel (ODS),
- č. 22 J. Chalupa (ČSSD) + č. 42 M. Kalousek (KDU-ČSL),
- č. 3 P. Bohatec (ODS) + č. 41 L. Hovorka (KDU-ČSL),
- č. 46 M. Bursík (SZ) + č. 47 K. Jacques (SZ) + č. 48 O. Liška (SZ),
- č. 19 J. Babor (ČSSD) + č. 26 S. Křeček (ČSSD) + č. 43 J. Kalas (KDU-ČSL),
- č. 20 V. Bohdalová (ČSSD) + č. 33 L. Šincl (ČSSD),
- č. 27 J. Paroubek (ČSSD) + č. 28 B. Petr (ČSSD) + č. 31 P. Rafaj (ČSSD) + č. 34 Z. Škromach (ČSSD).

Poslanci: č. 8 J. Krupka (ODS), č. 12 A. Rádl (ODS), č. 23 V. Jandák (ČSSD), č. 24 J. Krákora (ČSSD), č. 29 J. Petrů (ČSSD), č. 44 T. Kvapil (KDU-ČSL), č. 49 M. Melčák (NZ), č. 50 M. Pohanka (NZ) nevytvořili podshluky, ale přiřadili se později k výše vytvořeným podshlukům.

### 6.1.2. Metoda $k$ průměrů

Stejně jako pro hierarchické metody má MATLAB zabudované příkazy i pro metody rozkladu. Hlavním rozdílem od hierarchických metod je, že u metod rozkladu musíme předem stanovit počet shluků, do kterých mají být jednotlivé objekty shluknuty.

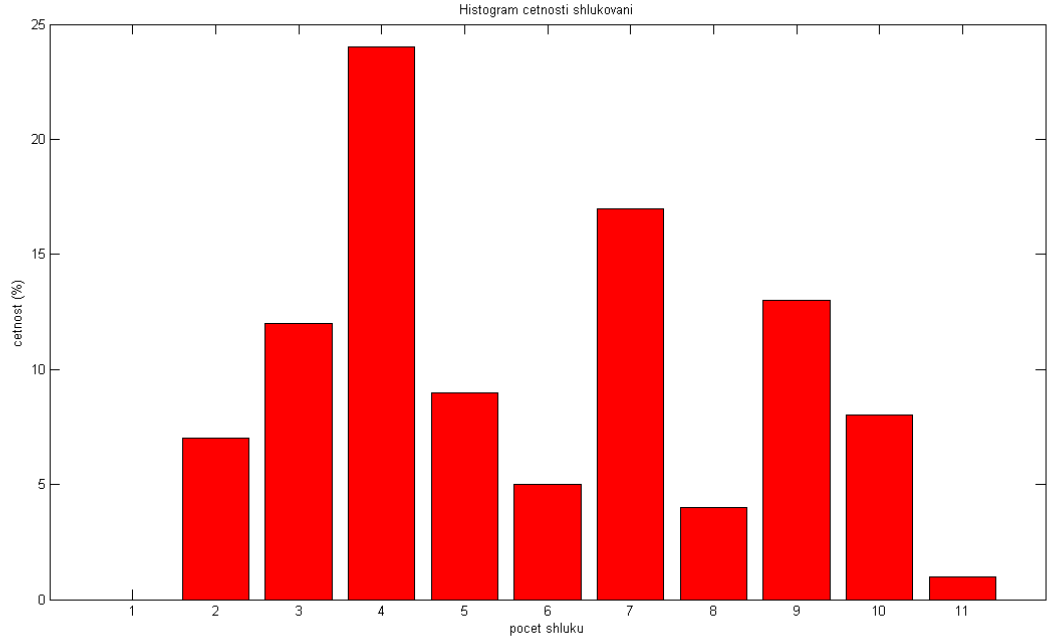
Princip metody $k$ průměrů si ukážeme na příkladu z poslanecké sněmovny, kde opět uvažujeme 11 hlasování 200 poslanců. Využili jsme příkazy:

$$\vdots$$

```
X=transformaceHLASOVANI(HLASOVANI);
Xdim=X;
idx7=kmeans(Xdim,7,'dist','city', 'display','iter');
[silh7,h]=silhouette(Xdim,idx7);
sedm=mean(silh7)
idx6=kmeans(Xdim,6,'dist','city', 'display','iter');
[silh6,h]=silhouette(Xdim,idx6);
mean(silh6)
idx5=kmeans(Xdim,5,'dist','city', 'display','iter');
[silh5,h]=silhouette(Xdim,idx5);
mean(silh5)
idx4=kmeans(Xdim,4,'dist','city', 'display','iter');
[silh4,h]=silhouette(Xdim,idx4);
mean(silh4)
```

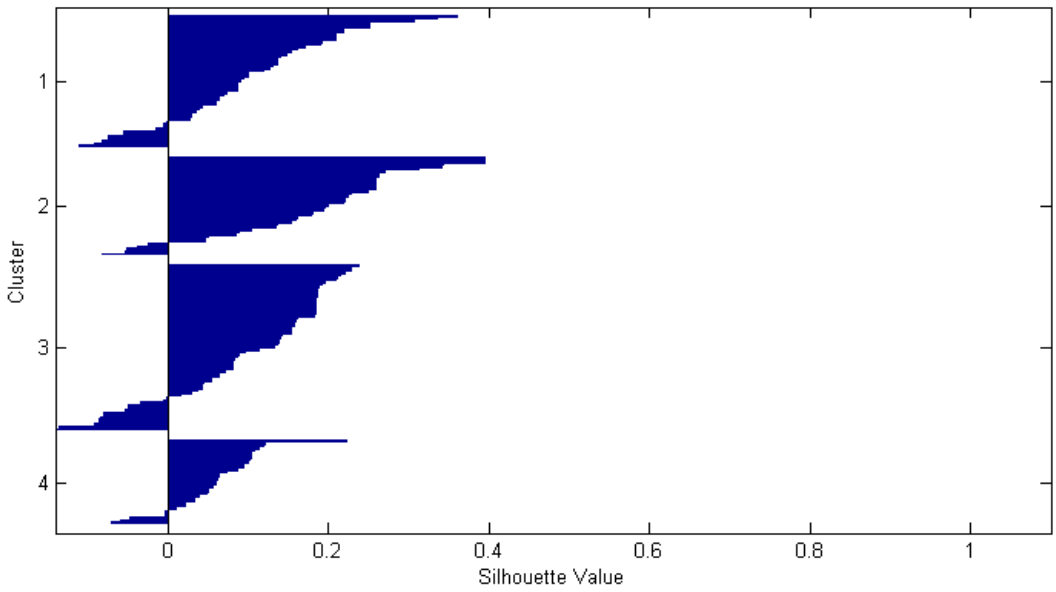
Funkce *kmean* nám rozdělí data do $k$ shluků a vrací index shluku, do kterého byl objekt zařazen, využívá se přitom euklidovská vzdálenost (14) objektu od centroidu (středu) shluku. Funkce *silhouette* vykresluje obrysy shluků, které jsou vytvořeny a funkce *mean* nám slouží k porovnání více řešení pomocí průměrné hodnoty daných obrysů shluků.

