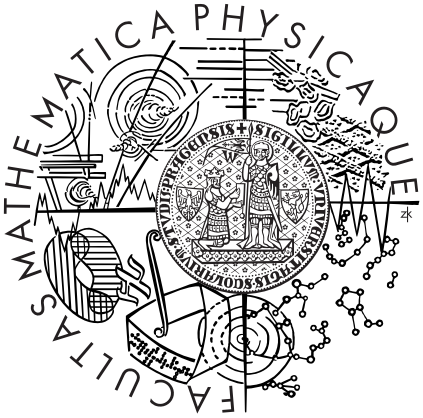
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Marek Dvořák

Metody shlukové analýzy a jejich aplikace v marketingu

Katedra pravděpodobnosti a matematické statistiky

Vedoucí diplomové práce: RNDr. Pavel Vaněček, Ph.D.

Studijní program: Matematika

Studijní obor: Pravděpodobnost, matematická statistika a ekonometrie

Studijní plán: Ekonometrie

2008

Mé poděkování patří RNDr. Pavlu Vaněčkovi, Ph.D. za volbu zajímavého a v současné době velmi rozvíjejícího se tématu, náměty a připomínky, které pomohly zkvalitnit tuto práci, a také za reálná data, na kterých jsem mohl vyzkoušet popsané metody.

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

V Praze dne 15. dubna 2008 Marek Dvořák

Obsah

Název práce: Metody shlukové analýzy a jejich aplikace v marketingu
Autor: Bc. Marek Dvořák
Katedra (ústav): Katedra pravděpodobnosti a matematické statistiky
Vedoucí diplomové práce: RNDr. Pavel Vaněček, Ph.D.
e-mail vedoucího: vanecekpavel@karlin.mff.cuni.cz

Abstrakt: V předložené práci studujeme algoritmy shlukové analýzy a jejich aplikace na data. V úvodu rozlišujeme jednotlivé typy dat a míry nepodobnosti mezi pozorovanými objekty i mezi jednotlivými shluky, abychom mohli provést shlukování a kvantitativně ohodnotit vzniklé rozklady. Kapitola 2 se věnuje nehierarchickým algoritmům shlukové analýzy a metodám pro nalezení optimálního počtu shluků. V další části je krátce uvedené zobecnění rozdělovacích metod – fuzzy shlukování. Hierarchické metody shlukové analýzy jsou popsány v kapitole 3, kde opět nechybí kritéria pro posouzení kvality shlukování. V závěru této kapitoly je provedeno porovnání všech shlukovacích metod vzhledem k navrženým funkcionálům kvality rozkladu. Kapitola 4 se věnuje archetypální analýze a algoritmům pro nalezení archetypů. Všechny výše zmíněné kapitoly obsahují ilustrační příklady. Hlavní aplikační část lze nalézt v kapitole 5, kde zkoumáme data z výzkumu životního stylu v ČR.

Klíčová slova: archetypy, hierarchické metody, k-means, míry nepodobnosti, shluková analýza

Title: Cluster analysis methods and their applications in marketing
Author: Bc. Marek Dvořák
Department: Department of Probability and Mathematical Statistics
Supervisor: RNDr. Pavel Vaněček, Ph.D.
Supervisor's e-mail address: vanecekpavel@karlin.mff.cuni.cz

Abstract: In this work we study algorithms for cluster analysis and their application to the real data. In the beginning, the various types of data are presented. We define dissimilarity measures for each type of data and for clusters to be able to do the clustering and evaluate the separation quantitatively. In the Chapter 2, there are described partitioning algorithms and some criteria to determine the optimal number of clusters. A part of this chapter is devoted to the fuzzy cluster analysis which is a generalization of partitioning techniques. Hierarchical algorithms are characterized in Chapter 3 as well as criteria for choosing the appropriate method. In the very end of this chapter, there is a comparison of all the methods in terms of various types of the separation functionals. Archetypal analysis, which is another data mining instrument, is described in Chapter 4. All chapters include illustration examples of usage. The main application part is the last chapter of this diploma thesis and it's based on the lifestyle survey in the Czech republic.

Keywords: archetypes, cluster analysis, dissimilarity measures, hierarchical algorithms, k-means,

Značení

Zpravidla budeme mít k dispozici soubor pozorování v p-rozměrném prostoru.

- *Objektem*, nebo někdy také *prvkem* $\mathbf{x}_i$ budeme rozumět p-rozměrný sloupcový vektor pozorování
$$\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top.$$
- Indikátor jevu $[\mathbf{x}_i \in S_k]$ je funkce
$$\mathbb{I}_{[\mathbf{x}_i \in S_k]} = \begin{cases} 1, & \text{když } \mathbf{x}_i \in S_k \\ 0, & \text{když } \mathbf{x}_i \notin S_k. \end{cases}$$
- Pro matici $\mathbf{A}$ značíme $|\mathbf{A}|$ její determinant, $\mathsf{tr}\,\mathbf{A}$ její stopu a $\mathsf{rank}\,\mathbf{A}$ její hodnost.
- $\mathsf{card}\,S$ je označení pro počet prvků množiny S.
- Pro spojitě diferencovatelnou funkci $f : \mathbb{R}^n \to \mathbb{R}$ podle všech proměnných definujeme její gradient v bodě $\mathbf{a} \in \mathbb{R}^n$ jako vektor
$$\nabla f(\mathbf{a}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{a}), \ldots, \frac{\partial f}{\partial x_n}(\mathbf{a}) \right)^\top.$$

Pro přehlednost bude v kapitolách 1, 2 a 3 rezervován index i (příp. i' a i'') pro číslování vektorů, indexem j budeme přistupovat k jednotlivým složkám vektoru a index k bude souviset s číslováním jednotlivých shluků. Budeme uvažovat N vektorů pozorování (každý p-rozměrný) a rozklady do K shluků. Index i bude tedy vždy probíhat množinou $1, \ldots, N$, index j množinou $1, \ldots, p$ a konečně index k množinou $1, \ldots, K$.

Kapitola 1

Metody shlukové analýzy

1.1 Cíle shlukové analýzy

Shluková analýza, nazývaná také často jako *segmentace dat*, zahrnuje mnoho postupů, jejichž společným cílem je najít v dané množině objektů podmnožiny (tzv. *shluky*, anglicky *clusters*) podobných prvků. Při určování těchto podmnožin se snažíme, aby prvky uvnitř podmnožin si byly co nejvíce „podobné“ a naopak prvky z různých shluků byly mezi sebou co nejvíc „odlišné“. Podobnost resp. odlišnost lze charakterizovat zavedením vhodné metriky na prostoru, kde se vyskytují pozorované vektory. Nalezení vlastností, které daný shluk charakterizují, je jeden ze základních úkolů shlukové analýzy.