Do příkladu jsme zadali, aby se objekty zařadily do 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 a 11 shluků. Je nutné si uvědomit, že funkce kmean si na začátku volí zástupce shluků náhodně, proto se výsledky mohou lišit, pokud výpočet provádíme opakovaně. Výpočet byl několikrát opakován. Ze získaných výpočtů jsme zjistili, že algoritmus vytvořil nejčastěji 4 shluky, následně 7 a 9 shluků, což můžeme vidět na následujícím histogramu četnosti přiřazení objektů do shluků.




Obrázek 8: Histogram četnosti shluků

Na následujícím obrázku můžeme vidět přiřazení objektů do 4 shluků s hodnotou $mean = 0.1079$ .




Obrázek 9: Metoda 4 - průměrů

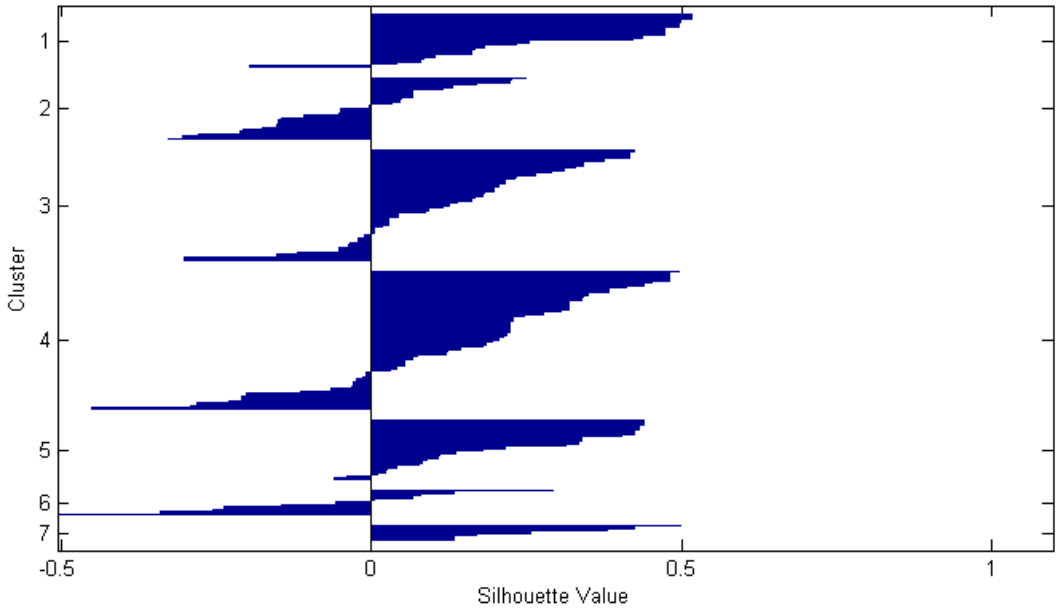
V příloze *idxPoslmean.xls* máme uveden vektor idx4, který jsme dostaly na výstupu. Tento vektro udává, do kterých shluků byly jednotlivé objekty přiřazeny. V našem případě se např. do čtvrtého shluky přiřadili poslanci:

T. Dub (ODS), E. (ODS), M. Hrbata (ODS), R. Chytka (ODS), L. Ježek (ODS), D. Kafka (ODS), J. Klas (ODS), F. Laudát (ODS), L. Libý (ODS), P. Nečas (ODS), O. Suk (ODS), V. Šoltys (ODS), V. Bohdalová (ČSSD), V. Jandák (ČSSD), Z. Jičínský (ČSSD), V. Klučka (ČSSD), Z. Kotouš (ČSSD), K. Kratochvíle (ČSSD), R. Kufa (ČSSD), A. Michalík (ČSSD), P. Rafaj (ČSSD), D. Rath (ODS), H. Šedivá (ČSSD), L. Šincl (ČSSD), M. Váňa (ČSSD), Z. Bebarová - Rujbrová (KSČM), M. Bičík (KSČM), P. Braný (KSČM), M. Grebeníček (KSČM), P. Hojda (KSČM), P. Rabas (SZ).

V tomto shluku máme nejvíce zástupců z ODS a ČSSD, na druhé straně se zde nevyskytují zástupci z KDU-ČSL a Nezávislí.

Pokud mají dva objekty stejnou hodnotu pro přiřazení do shluku, tak se jednomu z nich přiřadí záporná hodnota, aby se následně rozlišily. proto shluky obsahují objekty se zápornou hodnotou, což lze vidět na obrázku.

Na dalším obrázku můžeme pro porovnání vidět nejoptimálnější přiřazení objektů do 7 shluků s hodnotou $mean = 0.1378$.




Obrázek 10: Metoda 7 - průměrů

Vektor idx7 (rozměru $200 \times 1$) uvedený v příloze *idxPoslmean.xls* nám udává, do kterých shluků byly objekty zařazeny. Např. sedmý shluk obsahuje poslance: Walter Bartoš (ODS), Eva Dundáčková (ODS), Jozef Kochan (ODS), Daniel Reisiegel (ODS), Martin Říman (ODS), Pavel Svoboda (ODS), Boris Šťastný (ODS).

Do tohoto shluku se přiřadili pouze poslanci z ODS.

## 6.2. Metody shlukové analýzy implementované v SASu

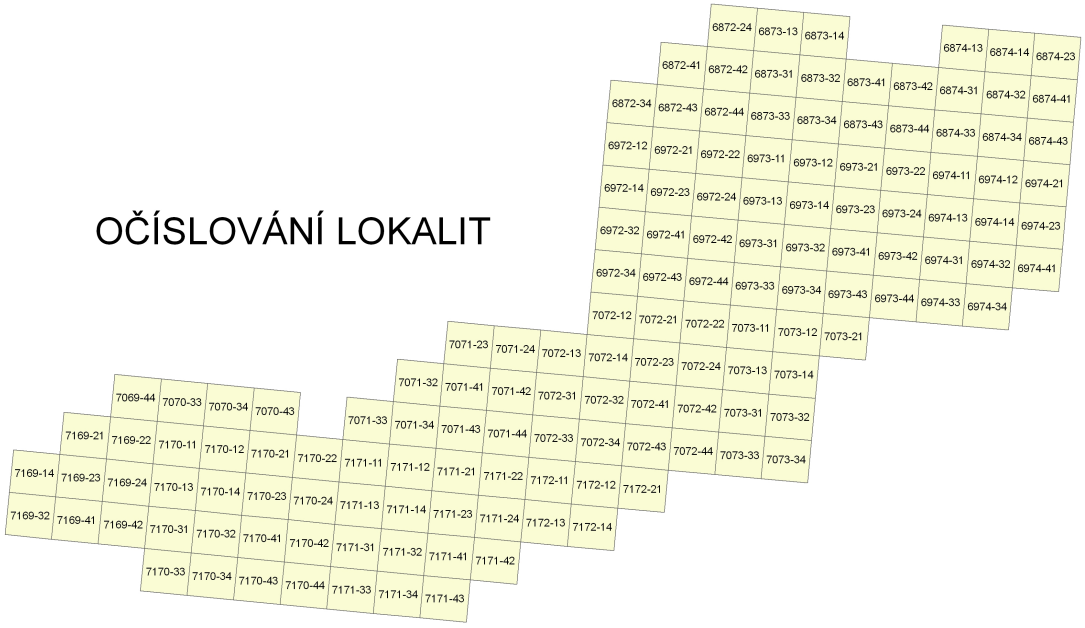
V předchozí kapitole jsme si ukázali shlukování pomocí programu MATLAB. V dnešní době však existuje mnoho statisticky zaměřených programů, jedním z nich je statistický program SAS. Tento software poskytuje integrovaný proces pro analýzu jakýchkoliv dat a nástroj pro prognózu a plánování změn vedoucích k úspěšnému řešení. V této kapitole si ukážeme řešení shlukové analýzy pomocí tohoto programu. SAS poskytuje řešení problému pomocí hierarchické metody nejbližšího souseda a to průměrným shlukováním, hierarchickou centroidní metodou a dále metodou rozkladu $k$ průměrů.

Shlukovou analýzu můžeme aplikovat na různé příklady, např. z oblasti ekonomie, bankovnictví, biologie, medicíny, atd., proto si ukážeme shlukování pomocí SASu na příkladu z biologického prostředí.

Byla nám poskytnuta data z katedry geoinformatiky Přf UP Olomouc [13] o výskytu 123 druhů motýlů na území Bílých Karpat. Toto území je pokryto 113 mapovými listy přibližně stejně velkých lokalit (kvadrátů). Každá lokalita je očíslována podle principů pro číslování map v kladu listů. U každé lokality máme uvedeno: jaký typ půdy se zde nachází nebo zda se zde vyskytují lesy, skály, louky či pole a další údaje. Tyto faktory ovlivňují výskyt rostlin a živočichů. Např. o lokalitě v mapovém listu č. 7170-33 víme:

| faktor | hodnota |
|---|---|
| výměra kvadrátu v ČR / (BK) (ha) | 138,18 |
| výměra celého kvadrátu (ha) | 850,87 |
| střední nadmořská výška v q (m n.m.) | 465 |
| minimální nadmořská výška v q (m n.m.) | 357 |
| maximální nadmořská výška v q (m n.m.) | 569 |
| rozdíl (m) | 212 |
| Délka vodní toků v q (m) | 2289 |
| Délka asfaltových komunikací v q (m) | 0 |
| kyselý flyš absolutní výměra v q (ha) | 0,75 |
| zásaditý flyš absolutní výměra v q (ha) | 137,43 |
| pěnovce a sutě absolutní výměra v q (ha) | 0 |
| glejové půdy absolutní výměra v q (ha) | 2,64 |
| ⋮ | ⋮ |

Podrobné informace o všech lokalitách můžeme najít v souboru *faktoryorez.csv*.

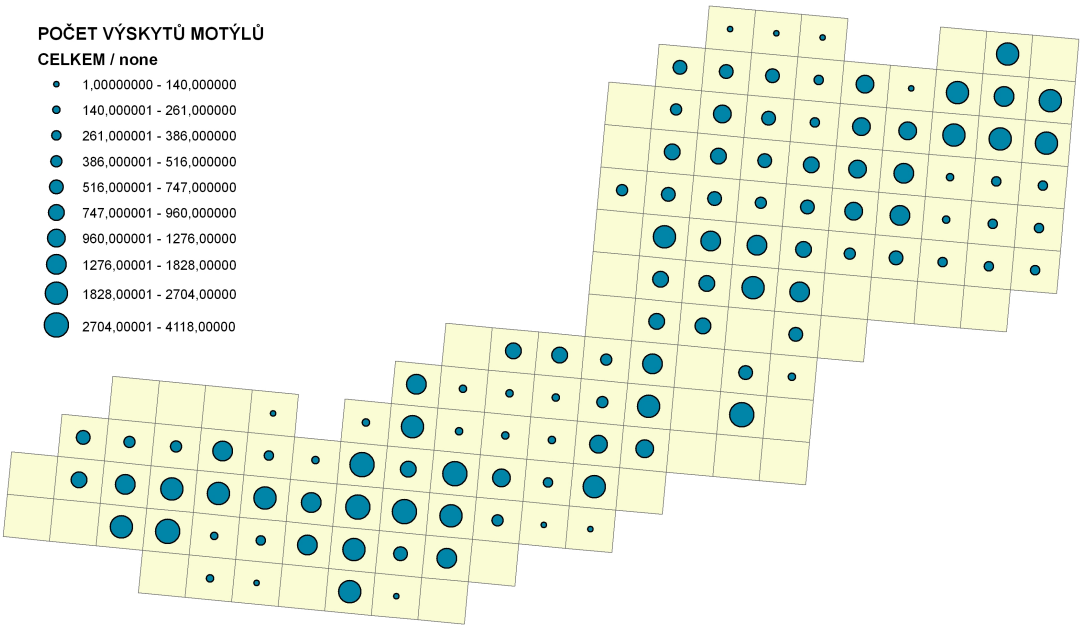



Obrázek 11: Očíslování lokalit

Máme k dispozici vstupní matici rozměru 123×113, kde máme údaje o výskytu 123 druzích motýlů ve 113 lokalitách. Uvedená data nám udávají výskyt daného druhu motýlů (objekty) v jednotlivých lokalitách (proměnné), obsahem není skutečný počet motýlů, ale jen počet pozorování (např. pokud tam bylo pozorováno v jeden den 10 motýlů, bere se to jako 1 výskyt, druhý den 30, opět bereme jako 1 výskyt, celkem tedy 2). Všechny potřebné informace o výskytu motýlů jsou uvedeny v souboru *nasimotyli.xls*, např.:

| název | N6872_41 | N6872_42 | N6872_43 | ... | N7172_13 | N7172_14 |
|---|---|---|---|---|---|---|
| AGLURT | 3 | 4 | 2 | ... | 1 | 0 |
| ANTCAR | 43 | 72 | 31 | ... | 12 | 0 |
| APAILI | 0 | 0 | 0 | ... | 0 | 0 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| COEGLZ | 20 | 22 | 12 | ... | 0 | 0 |
| COEPAM | 43 | 72 | 43 | ... | 51 | 7 |
| COLALF | 0 | 0 | 0 | ... | 0 | 0 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| VANATA | 15 | 20 | 5 | ... | 0 | 1 |
| VANCAR | 14 | 18 | 1 | ... | 1 | 0 |
| ZERPOL | 0 | 0 | 0 | ... | 0 | 0 |

Na obrázku můžeme vidět zkoumané území rozdělené na příslušné lokality s danými výskyty motýlů.