S rostoucími možnostmi zaznamenávání dat a tvorbou datových skladů může být dalším úkolem shlukové analýzy redukce dat pro jejich následné zpracování. Z velkého množství dat se vyberou jen někteří „reprezentanti“, kteří se posléze dále zkoumají. Možností je také detailnější analýza dat uvnitř jednotlivých shluků.

Shlukování lze rovněž aplikovat na proměnné. Má-li datový soubor velké množství proměnných, pak lze vybrat ty, jejichž hodnoty jsou si blízké, a nahradit je jedinou proměnnou.

Nyní si popíšeme úlohu shlukové analýzy formálním způsobem:

Pro množinu objektů (pozorování) $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ nalezneme rozklad $S_1, \ldots, \ldots, S_K$ takový, že

$$\bigcup_{k=1}^{K} S_k = \mathcal{X}, \qquad S_k \cap S_{k'} = \emptyset, \quad k \neq k'. \tag{1.1}$$

Úloha (1.1) v sobě zahrnuje i problém nalezení optimálního počtu shluků K, které nemusí být předem dáno. Přitom se snažíme, aby K bylo výrazně nižší než počet pozorování. Jestliže $\mathbf{x}_i \in S_k$, pak řekneme, že i-té pozorování bylo zařazeno do k-tého shluku.

Algoritmy shlukové analýzy se rozdělují do dvou hlavních skupin. *Nehierarchické* metody obsahují postupy, kde se pracuje s konkrétním předem daným počtem

shluků, do kterých se provede rozklad datového souboru. *Hierarchické* postupy naproti tomu pracují s rozklady do 2, 3, až $N-1$ shluků tak, že postupně buď spojují nebo naopak štěpí vytvořené shluky. Více bude o nehierarchických metodách pojednávat kapitola 2, hierarchické metody pak rozebereme v kapitole 3. Zobecněním nehierarchických metod se budeme věnovat v části o fuzzy shlukové analýze.

Výsledkem shlukové analýzy je nalezený rozklad N objektů do K shluků. Protože ten může být pokaždé jiný v závislosti na zvolené metodě, existují funkce, které pro různá shlukování posuzují jejich kvalitu. Věnována jim bude část kapitol 2 a 3. Porovnáním hodnot těchto funkcí budeme moci později kvantitativně porovnávat jednotlivé segmentace a následně rozhodnout o vhodnosti použitých algoritmů.

V celé práci pracujeme s volně šiřitelným softwarem `R`. Kratší programové kódy budeme uvádět přímo do textu, procedury pak budou uvedeny v rámci příloh a na přiloženém nosiči CD.

1.2 Typy dat

Použití algoritmů, které realizují segmentaci dat, je závislé na druhu proměnných. Ty mohou být buď spojité nebo diskrétní. Rozlišíme tedy jednotlivé druhy proměnných a připomeneme jejich vlastnosti:

- *Kategoriální (kvalitativní)* proměnné jsou proměnné, kde vytříděním jednotek souboru podle takovéto proměnné vznikají skupiny nebo kategorie. Rozlišujeme dvě skupiny kategoriálních dat:
  - *Nominální (jmenné, názvové)* proměnné jsou kategoriální proměnné, které popisují jména nebo skupiny objektů bez důrazu na jejich pořadí. Údaje o barvě očí respondentů (1 = modrá, 2 = hnědá, 3 = zelená) jsou typickým příkladem nominálních proměnných. Zde je nutné podotknout, že vyšší číslo neznamená vyšší prioritu, jediným smyslem je rozlišit dané atributy a nikoliv je řadit podle důležitosti. O dvou hodnotách nominální proměnné lze pouze konstatovat, že jsou buď stejné nebo různé. Pod tuto kategorii řadíme i speciální případ − tzv. binární proměnné − tedy proměnné nabývající pouze dvou hodnot. Často se jedná o data z odpovědí typu „ano/ne“, popř. informace o pohlaví (muž/žena).
  - *Ordinální (pořadové)* proměnné jsou kategoriální proměnné, vyjadřující přirozené pořadí jejich možných hodnot. Jedná se například o hodnocení ve škole (1 = výborně, 2 = chvalitebně, 3 = dobře, atd.). O jejich variantách lze proto konstatovat nejen to, že jsou stejné nebo různé, ale také je lze seřadit od nejmenší po největší.
- *Kvantitativní (metrické)* proměnné jsou proměnné s numerickým významem. Lze je nejen řadit podle velikosti, ale je také možné změřit, o kolik je jedna větší než druhá.

– Spojité – např. nejvyšší dosažená teplota ve vybraných městech ČR pro každý měsíc roku.
– Diskrétní – např. počet vstřelených branek fotbalistů klubu za celou kariéru.

1.3 Úplné shlukování

Jak již bylo řečeno, základním cílem shlukové analýzy je vytvoření kompaktních a pokud možno separovaných shluků. K nalezení optimálního rozkladu bychom potřebovali vyšetřit všechna možná přiřazení N objektů do K shluků. Následující věta udává, kolik takových rozkladů existuje:

Věta 1.1: *Počet způsobů, jak rozdělit N různých pozorování do K různých shluků tak, aby žádný shluk nezůstal prázdný, je*

$$\frac{1}{K!} \cdot \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^N. \tag{1.2}$$

Důkaz: Viz [Duran a Odell, 1974], str. 37, věta 1.1. Věta je odvozena přes vytvořující funkce. Číslo (1.2) se nazývá *Stirlingovo číslo II. druhu.* ∎

Vidíme, že rozkladů (1.2) pro pevné N a K je bohužel velmi mnoho, např. už pro $N = 20$ a $K = 3$ je jich 580 606 446. Proto algoritmy užívají pro nalezení vhodného rozkladu vesměs iterativní postupy, které nejsou výpočetně tolik náročné. V každém kroku pozměňujeme přiřazování do jednotlivých shluků a vylepšujeme tak rozklad. Cenou za jednodušší a časově méně náročný výpočet je nalezení pouze lokálně-optimálních řešení.