Obrázek 12: Výskyt motýlů v daných lokalitách

Jména motýlů jsou uvedena jako zkratky latinských názvů. Např.:

Babočka Admirál = Vanessa Atalanta - VANATA

Bělásek řepkový = Pieris napi - PIENAP

Okáč zední = Lasiommata megera - LASMEG

Soumračník jahodníkový = Pyrgus malvae - PYRMAL

Obrázek 13: Babočka Admirál

Obrázek 14: Bělásek řepkový

Obrázek 15: Okáč zední

Obrázek 16: Soumračník jahodníkový

Budeme zkoumat, které druhy motýlů jsou si nejvíce podobné z hlediska výskytu v jednotlivých lokalitách. Odtud se můžeme např. dozvědět, jaké druhy motýlů žijí ve stejných lokalitách, což je ovlivněno faktory jednotlivých lokalit.

### 6.2.1. Metoda nejbližšího souseda

Program SAS využívá pro hierarchické shlukování metodu nejbližšího souseda s průměrným propojením shluků, tj. při shlukování objektů bereme v úvahu aritmetický průměr hodnot objektů v daném shluku.

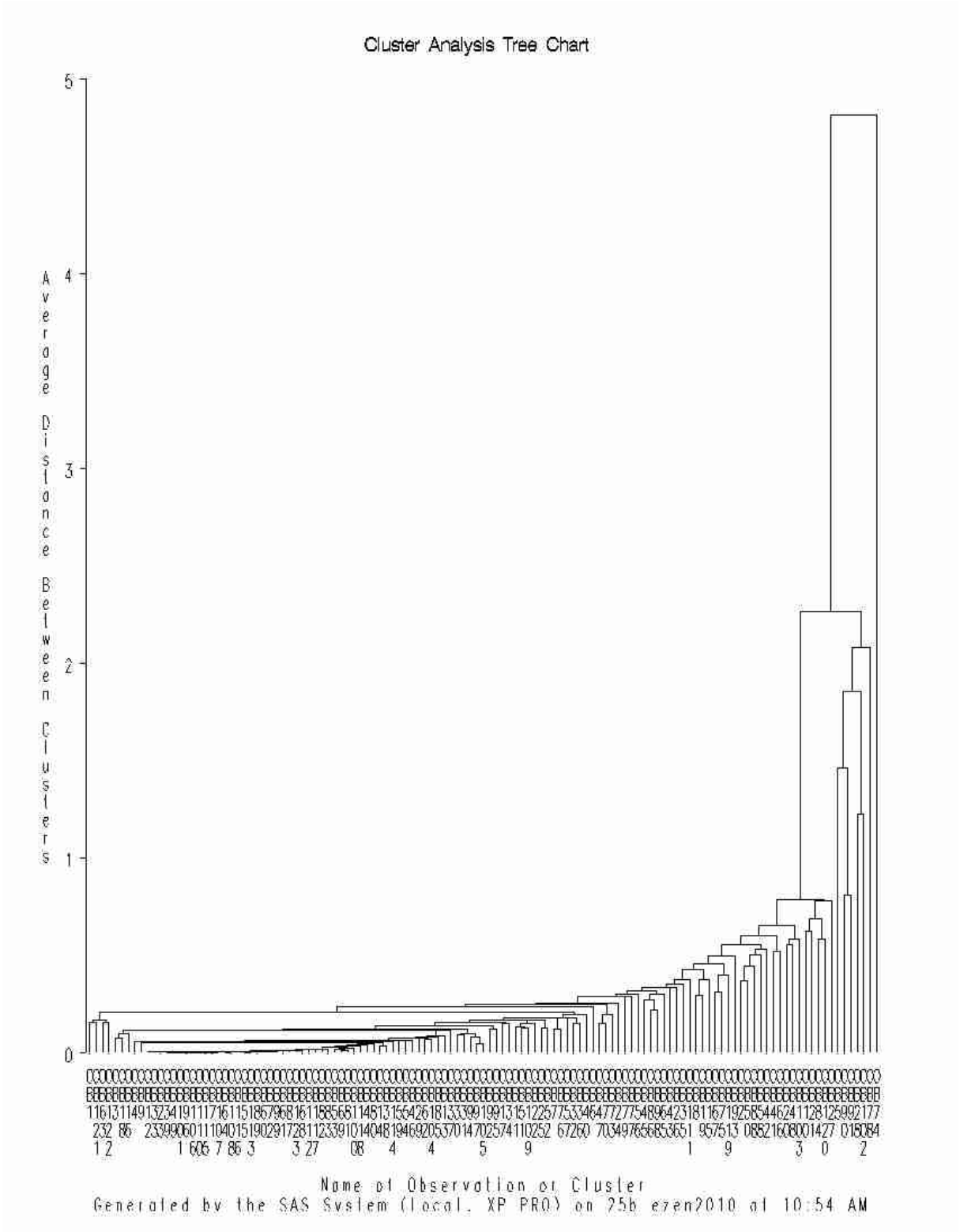
Po zadání příkladu do SASu dostáváme základní údaje o shlukování, tj. které objekty se nám k sobě postupně shlukly a také v jaké iteraci. Výsledky výpočtů můžeme vidět v přiloženém souboru *hierMetodaSAS.pdf* na straně $9 - 12$ a *hierSASmot.xls*.

Cluster History

| NCL | Clusters Joined | | FREQ | Norm RMS Dist | Tie |
|---|---|---|---|---|---|
| 122 | OB49 | OB101 | 2 | 0 | |
| 121 | OB39 | CL122 | 3 | 0.0013 | T |
| 120 | CL121 | OB96 | 4 | 0.0013 | T |
| 119 | CL120 | OB106 | 5 | 0.0013 | T |
| 118 | CL119 | OB110 | 6 | 0.0013 | |
| 117 | OB64 | OB108 | 2 | 0.0021 | |
| 116 | CL118 | OB115 | 7 | 0.0028 | T |

Obrázek 17: Historie shlukování

Na základě výpočtu shlukování dostáváme dendrogram motýlů:




Obrázek 18: Dendrogram - metoda nejbližšího souseda

Dendrogram vytvořený v programu SAS je přehlednější než vytvořený v programu MATLAB.

SAS nám také při výpočtu udává základní statistické údaje o jednotlivých proměnných a to střední hodnotu, směrodatnou odchylku, šikmost a špičatost. Jednotlivé výpočty jsou uvedeny v souboru *hierMetodaSAS.pdf* na straně $1 - 4$.

### 6.2.2. Centroidní metoda

Další metodou, kterou nám poskytuje SAS, je hierarchická centroidní metoda. U centroidní metody se vychází z těžišť (centroidů) vytvořených shluků během shlukování. U této metody je velice důležité, že pro výpočet vzdáleností se používá pouze kvadrát euklidovské vzdálenosti.