1.4 Míry vzdáleností objektů

Pro měření, jak jsou si které objekty nebo shluky blízké, popř. vzdálené, se používají tzv. *koeficienty podobnosti*, popř. *koeficienty nepodobnosti*. Tyto koeficienty se definují v závislosti na charakteru dat a mají svou logickou interpretaci.

Míry pro objekty s kvantitativními proměnnými

Jeden z nejběžnějších způsobů vyjádření podobnostních vztahů mezi objekty jsou metriky vycházející z geometrického modelu dat.

Připomeňme, že metrika $d : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ je funkce splňující:

1. $d(\mathbf{x}_i, \mathbf{x}_{i'}) = 0 \Leftrightarrow \mathbf{x}_i = \mathbf{x}_{i'}$,
2. $d(\mathbf{x}_i, \mathbf{x}_{i'}) \geq 0$ (nezápornost),

3. $d(\mathbf{x}_i, \mathbf{x}_{i'}) = d(\mathbf{x}_{i'}, \mathbf{x}_i)$ (symetrie),

4. $d(\mathbf{x}_i, \mathbf{x}_{i'}) \leq d(\mathbf{x}_i, \mathbf{x}_{i''}) + d(\mathbf{x}_{i''}, \mathbf{x}_{i'})$ (trojúhelníková nerovnost),

pro všechna $\mathbf{x}_i$, $\mathbf{x}_{i'}$, $\mathbf{x}_{i''} \in \mathbb{R}^p$.
Nejčastěji se používají následující metriky:

- Eukleidovská vzdálenost

$$d_2(\mathbf{x}_i, \mathbf{x}_{i'}) = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}. \tag{1.3}$$

  Často se používá v některých algoritmech také druhá mocnina eukleidovské vzdálenosti $d_2^2(\mathbf{x}_i, \mathbf{x}_{i'})$.

- Hemmingova vzdálenost

$$d_1(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^{p} |x_{ij} - x_{i'j}|. \tag{1.4}$$

  Tato vzdálenost se v anglicky psané literatuře označuje také jako *City-block* nebo *Manhattan distance*.

- Supremální vzdálenost

$$d_\infty(\mathbf{x}_i, \mathbf{x}_{i'}) = \max_{1 \leq j \leq p} |x_{ij} - x_{i'j}|. \tag{1.5}$$

- Mahalanobisova vzdálenost

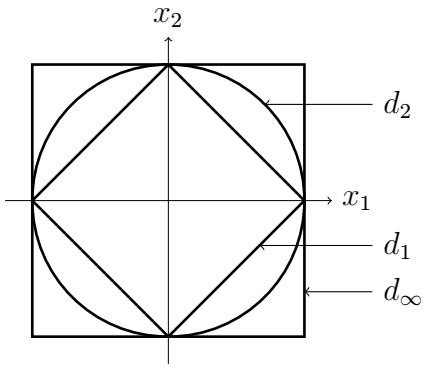
$$d_{\mathrm{M}}(\mathbf{x}_i, \mathbf{x}_{i'}) = \left((\mathbf{x}_i - \mathbf{x}_{i'})^\top \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{x}_{i'}) \right)^{\frac{1}{2}},$$

  kde $\mathbf{S}$ je výběrová varianční matice dat.

Vzdálenosti (1.3), resp. (1.4), resp. (1.5) jsou speciálními případy tzv. Minkowského metriky

$$d_m(\mathbf{x}_i, \mathbf{x}_{i'}) = \left(\sum_{j=1}^{p} |x_{ij} - x_{i'j}|^m \right)^{\frac{1}{m}}$$

pro $m = 2$, resp. pro $m = 1$, resp. pro $m \to \infty$. Geometrické útvary na obrázku 1.1 vyznačují množiny bodů v prostoru $\mathbb{R}^2$ se stejnou eukleidovskou, Hemmingovou a supremální vzdáleností od počátku soustavy souřadnic.



Obrázek 1.1: Množiny bodů v $\mathbb{R}^2$ se stejnou eukleidovskou, Hemmingovou a supremální vzdáleností od středu.

Míry pro objekty s kategoriálními proměnnými

Předpokládejme nejprve, že máme k dispozici binární datový soubor, tedy všechny složky vektorů pozorování $\mathbf{x}_i$, $i = 1, \ldots, N$ nabývají pouze hodnot 0 a 1. Jestliže $x_{ij} = 1$, pro nějaké i a j, označme to jako *pozitivní výsledek*, v případě $x_{ij} = 0$, pro nějaké i a j, budeme mluvit o *negativním výsledku*. Pro 2 binární vektory $\mathbf{x}_i$ a $\mathbf{x}_{i'}$ budeme měřit tzv. *koeficienty podobnosti* $a_{\mathrm{SM}}(\mathbf{x}_i, \mathbf{x}_{i'})$. Jedná se o číslo z intervalu $[0, 1]$, vyšší hodnoty znamenají větší blízkost obou vektorů.

- Koeficient prosté shody (SM-koeficient[1])

$$a_{\mathrm{SM}}(\mathbf{x}_i, \mathbf{x}_{i'}) = \frac{1}{p} \cdot \sum_{j=1}^{p} \Big(x_{ij} x_{i'j} + (1 - x_{ij})(1 - x_{i'j}) \Big). \tag{1.6}$$

  V případě shodnosti vektorů $\mathbf{x}_i$ a $\mathbf{x}_{i'}$ na j-té pozici je právě jeden sčítanec uvnitř sumy nulový a jestliže se vektory na j-té pozici liší, jsou oba sčítanci rovny nule. SM-koeficient tedy vyjadřuje relativní četnost shodných znaků. Vzorec (1.6) lze dále upravit a platí

$$a_{\mathrm{SM}}(\mathbf{x}_i, \mathbf{x}_{i'}) = 1 - \frac{1}{p} \cdot \sum_{j=1}^{p} (x_{ij} + x_{i'j} - 2x_{ij}x_{i'j}) = 1 - \frac{1}{p} \cdot \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2.$$

  Poslední rovnost platí proto, že $x_{ij} = 0 \Leftrightarrow x_{ij}^2 = 0$ a rovněž $x_{ij} = 1 \Leftrightarrow x_{ij}^2 = 1$, neboť složky vektorů nabývají pouze hodnot 0 a 1.