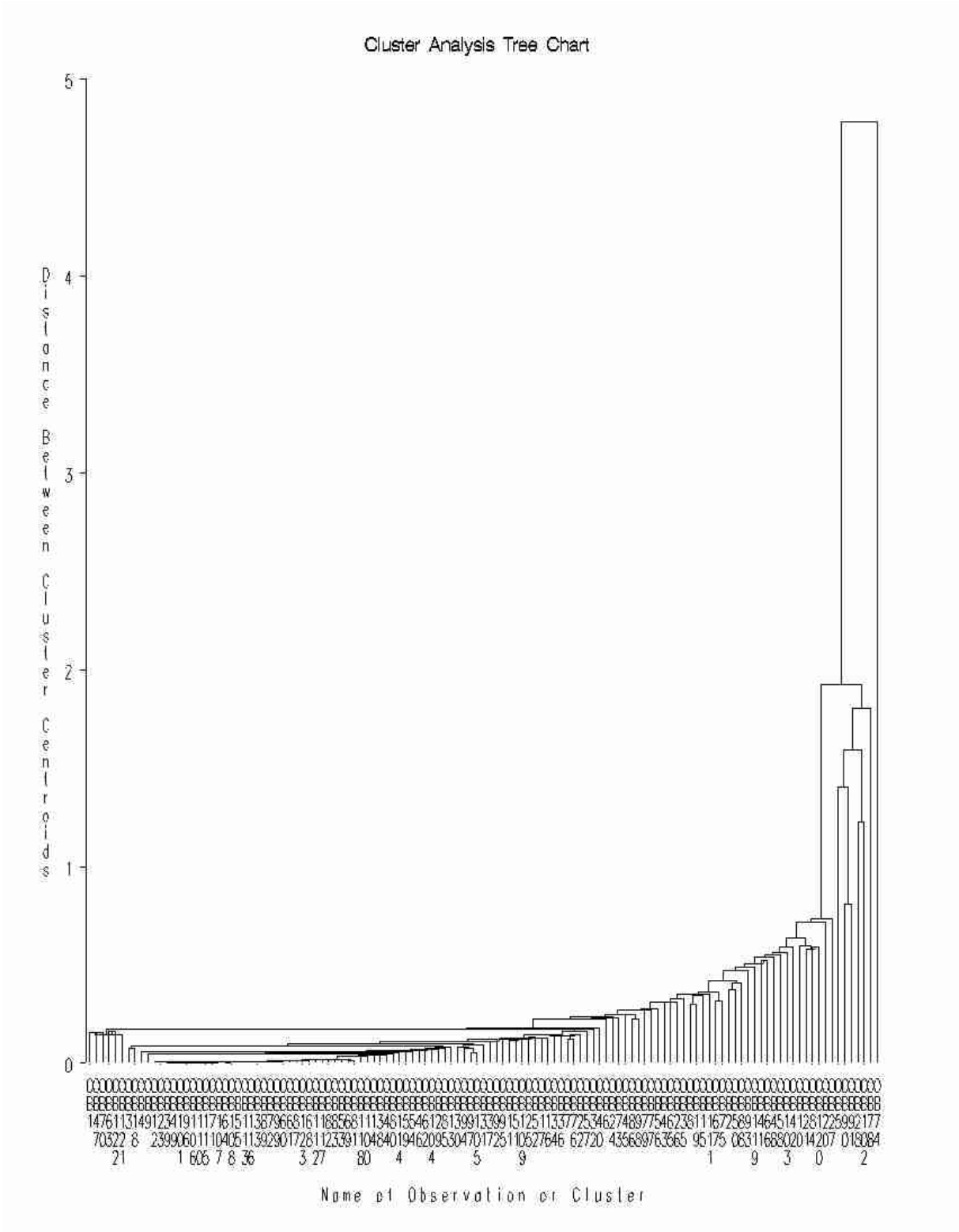
Výstupem SASu u centroidní metody je opět výpis postupného shlukování, tj. které objekty se k sobě shlukly a při jaké iteraci. Vše je uvedeno v souboru *centroidMetSAS.pdf* na straně $9 - 12$ a *centroidMetSAS.xls*.

Cluster History

| NCL | Clusters Joined | | FREQ | Norm Cent Dist | Tie |
|---|---|---|---|---|---|
| 122 | OB49 | OB101 | 2 | 0 | |
| 121 | OB39 | CL122 | 3 | 0.0013 | T |
| 120 | CL121 | OB96 | 4 | 0.0012 | T |
| 119 | CL120 | OB106 | 5 | 0.0011 | T |
| 118 | CL119 | OB110 | 6 | 0.0011 | |
| 117 | OB64 | OB108 | 2 | 0.0021 | |
| 116 | CL118 | OB115 | 7 | 0.0027 | |
| 115 | OB71 | OB107 | 2 | 0.0045 | |
| 114 | CL116 | CL115 | 9 | 0.0047 | |
| 113 | CL114 | CL117 | 11 | 0.0054 | |

Obrázek 19: Historie shlukování - centroidní metoda

Proces shlukování je znázorněn na následujícím dendrogramu.




Obrázek 20: Dendrogram - centroidní metoda

Stejně jako u metody nejbližšího souseda nám SAS u centroidní metody dává na výstupu statistické údaje o jednotlivých proměnných, které jsou uvedeny v souboru *centroidMetSAS.pdf* na straně $1 - 4$.

### 6.2.3. Metoda $k$ průměrů

Vedle hierarchických metod SAS poskytuje pro řešení klasifikace dat i metodu rozkladu $k$ průměrů. Zde je důležité na začátku stanovit počet shluků, do kterých se mají dané objekty přiřadit, dále konvergenční kritérium (obvykle volí 0.2) a maximální počet iterací. SAS nám na výstupu dává základní statistické údaje jak o proměnných, tak o vytvořených shlucích.

My jsme aplikovali metodu $k$ průměrů na příklad z entomologického výzkumu. Na začátku jsme si zvolili, že se objekty mají rozdělit do 10 shluků. To je dáno metodou pokus-omyl. Konvergenční kritérium jsme zvolili rovno 0.2. Na výstupu jsme dostali:

Iteration History

| Iteration | Criterion | Relative Change in Cluster Seeds | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 60.5750 | 0.7095 | 0 | 0 | 0.4749 | 0 | 0 | 0.5371 | 0.5769 | 0 | 0.4789 |
| 2 | 21.7753 | 0.00909 | 0 | 0 | 0 | 0 | 0 | 0.0475 | 0.0675 | 0 | 0 |
| 3 | 21.6796 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Obrázek 21: Historie shlukování - metoda $k$ průměrů

Zde máme uvedeny hodnoty relativní změny pro středy shluků, které se nám ustálily po třech iteracích. V následující tabulce máme uvedeno, kolik objektů bylo přiřazeno do příslušných shluků, směrodatnou odchylku daného shluku, maximální vzdálenost objektu od středu shluku, nejbližší shluk a vzdálenost mezi centroidy těchto shluků. Dále je v souboru *10meanSAS.pdf* na straně 13 a 14 uvedena hodnota radiusu překročení, jehož hodnoty jsou rovny 0, proto je v ná-

sledující tabulce vynechán.