Poznamenejme, že jako koeficient nepodobnosti lze použít $1 - a_{\mathrm{SM}}(\mathbf{x}_i, \mathbf{x}_{i'})$. Takto lze získat koeficienty nepodobnosti i pro níže uvedené podobnostní koeficienty.

[1] Zkratka SM pochází z anglického názvu Simple Matching coefficient

V případě SM-koeficientu (1.6) se hodnotí absence daného znaku (= 0) stejně jako jeho přítomnost (= 1) − tedy obě varianty (např. muž/žena) pokládáme za stejně důležité. Binárním datům, kde jsou obě varianty rovnocenné z hlediska priority, říkáme *symetrická binární data*. Často se však setkáváme s *asymetrickými binárními daty*. V tomto případě je přítomnost znaku považována za důležitější než jeho nepřítomnost. Příkladem může být výskyt (= 1) nebo absence (= 0) určitého slova na vybraných webových stránkách. Stránky obsahující toto slovo mohou mít společné ještě některé další znaky, např. obrázky, a proto je pokládáme za důležitější. Stránky, na kterých vybrané slovo není, mohou být v dalších vlastnostech zcela odlišné. Koeficient podobnosti pro asymetrická binární data tak nezohledňuje případ shody dvou negativních výsledků. Je to např.

- Russelův-Raoův koeficient (RR-koeficient)

$$a_{\mathrm{RR}}\left(\mathbf{x}_i, \mathbf{x}_{i'}\right) = \frac{1}{p} \cdot \sum_{j=1}^{p} x_{ij} x_{i'j}.$$

  Jedná se o relativní četnost shodně pozitivních výsledků, neboť $x_{ij}\ x_{i'j} \neq 0 \Leftrightarrow (x_{ij} = 1 \wedge x_{i'j} = 1)$.

- Jaccardův koeficient (J-koeficient)

$$a_{\mathrm{J}}\left(\mathbf{x}_i, \mathbf{x}_{i'}\right) = \frac{a}{a+b+c},$$

  kde

$$a = \sum_{j=1}^{p} x_{ij} x_{i'j}, \quad b = \sum_{j=1}^{p} x_{ij}(1 - x_{i'j}), \quad c = \sum_{j=1}^{p} (1 - x_{ij}) x_{i'j}.$$

  Výraz $b + c$ tedy udává počet složek, kde se oba vektory $\mathbf{x}_i, \mathbf{x}_{i'}$ liší.

Nyní se podívejme na nominální proměnné, které mohou nabývat více hodnot (stavů). Označme $u_{ii'} = \mathsf{card}\,\{j \in \{1, 2, \ldots, p\} : x_{ij} = x_{i'j}\}$ počet těch proměnných vektorů $\mathbf{x}_i$, $\mathbf{x}_{i'}$, pro které je $x_{ij} = x_{i'j}$. Pak koeficientem podobnosti může být např.

- SM-koeficient

$$a_{\mathrm{SM}}\left(\mathbf{x}_i, \mathbf{x}_{i'}\right) = \frac{u_{ii'}}{p}.$$

Věnujme se krátce ještě míře podobnosti pro data, která obsahují vesměs ordinální proměnné. Tyto proměnné se používají zejména pro oceňování, hodnocení kvality, např. 1 = nesnášet, 2 = nelíbit se, 3 = neutrální názor, 4 = mít rád, 5 = zbožňovat. Budeme předpokládat, že j-tá ordinální proměnná může nabývat hodnot $1, 2, \ldots, M_j$, $M_j \in \mathbb{N}$, $M_j > 2$, $j = 1, \ldots, p$. Je-li totiž $M_j = 2$ pro nějaké j, pak j-tá proměnná je binární a tento případ jsme řešili výše. Nechť r_{ij} je hodnota

i-tého pozorování v j-té ordinální proměnné. Pak [Kaufman a Rousseeuw, 1990] doporučuje převést ordinální proměnné na kvantitativní pomocí transformace

$$z_{ij} := \frac{r_{ij} - 1}{M_j - 1}, \quad i = 1, \ldots, N,\ j = 1, \ldots, p. \tag{1.7}$$

Platí $0 \leq z_{ij} \leq 1$, $i = 1, \ldots, N$, $j = 1, \ldots, p$. Na vektory $\mathbf{z}_i = (z_{i1}, \ldots, z_{ip})^\top$ pak lze pohlížet jako na pozorování s vesměs kvantitativní informací. Další koeficienty nepodobnosti lze nalézt např. v [Hebák a kol., 2005].

Míry pro objekty se smíšenými proměnnými

V praktických aplikacích se často soubor pozorování skládá z více druhů proměnných. Příkladem mohou být data o zákaznících mobilního operátora, kde jsou k dispozici informace o pohlaví (binární proměnná), bydlišti (nominální proměnná), počtu provolaných vteřin za měsíc (kvantitativní proměnná). Pak tato data lze zpracovat různými způsoby, jak se navrhuje v [Kaufman a Rousseeuw, 1990], str. 34. Můžeme provést shlukovou analýzu dat pro každý typ proměnných zvlášť a v závěru porovnat výsledky. Jestliže je většina objektů rozřazena stejným způsobem, lze takový postup doporučit.

Jestliže výsledky těchto analýz jsou vzájemně rozporuplné, pak je třeba zvolit jiné postupy. Lze třeba ignorovat různé typy dat a provést shlukování, jako by všechna data byla kvantitativní. Tento postup se nedoporučuje v případě, že data obsahují asymetrické binární nebo nominální proměnné s více jak dvěma stavy.

Opačný přístup, jak se vypořádat s různými kategoriemi proměnných v datovém souboru, je možnost data převést na binární proměnné. Pro kvantitativní data lze zavést tzv. práh překročení. Jestliže proměnná nabyde hodnoty větší než je hodnota prahu, přiřadíme jí hodnotu 1, a 0 v opačném případě. V případě kategoriálních dat s více jak dvěma stavy lze definovat $M_j - 1$ binárních proměnných. Můžeme také sloučit stavy do dvou základních, ale těmito předběžnými úpravami se ztratí část informace obsažená v původním datovém souboru, což je vždy považováno za nevýhodu.