| Cluster | freque. | Std deviat. | Max. dist. | Near. clust. | Dis. betw. cl. centroid |
|---|---|---|---|---|---|
| 1 | 95 | 15.5842 | 642.3 | 7 | 373.6 |
| 2 | 1 | 0 | 0 | 9 | 2957.1 |
| 3 | 1 | 0 | 0 | 6 | 1291.8 |
| 4 | 2 | 56.8329 | 427.2 | 5 | 1482.5 |
| 5 | 1 | 0 | 0 | 4 | 1482.5 |
| 6 | 1 | 0 | 0 | 3 | 1291.8 |
| 7 | 8 | 38.9424 | 499.6 | 1 | 373.6 |
| 8 | 10 | 44.4122 | 578.9 | 7 | 386.7 |
| 9 | 1 | 0 | 0 | 5 | 1946.7 |
| 10 | 3 | 41.130 | 430.8 | 8 | 430.4 |

Z tabulky je zřejmé, že nejvíce objektů bylo přiřazeno do prvního shluku. Hlavní nevýhodou metody $k$ průměrů u programu SAS je, že zde nemáme přehled, které objekty byly přiřazeny k daným shlukům. Je nutné si projít informace o střední hodnotě shluků, v našem případě jsou uvedeme v souboru *10meanSAS.pdf* na straně $19-31$. Jelikož máme k dispozici 113 pozorování u každého objektů, je to z časového hlediska velice náročné. Jako další nevýhodou je, že SAS neposkytuje grafické znázornění řešení.

Veškeré informace o výsledku řešení příkladu výskytu motýlů jsou uvedeny ve výše zmíněném souboru, kde se můžeme dočíst i základní statistické údaje o jednotlivých proměnných. V příloze je uvedeno řešení shlukování pro 15 a 20 shluků.

## 6.3. Naprogramované aplikace

### 6.3.1. Metoda nejbližšího souseda

Program MATLAB nám poskytuje opravdu rychlé řešení, ale už zde nevidíme postupný výpočet a jednotlivé iterace shlukování. Z dendrogramu můžeme vyčíst, které objekty se shlukly. Neznáme už ale postup shlukování a hodnoty z matice vzdáleností, podle kterých došlo ke shlukování. Proto jsme si vytvořili vlastní shlukovací algoritmus v programu MATLAB, který popíšeme na příkladu z poslanecké sněmovny pro 200 poslanců.

Uvažujeme výše uvedená data z 11 hlasování 200 poslanců a metodu nejbližšího souseda. Máme tedy vstupní matici $200 \times 11$. Pro výpočet vzdálenosti použijeme Hammingovu vzdálenost (21), odtud dostáváme symetrickou matici vzdáleností.

V následující části popíšeme algoritmus pouze pro 5 vybraných poslanců, abychom mohli samotnou metodu lépe vysvětlit. Mezi vybranými poslanci je M. Topolánek (ODS), J. Paroubek (ČSSD), L. Mlčák (KSČM), M. Kalousek (KDU-ČSL) a M. Bursík (SZ).

Uvažujeme tedy vstupní matici $X_1$ rozměru $(5 \times 11)$:

$$\mathbf{X}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Na základě Hammingovy vzdálenosti vypočítáme matici $D$:

$$\mathbf{D} = \begin{pmatrix} 0 & 2.645 & 2.828 & 1.414 & 1.732 \\ 2.645 & 0 & 3 & 2.236 & 2.828 \\ 2.828 & 3 & 0 & 2.449 & 3.317 \\ 1.414 & 2.236 & 2.449 & 0 & 2.236 \\ 1.732 & 2.828 & 3.317 & 2.236 & 0 \end{pmatrix}$$

Dle metody nejbližšího souseda vybereme nejmenší vzdálenost, v naše případě je to hodnota 1.414 pro první a čtvrtý objekt. Vzniká nám shluk $C_1 = \{x_1, x_4\}$,

který je tvořen poslanci M. Topolánkem (ODS) a M. Kalouskem (KDU-ČSL). Charakteristikou tohoto shluku je aritmetický průměr hodnot objektů 1 a 4, tedy $(1, 0.5, 0.5, 1, 0, 0, 0, 0, 0, 0, 0)$.

Tento shluk umístíme na pozici prvního shluku ve vstupní matici $X_2$ rozměru $(4 \times 11)$.

Po tomto kroku znovu vypočítáme matici vzdáleností, nyní dostáváme matici o rozměru $(4 \times 4)$:

$$\mathbf{D}_1 = \begin{pmatrix} 0 & 2.449 & 2.646 & 2 \\ 2.449 & 0 & 3 & 2.828 \\ 2.646 & 3 & 0 & 3.317 \\ 2 & 2.828 & 3.317 & 0 \end{pmatrix}$$

V matice $D_1$ najdeme nejmenší vzdálenost, což je hodnota 2 pro shluk $C_1$ a čtvrtý objekt (v původní matici by to byl objekt 5). Dostáváme shluk $C_2 = \{C_1, x_4\}$, je tvořen poslanci M. Topolánkem (ODS), M. Kalouskem (KDU-ČSL) a M. Bursíkem (SZ). Shluk $C_2$ opět zařadíme na první pozici ve vstupní matici. Jeho charakteristikou je aritmetický průměr hodnot shluku $C_1$ a 5. objektu,tj. $(0.50, 0.75, 0.75, 1, 0, 0, -0.50, 0, 0, 0.50, 0)$.

Opět přepočítáme matici vzdáleností rozměru $3 \times 3$.

$$\mathbf{D}_2 = \begin{pmatrix} 0 & 2.646 & 3 \\ 2.646 & 0 & 3 \\ 3 & 3 & 0 \end{pmatrix}$$

Opět najdeme nejmenší vzdálenost, což je hodnota 2.646 pro objekt 2 a shluk $C_2$. Vzniká mám nový shluk $C_3 = \{C_2, x_2\}$, jehož hodnoty jsou $(0.75, -0.125, -0.125, 1, 0, 0, 0.25, 0, 0.50, 0.75, 0)$. K výše uvedeným poslancům se nám přidá J. Paroubek (ČSSD).

Pro 3. iteraci dostáváme

$$\mathbf{D}_3 = \begin{pmatrix} 0 & 2.646 \\ 2.646 & 0 \end{pmatrix}$$

Zde už naposledy sloučíme shluk $C_3$ s 2. objektem, jež je reprezentován poslancem L. Mlčákem (KSČM) (v původní matici by to byl čtvrtý objekt). Dostáváme

Shluk $C_4$ s charakteristikou
$(0.875, -0.0625, -0.0625, 0.5, 0.5, 0.5, 0.125, 0.5, 0.75, 0.375, 0.5)$, který obsahuje všech 5 objektů, které jsme sloučili na základě metody nejbližšího souseda.

S postupným shlukování je třeba si uvědomit, že pokud sloučíme objekty do jednoho shluku, tak se pro zbytek objektů změní jejich původní pořadí.