Literatura [Kaufman a Rousseeuw, 1990], str. 35, doporučuje použít na smíšená data míru nepodobnosti

$$d(\mathbf{x}_i, \mathbf{x}_{i'}) = \frac{\sum_{j=1}^{p} \mathbb{I}_{ii',j} \cdot d_{ii',j}}{\sum_{j=1}^{p} \mathbb{I}_{ii',j}}, \tag{1.8}$$

kde

$$\mathbb{I}_{ii',j} = \begin{cases} 1, & \text{když } x_{ij} \text{ a } x_{i'j} \text{ nejsou chybějící pozorování,} \\ 0, & \text{jinak.} \end{cases}$$

Dále $\mathbb{I}_{ii',j} = 0$, jestliže proměnná j je asymetrická binární a platí $x_{ij} = x_{i'j} = 0$.

Předpokládejme, že x_{ij} i $x_{i'j}$ nejsou chybějící pozorování, jinak by nemělo smysl počítat $d_{ii',j}$.

Jestliže proměnná j je binární nebo nominální, pak $d_{ii',j}$ je definováno jako

$$\begin{aligned} d_{ii',j} &= 1, \text{ jestliže } x_{ij} \neq x_{i'j} \\ &= 0, \text{ jestliže } x_{ij} = x_{i'j}. \end{aligned}$$

Jestliže proměnná j je kvantitativní, pak

$$d_{ii',j} = \frac{|x_{ij} - x_{i'j}|}{R_j}, \tag{1.9}$$

kde $R_j = \max_i\{x_{ij}\} - \min_i\{x_{ij}\}$ je maximální rozpětí j-té proměnné.

Je-li j-tá proměnná ordinální, přeškálujeme ji pomocí (1.7) na kvantitativní a dále použijeme vzorec (1.9).

Poznámky 1.2:

1. Jestliže pro všechna $j = 1, \ldots, p$ platí $\mathbb{I}_{ii',j} = 0$, pak musíme objekt i nebo i' vynechat.

2. Pokud všechny proměnné jsou binární symetrické, pak si vzorec nepodobnosti (1.8) odpovídá s a_{SM} koeficientem podobnosti (1.6).

Vzdálenost mezi objekty se reprezentuje pomocí *matice vzdáleností* $\mathbf{D} = \{d_{ii'}\}$. Jedná se o symetrickou matici typu $N \times N$ s nulami na hlavní diagonále, takovou, že $d_{ii'} = d(\mathbf{x}_i, \mathbf{x}_{i'})$, kde d je zvolená míra nepodobnosti. Její výpočet provádí v softwaru `R`, v balíku `cluster` příkaz

```
daisy(x, metric = "...", stand = ..., type = list.x)
```

`x`	matice dat, sloupce odpovídají proměnným, řádky pozorováním,
`metric`	metrika použitá k určení nepodobností mezi objekty,
`stand`	logický znak: je-li `T`, pak se provede standardizace dat odečtením průměru $\overline{x}_j$ a vydělením odchylkou sm_j, kde

$$\begin{aligned} \overline{x}_j &:= \frac{1}{N} \cdot \sum_{i=1}^{N} x_{ij}, \quad j = 1, \ldots, p, \\ sm_j &:= \frac{1}{N} \cdot \sum_{i=1}^{N} |x_{ij} - \overline{x}_j|, \quad j = 1, \ldots, p. \end{aligned}$$

`type`	objekt typu `list`, jehož použití si ukážeme na příkladě.

Načítáme-li data z externího souboru, software `R` přistupuje ke všem datům jako kvantitativním. Proto je třeba vložit do datového souboru informaci o typu proměnných. Ukážeme si to na následujícím příkladě převzatém z [Kaufman a Rousseeuw, 1990]:

Příklad 1.3: V souboru `rostliny.txt` máme k dispozici seznam 18 zahradních květin s následujícími typy proměnných:

1. rostlina může stát ve stínu − symetrická binární proměnná
   hodnoty: 1 = ano, 0 = ne,
2. rostlina může být v mrazu − symetrická binární proměnná
   hodnoty: 1 = ano, 0 = ne,
3. výška rostliny − kvantitativní proměnná
4. rostlina má hlízu − asymetrická binární proměnná; dvě rostliny s hlízou mohou mít společné ještě nějaké další znaky, kdežto rostliny bez hlízy mohou růst každá za úplně jiných podmínek
   hodnoty: 1 = ano, 0 = ne,
5. barva květu − nominální proměnná
   hodnoty: 1 = bílá, 2 = žlutá, 3 = růžová, 4 = červená, 5 = modrá
6. vhodná půda − ordinální proměnná; čím vyšší kategorie, tím větší vlhkost půdy
   hodnoty: 1 = suchá, 2 = normální, 3 = vlhká
7. preference dotazovaných osob − ordinální proměnná
   hodnoty: 1 (nejméně oblíbená) až 18 (nejvíce oblíbená)

V `R` načteme data z externího souboru a ke každé proměnné přidáme informaci o jejím typu pomocí následujících příkazů:

```
set.ordinal<-c(6,7)
for (i in set.ordinal) {data[,i] <- ordered(data[,i])}
set.nominal<-c(5)
for (i in set.nominal) {data[,i] <- factor(data[,i])}
list.data<-list(symm=c(1,2), asymm=c(4))
```

Následně zavoláním

```
daisy(data,type=list.data)
```

získáme matici vzdáleností pro daný datový soubor. Zatímco informace o nominálních a ordinálních proměnných jsme uložili pomocí příkazů `ordered` a `factor` přímo do matice `data`, specifikace (a)symetrických binárních proměnných se připisuje jako další argument funkce `daisy`. Kdybychom zavolali na stejná data

```
daisy(data, metric="euclidean", stand=T, type=list.data),
```

dostali bychom stejnou matici vzdáleností jako v předchozím případě. Důvodem je skutečnost, že jestliže alespoň 1 proměnná v datovém souboru `data` je nominální, ordinální nebo (a)symetrická binární, jsou argumenty `metric` a `stand` ignorovány a při výpočtu nepodobnosti mezi objekty se ihned použije nepodobnostní koeficient (1.8). ♢

1.5 Míry nepodobnosti shluků

Zatímco tzv. nehierarchické postupy používají k vyřešení úloh shlukové analýzy pouze koeficienty nepodobnosti mezi pozorovanými objekty, hierarchické postupy potřebují definovat i nepodobnosti mezi shluky (tedy mezi množinami objektů). Uvažujme shluky G a H, které obsahují N_G a N_H pozorovaných objektů. Zvolme v případě kvantitativních dat nějakou metriku d pro měření vzdáleností objektů v $\mathbb{R}^p$, popř. koeficient nepodobnosti mezi 2 objekty. Pro měření nepodobnosti shluků se používají následující funkce:

- Single linkage (SL)

$$d_{\mathrm{SL}}(G,H) = \min_{\substack{\mathbf{x}\in G\\ \mathbf{y}\in H}} d(\mathbf{x},\mathbf{y}). \tag{1.10}$$

- Complete linkage (CL)

$$d_{\mathrm{CL}}(G,H) = \max_{\substack{\mathbf{x}\in G\\ \mathbf{y}\in H}} d(\mathbf{x},\mathbf{y}). \tag{1.11}$$

- Průměrná vzdálenost (Ø)

$$d_{\emptyset}(G,H) = \frac{1}{N_G N_H}\cdot\sum_{\mathbf{x}\in G}\sum_{\mathbf{y}\in H} d(\mathbf{x},\mathbf{y}). \tag{1.12}$$

- Centroidní vzdálenost (C)

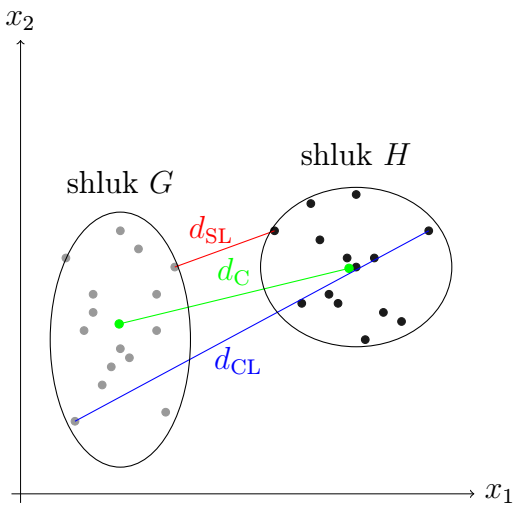
$$d_{\mathrm{C}}(G,H) = d(\overline{\mathbf{x}}_G,\overline{\mathbf{x}}_H), \tag{1.13}$$

  kde

$$\overline{\mathbf{x}}_G = \frac{1}{N_G}\cdot\sum_{\mathbf{x}\in G}\mathbf{x},$$
$$\overline{\mathbf{x}}_H = \frac{1}{N_H}\cdot\sum_{\mathbf{y}\in H}\mathbf{y}.$$

- Kolmogorovova zobecněná vzdálenost (K,r)

$$d_{\mathrm{K},r}(G,H) = \left[\frac{1}{N_G N_H}\cdot\sum_{\mathbf{x}\in G}\sum_{\mathbf{y}\in H} d(\mathbf{x},\mathbf{y})^r\right]^{\frac{1}{r}}.$$



Obrázek 1.2: Znázornění single linkage, complete linkage a centroidní vzdálenosti pro případ kvantitativních dat v prostoru $(\mathbb{R}^2, d_2)$.

Grafické zobrazení vzdáleností (1.10), (1.11) a (1.13) v případě objektů z prostoru $\mathbb{R}^2$ je na obrázku 1.2.

Kapitola 2

Nehierarchické postupy

V následujících dvou kapitolách se budeme věnovat možnostem rozdělování dat do jednotlivých shluků, přičemž v žádném z uvedených algoritmů nebude třeba zavádět pravděpodobnostní modely ani hledat pravděpodobnostní rozdělení pro daná data. Níže popisované algoritmy shlukové analýzy se budou opírat pouze o vzdálenosti mezi pozorovanými objekty.

V této kapitole zmíníme některé *nehierarchické* nebo také *rozdělovací* metody, které dané objekty rozdělí do předem zvoleného počtu shluků $K > 1$. Každé pozorování je jednoznačně označeno indexem $i \in \{1, \ldots, N\}$ a je rovněž jednoznačně přiřazeno do shluku $k \in \{1, \ldots, K\}$. Zobrazení $C : \{1, \ldots, N\} \to \{1, \ldots, K\}$ definované předpisem

$$C(i) = k \tag{2.1}$$

se nazývá *funkce příslušnosti* a značí, že pozorování $\mathbf{x}_i$ je zařazeno do shluku S_k. V anglické literatuře se zobrazení C často značí termínem *encoder*. Cílem většiny nehierarchických metod shlukové analýzy je minimalizace tzv. ztrátové funkce W, která souvisí s mírou variability pozorování uvnitř shluků. Často se definuje jako

$$W(C) = \frac{1}{2} \cdot \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} \sum_{\{i':\, C(i')=k\}} d\left(\mathbf{x}_i, \mathbf{x}_{i'}\right), \tag{2.2}$$

kde $d\left(\mathbf{x}_i, \mathbf{x}_{i'}\right)$ je zvolená metrika mezi dvěma objekty. Ztrátová funkce samozřejmě závisí na funkci C, která určuje rozřazení pozorování do jednotlivých shluků.

2.1 Metoda k-means

Algoritmus k-means patří mezi nejpopulárnější algoritmy shlukové analýzy a je založen na postupném přesunování objektů mezi vytvářejícími se shluky. Lze použít v situacích, kdy máme všechna data kvantitativního charakteru, nebo je za kvantitativní považujeme. Algoritmus využívá jako míru nepodobnosti (= metriku) mezi

objekty druhou mocninu eukleidovské vzdálenosti, tedy

$$d(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = \|\mathbf{x}_i - \mathbf{x}_{i'}\|^2. \tag{2.3}$$

Dosadíme-li (2.3) do (2.2), dostáváme

$$W(C) = \frac{1}{2} \cdot \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} \sum_{\{i':\, C(i')=k\}} \|\mathbf{x}_i - \mathbf{x}_{i'}\|^2.$$

Statistika $W(C)$ tedy vyjadřuje součet odchylek pozorování uvnitř shluků. Lze také interpretovat jako vážený součet odchylek od průměru, jak ukáže následující věta.