Stejným způsobem jsme provedli výpočet pro všech 200 poslanců. Díky vytvořenému algoritmu dostáváme postupné řešení hierarchické metody nejbližšího souseda. V tomto algoritmu si můžeme libovolně změnit vzorec pro výpočet vzdáleností a metodu nejbližšího souseda vyměnit např. za metodu nejvzdálenějšího souseda, kde uvažujeme maximální vzdálenost mezi objekty.

### 6.3.2. Genetický algoritmus $k$ průměrů

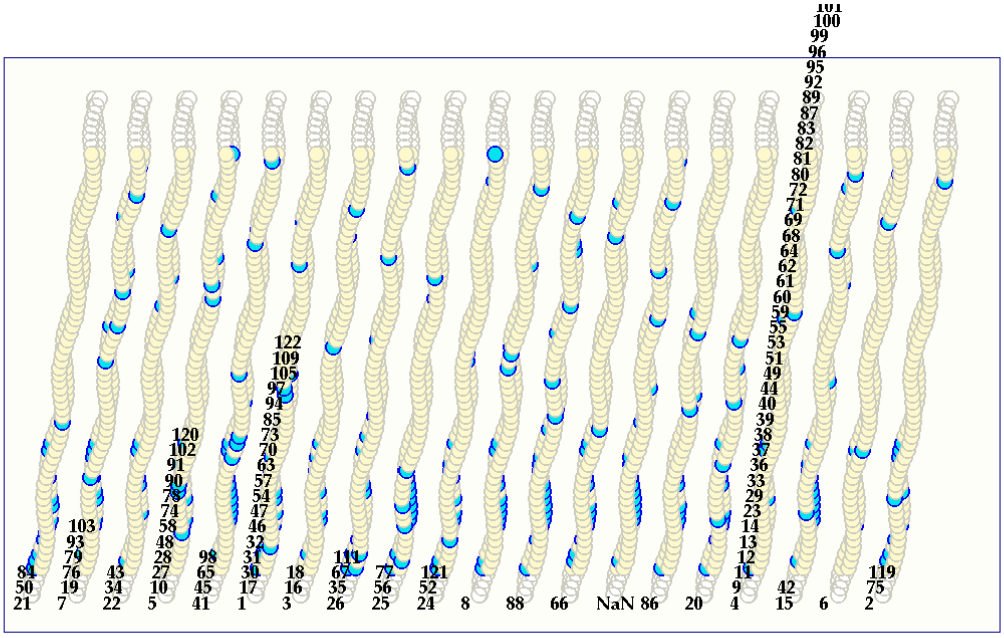
V dnešní době se pro klasifikaci dat stále více využívají alternativní přístupy inspirované přírodou. Pro shlukovou analýzu se využívá znalosti umělých neuronových sítí a genetických algoitmů. V následující části si ukážeme algoritmus GA $k$ průměrů, který aplikujeme na výše uvedeném příkladu, týkající se výskytu motýlů na určitém území.

Máme k dispozici data z výskytu 123 druhů motýlů ve 113 lokalitách, [13]. Hlavním principem algorimu je přiřazení objektů do předem stanoveného počtu shluků. V našem případě uvažujeme přiřazení motýlů do 20 shluků. V několika krocích si zde naznačíme průběh námi vytvořeného algoritmu $k$ průměrů:

1. Sestavíme počáteční matici vah $\mathbf{W}$ splňující podmínku (50). Matici vah jsme zvolili náhodně tak, aby splňovala dané podmínky.
2. Náhodně vybereme 20 objektů (z původních 123). Vypočítáme středy shluků dle (51).
3. Aplikujeme operátor selekce dle (54), (55).
4. Aplikujeme operátor křížení.
5. Aplikujeme operátor mutace na základě vypočítaných pravděpodobností dle (56).

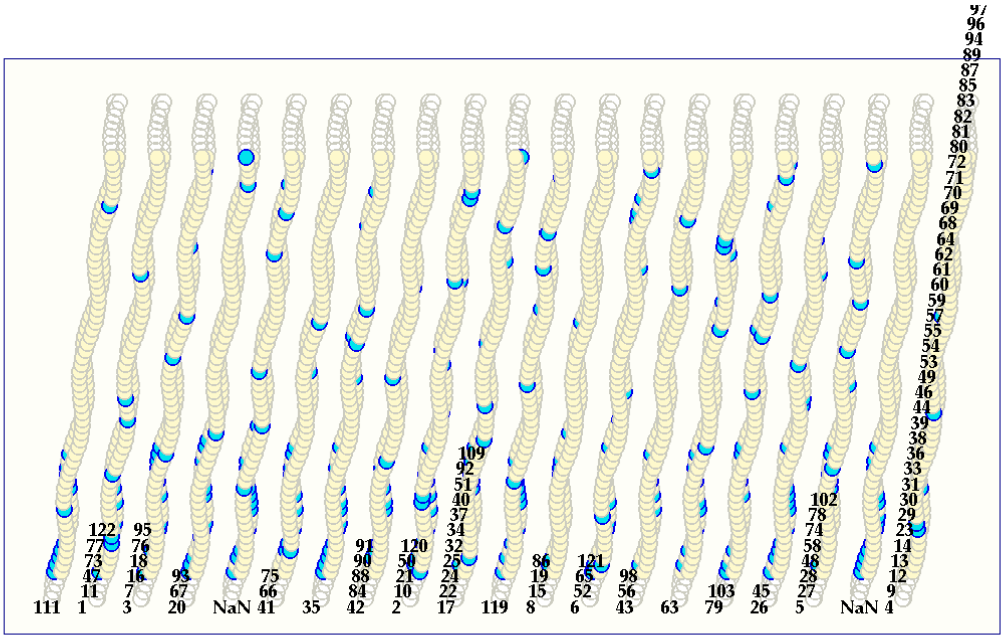
6. Přepočítáme matici vah **W** a středy shluků.
7. Přiřadíme objekty do shluků, tak aby vzdálenost objektu od příslušného shluku byla menší než vzdálenost objektu od ostatních středů.

Celý postup opakujeme pro příslušný počet iterací. V našem případě jsme celý proces opakovali pro 1000 iterací. Na následujícím obrázku můžeme vidět, jak dopadne shlukování pomocí vlastního algoritmu GA $k$ průměrů vytvořeného programem MATLAB. Máme 20 chromozomů, které nám reprezentují shluky. U každého chromozomu jsou uvedeny čísla objektů (motýlů), které byly přiřazeny do daného shluku na základě nejmenší vzdálenosti od středu shluku. Některé shluky mohou zůstat prázdné, tzn. že do shluku nebyl přiřazen žádný objekt. V takovém případě je shluku přiřazena hodnota NaN.




Obrázek 22: Shlukování motýlů (1)

Je nutné si uvědomit, že na začátku algoritmu vybíráme objekty náhodně, proto pokud náš algoritmus spustíme vícekrát, můžeme dostat rozdílné výsledky, což můžeme vidět na dalším obrázku.




Obrázek 23: Shlukování motýlů (2)

Opětovným spuštěním našeho algoritmu nezískáváme shodné výsledky, proto je vhodné výsledky porovnat. Můžeme tak posoudit, zda byly vytvořeny podobné nebo stejné shluky a získat tak objekty, které jsou si opravdu nejvíce podobné.