Věta 2.1: *Označme počet pozorování zařazených do k-tého shluku jako*

$$N_k := \sum_{\{i:\, C(i)=k\}} 1 \;=\; \sum_{i=1}^{N} \mathbb{I}_{[C(i)=k]}$$

a průměr prvků v k-tém shluku

$$\overline{\mathbf{x}}_k := \frac{1}{N_k} \cdot \sum_{\{i:\, C(i)=k\}} \mathbf{x}_i.$$

Pak platí

$$W(C) = \sum_{k=1}^{K} N_k \sum_{\{i:\, C(i)=k\}} \|\mathbf{x}_i - \overline{\mathbf{x}}_k\|^2. \tag{2.4}$$

Důkaz: Tvrzení se ověří přímým výpočtem:

$$\begin{aligned}
W(C) \;&=\; \frac{1}{2} \cdot \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} \sum_{\{i':\, C(i')=k\}} \|\mathbf{x}_i - \mathbf{x}_{i'}\|^2 = \\
&=\; \frac{1}{2} \cdot \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} \sum_{\{i':\, C(i')=k\}} \|(\mathbf{x}_i - \overline{\mathbf{x}}_k) - (\mathbf{x}_{i'} - \overline{\mathbf{x}}_k)\|^2 = \\
&=\; \frac{1}{2} \cdot \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} \sum_{\{i':\, C(i')=k\}} \|\mathbf{x}_i - \overline{\mathbf{x}}_k\|^2 - \\
&-\; \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} \sum_{\{i':\, C(i')=k\}} (\mathbf{x}_i - \overline{\mathbf{x}}_k)^\top \cdot (\mathbf{x}_{i'} - \overline{\mathbf{x}}_k) + \\
&+\; \frac{1}{2} \cdot \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} \sum_{\{i':\, C(i')=k\}} \|\mathbf{x}_{i'} - \overline{\mathbf{x}}_k\|^2 \overset{(\bigstar)}{=}
\end{aligned}$$

$$\overset{(\bigstar)}{=} \frac{1}{2} \cdot \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} N_k \, \|\mathbf{x}_i - \overline{\mathbf{x}}_k\|^2 + \frac{1}{2} \cdot \sum_{k=1}^{K} \sum_{\{i':\, C(i')=k\}} N_k \, \|\mathbf{x}_{i'} - \overline{\mathbf{x}}_k\|^2 =$$

$$= \sum_{k=1}^{K} N_k \sum_{\{i:\, C(i)=k\}} \|\mathbf{x}_i - \overline{\mathbf{x}}_k\|^2.$$

Rovnost (★) platí proto, že

$$\sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} \sum_{\{i':\, C(i')=k\}} (\mathbf{x}_i - \overline{\mathbf{x}}_k)^\top (\mathbf{x}_{i'} - \overline{\mathbf{x}}_k) =$$

$$= \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} (\mathbf{x}_i - \overline{\mathbf{x}}_k)^\top \sum_{\{i':\, C(i')=k\}} (\mathbf{x}_{i'} - \overline{\mathbf{x}}_k) = 0,$$

čímž je tvrzení dokázáno. ∎

Naimplementovné algoritmy počítají ztrátovou funkci $W(C)$ pro nejméně 2 shluky − tedy pro $K \geq 2$. Často je třeba rozhodnout, zda-li není shlukování nadbytečné. Pro úplnost tedy vypočteme hodnotu statistiky $W(C)$ pro $K = 1$. Pak $C(i) = 1$, $\forall i = 1, \ldots, N$ a $\overline{\mathbf{x}}_k = \overline{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$, $k = 1, \ldots, K$ a (2.4) přejde na

$$W(C) = N \cdot \sum_{i=1}^{N} \|\mathbf{x}_i - \overline{\mathbf{x}}\|^2 = N \cdot \sum_{i=1}^{N} \sum_{j=1}^{p} (x_{ij} - \overline{x}_j)^2 =$$

$$= N(N-1) \cdot \sum_{j=1}^{p} \frac{1}{N-1} \sum_{i=1}^{N} (x_{ij} - \overline{x}_j)^2 =$$

$$= N(N-1) \cdot \sum_{j=1}^{p} \mathsf{var}\, \mathbf{X}_j, \tag{2.5}$$

kde $\mathsf{var}\, \mathbf{X}_j$ je výběrový rozptyl j-té proměnné.

Pro rozlišení označme $\mathbf{c}_k$, $k = 1, \ldots, K$ průměry, které se mění v průběhu procedury k-means a dále $\overline{\mathbf{x}}_k$, $k = 1, \ldots, K$ průměry ve shlucích po ukončení procedury k-means. Algoritmus byl poprvé implementován Lloydem (1957) v podobě jakou uvádí [Tibshirani a kol., 2001], str. 462.

ALGORITMUS k-MEANS:

0. **Inicializace:**
   Zvolíme K a počáteční hodnoty průměrů $\mathbf{c}_1, \ldots, \mathbf{c}_K$.

1. **Klasifikace:**
   Všechny objekty $\mathbf{x}_1, \ldots, \mathbf{x}_N$ přiřadíme k nejbližšímu průměru $\mathbf{c}_1, \ldots, \mathbf{c}_K$, tedy

$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{c}_k\|^2, \quad i = 1, \ldots, N.$$

2. **Učení:**
   V každém shluku vypočteme nový střed jako těžiště prvků shluku, tedy

$$\mathbf{c}_k = \frac{1}{N_k} \sum_{\{i:\, C(i)=k\}} \mathbf{x}_i, \quad k = 1, \ldots, K.$$

3. Kroky 1. a 2. opakujeme, dokud se nezmění klasifikace v kroku 1.

Snahou algoritmu k-means je tedy vyřešit rozšířenou optimalizační úlohu

$$\min_{C, \{\mathbf{c}_k\}_{k=1}^K} \left\{ \sum_{k=1}^{K} N_k \sum_{\{i:\, C(i)=k\}} \|\mathbf{x}_i - \mathbf{c}_k\|^2 \right\}. \tag{2.6}$$

Mezi výhody algoritmu k-means patří bezesporu jeho výpočetní složitost, která je $O(NKt)$, kde N je počet pozorování, K je počet shluků, t je počet iterací. Algoritmus navíc velmi rychle konverguje (zpravidla $t \ll N$) a lze jej tedy volat opakovaně pro různé počáteční volby náhodných středů.

Mezi základní nevýhody algoritmu patří nutnost zadání informace o počtu shluků K, do kterých se objekty mají rozřadit. O nemožnosti použít k-means na jiná než kvantitativní data jsme se již zmínili. Dalším problémem týkající se samotných dat může být přítomnost odlehlých pozorování, neboť ty v případě k-means významně ovlivňují výsledek shlukování. Důvodem jsou dvě skutečnosti: Jednak algoritmus používá jako metriku druhou mocninu eukleidovské vzdálenosti, která zvyšuje vliv odlehlých pozorování, a v neposlední řadě k-means počítá s aritmetickými průměry, které jsou málo citlivé vůči odlehlým pozorováním. Úloha (2.6) je úloha konvexní optimalizace, takže vzhledem k povaze algoritmu není možnost detekovat nekonvexní shluky.