Ve většině případů vznikl nejméně jeden prázdný shluk. Dále vždy vznikl jeden shluk, který obsahoval velký počet motýlů, např. na prvním obrázku je to shluk reprezentovaný 17. chromozomem a obsahující 53 motýlů, u druhého obrázku je to 20. shluk obsahující 57 motýlů.

Vždy nám vznikaly jednoprvkové shluky, které byly nejčastěji tvořeny motýli 66, 111, 119. Pokud tito motýli byli ve shluku s jinými, tak daný shluk nebyl tvořen více jak 5 objekty. Číslo 66 reprezentuje motýla Ohniváčeka černoskvrnného (Lycaena tityrus = LYCTIT), č. 8 Perleťovce prostředního (Argynnis adippe=ARGADI) a č. 111 Soumračníka mochnového (Pyrgus serratulae=PYRSER).

Při shlukování k sobě byli vždy přiřazeni motýli č. 30, 31, 32, 60, 61, 62, 64, 80, 81, 82 a 83. O těchto motýlech lze říci, že se chovají nejpodobněji. Vyskytují

se ve stejných územích, tedy v lokalitách, které jsou si podobné např. ve složení půdy, povodí řek, zástavbě, atd.

Výsledky o přiřazení objektů do shluků jsou uvedeny v souboru *Chromozom-Přiřazení.xls*.

# 7. Závěr

Cílem práce bylo nastudovat problematiku shlukové analýzy a její metody, následně provést aplikaci na praktických příkladech. Dalším cílem bylo porovnat základní shlukovací metody s alternativními metodami.

V práci jsem se zaměřila na základní metody shlukové analýzy, jako je metoda nejbližšího souseda, metoda nejvzdálenějšího souseda, centroidní metoda a metoda $k$ průměrů. Rychlé řešení těchto metod nám poskytuje program MATLAB, který obsahuje zabudované příkazy. Dané příkazy jsem aplikovala na příkladu z Poslanecké sněmovny. Rozdílné výsledky daného příkladu z hierarchických metod jsou způsobeny různou volbou výpočtu vzdáleností mezi objekty. U metody $k$ průměrů je nutné si uvědomit, že na začátku jsou objekty voleny náhodně jako reprezentanti počátečních shluků, proto se mohou jednotlivé výsledky lišit. U této metody je důležité znát optimální počet shluků. V našem případě jsem u algoritmu $k$ průměrů sledovala četnosti přiřazení objektů do shluků. Na základě toho jsem stanovila optimální počet shluků.

Pro lepší pochopení principu hierarchických metod jsem vytvořila algoritmus metody nejbližšího souseda, na kterém můžeme vidět postupné shlukování jednotlivých objektů do jednoho shluku.

Další řešení základních statistických metod jsem ukázala pomocí programu SAS. Ten nám poskytuje rychle výsledky i pokud máme velkou vstupní datovou matici. Aplikaci shlukování pomocí SASu jsem ukázala na příkladu výskytu 123 druhů motýlů ve 113 lokalitách. Pro hierarchické metody nám SAS poskytuje přehledné řešení shlukování, v jakých iteracích došlo ke sloučení daných objektů. U metody $k$ průměrů můžeme určit kolik iterací je zapotřebí pro přířazení objektů do shluků, také kolik objektů padlo do daného shluku. Už je ale velmi obtížné zjistit, kam byly objekty přiřazeny.

Hlavním cílem práce bylo nastudovat alternativní přístupy pro shlukování. V práci jsem se zaměřila na neuronové sítě, konkrétně na Kohonenovy samoorganizující mapy a na genetické algoritmy a to na GA $k$ průměrů. Pro aplikaci jsem si naprogramovala GA $k$ průměrů, který jsem použila na příkladu výskytu

motýlů. Tento algoritmus nám poskytuje rychlé řešení shlukování. Na začátku je však nutné si zvolit do kolika shluků chceme, aby se dané objekty (motýli) přiřadily. Pro příslušný počet iterací se výpočet ustálí a nedochází ke změnám ve shluku.

Výsledky jednotlivých metod jsou v práci pro lepší představu graficky znázorněny.

V diplomové práci jsou uvedeny základní informace o shlukové analýze a jejich alternativních přístupech. Práce zároveň dává čtenáři prostor k dalšímu studiu týkající se dané problematiky a seznamuje ho s možnostmi, které poskytuje MATLAB a SAS. Moje zkušenosti s užitím shlukové analýzy pak může využít u jiných úloh.

# Literatura


[1] A Tutorial on Clustering Algorithms.
http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/index.html
[online 15.3.2010]

[2] Genetický algoritmus.
http://cs.wikipedia.org/wiki/Genetick%C3%BD_algoritmus[online 1.4.2010]

[3] Hebák, P., Hustopecký, J., Malá, I.: Vicerozměrné statistické metody 1, Informatorium, Praha, 2003.

[4] Hebák, P., Hustopecký, J., Malá, I.: Vicerozměrné statistické metody 2, Informatorium, Praha, 2003.

[5] Hebák, P., Hustopecký, J., Malá, I.: Vicerozměrné statistické metody 3, Informatorium, Praha, 2005.

[6] Kay, A.: Co jsou to umělé neuronové sítě?
http://scienceworld.cz/technologie/co-jsou-to-umele-neuronove-site-4077
[online 30.3.2010]

[7] Křivan, M.: Úvod do umělých neuronových sítí, Oeconomica VŠE, Praha, 2008.

[8] Lukasová, A., Šarmanová, J.: Metody shlukové analýzy, SNTL, Praha, 1985.

[9] Luner, P.: Jemný úvod do genetických algoritmů.
http://cgg.mff.cuni.cz/ pepca/prg022/luner.html [online 1.4.2010]

[10] Moore, A., W.: K-means and Hierarchical Clustering.
http://www.autonlab.org/tutorials/kmeans11.pdf [online 2.4.2010]

[11] Moustafa, E., A.: A Plot for Visualizing Multivariate Data, 2003.
http://www.galaxy.gmu.edu/interface/I03/ [online 2.4.2010]

[12] Ošmera, P.: Evoluční algoritmy a jejich aplikace, VŠE, Praha, 2008.

[13] Pechanec, V., Janoška, Z., Geletič, J.: Datové soubory s četnostmi výskytu rostlin, motýlů a údaji o faktorech půdy v lokalitě Bílých Karpat, UP Olomouc, 2010.

[14] Poslanecká sněmovna. www.psp.cz [online 1.2.2010]

[15] Řezánková, H., Húsek, D., Snášel, V.: Shluková analýza dat, Kamil Mařík-Profesional Publishing, Praha, 2007.

[16] Tryon C.R., Bailey, D.E.: Cluster analysis, McGraw-Hill Book Company, 1970.

[17] Zelinka, I.: Umělá inteligence I.: Neuronové sítě a genetické algoritmy, Vutium, Brno, 1998.