Metoda k-means je naimplementována v mnoha statistických programech. My pracujeme v celé práci s prostředím `R`.

```
kmeans(x, centers = ..., iter.max = ..., nstart = ...,
       algorithm = "...")
```

`x`	matice dat, sloupce odpovídají proměnným, řádky pozorováním
`centers`	vektor inicializačních center $\mathbf{c}_1, \ldots, \mathbf{c}_K$ nebo předem zvolený počet shluků $K > 1$ − v tomto případě jsou počáteční centra volena náhodně,
`iter.max`	maximální počet iterací,
`nstart`	značí, kolikrát jsou náhodně zvolena počáteční centra (zadává se v případě, že `centers` udává počet shluků),
`algorithm`	název algoritmu podle tvůrců modifikací; viz nápověda v `R`.

Procedura `kmeans` spočte

- `size` - počet pozorování N_k v k-tém shluku, $k = 1, \ldots, K$,
- `cluster means` - středy $\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_K$ všech shluků,
- `clustering vector` - vektor délky N značící, v jakém shluku jsou jednotlivá pozorování, tedy $(C(1), C(2), \ldots, C(N))$, kde C je funkce příslušnosti (2.1),
- `within cluster sum of squares` - vnitroshlukovou vzdálenost pro k-tý shluk podle vzorce

$$WSS^{(k)} = \sum_{\{i:\, C(i)=k\}} \|\mathbf{x}_i - \overline{\mathbf{x}}_k\|^2, \quad k = 1, \ldots, K. \tag{2.7}$$

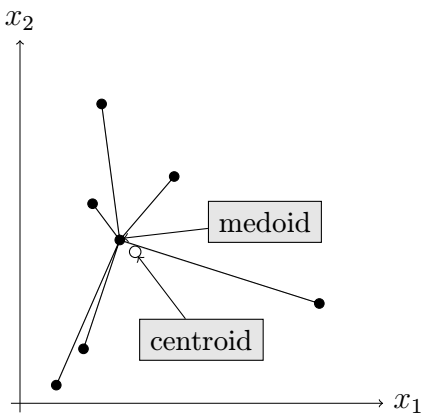
2.2 Metoda k-medoids

Jak už bylo výše řečeno, algoritmus k-means je málo robustní vůči odlehlým pozorováním, neboť počítá s průměry objektů a navíc je jeho použití omezeno jen na kvantitativní data vzhledem k nutnosti používat eukleidovsku metriku. S nástupem kategoriálních dat bylo potřeba umět shlukovat i takové objekty, a proto byl vyvinut zobecňující algoritmus, tzv. *k-medoids*. Tato metoda, která lze použít na data jakéhokoliv druhu, navíc odstraňuje oba problémy procedury k-means: Namísto eukleidovské vzdálenosti jí lze aplikovat s libovolnou metrikou d a místo center počítá s tzv. medoidy.

Definice 2.2: *Nechť $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_k}$ jsou pozorování ve shluku S_k. Pak* medoidem *tohoto shluku vzhledem k metrice $d : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ budeme rozumět pozorování $\mathbf{x}_{i^*}$, kde*

$$i^* = \underset{i=1,\ldots,N_k}{\operatorname{argmin}} \left\{ \sum_{i'=1}^{N_k} d\left(\mathbf{x}_i, \mathbf{x}_{i'}\right) \right\}.$$

Medoid shluku S_k je tedy podle definice 2.2 takové pozorování, které má nejmenší součet vzdáleností ve smyslu metriky d s ostatními body uvnitř tohoto shluku. Na obrázku 2.1 je pak znázorněn pro daný shluk centroid i medoid vzhledem k eukleidovské metrice (1.3).



Obrázek 2.1: Porovnání medoidu a centroidu u shluku v prostoru $(\mathbb{R}^2, d_2)$.

Všimněme si, že centroid nemusí být jedním z pozorování, kdežto medoid ano.

ALGORITMUS k-MEDOIDS:

0. **Inicializace:**
   Zvolíme K, vhodnou metriku d a počáteční hodnoty medoidů $\mathbf{x}_{i_1^*}, \ldots, \mathbf{x}_{i_K^*}$.

1. **Klasifikace:**
   Všechny objekty $\mathbf{x}_1, \ldots, \mathbf{x}_N$ přiřadíme k nejbližšímu medoidu, tedy

$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}}\, d\left(\mathbf{x}_i, \mathbf{x}_{i_k^*}\right), \quad i = 1, \ldots, N.$$

2. **Učení:**
   V každém shluku nalezneme nový medoid, který minimalizuje součet vzdáleností ve smyslu metriky d s ostatními body ve shluku:

$$i_k^* = \underset{\{i:\, C(i)=k\}}{\operatorname{argmin}} \left\{ \sum_{\{i':\, C(i')=k\}} d\left(\mathbf{x}_i, \mathbf{x}_{i'}\right) \right\}, \quad k = 1, \ldots, K.$$

   Pak $\mathbf{x}_{i_k^*}$, $k = 1, \ldots, K$ jsou nové medoidy.

3. Kroky 1. a 2. opakujeme, dokud se nezmění klasifikace v kroku 1.

Poznamenejme, že narozdíl od metody k-means, kde byla počáteční centra volena náhodně, existuje v případě k-medoids postup, jakým zvolit počáteční medoidy v kroku 0 algoritmu. Tato fáze se obvykle nazývá *build* a je popsána v [Kaufman

a Rousseeuw, 1990], str. 102–103. Proto se u algoritmu k-medoids nemusí zadávat počet náhodných startů, jako tomu je u k-means.

Mezi výhody algoritmu k-medoids patří již zmíněná menší citlivost na odlehlé objekty, za což se platí nepatrně větší složitostí při výpočtu medoidů.

Metoda k-medoids (PAM[1]) se v prostředí `R` nachází v balíku `cluster` a volá se příkazem

```
pam(x, k, diss = ..., metric = "...", medoids = NULL, stand = ...)
```

`x`	datová matice nebo matice vzdáleností, podle argumentu `diss`,
`k`	kladné číslo specifikující počet shluků,
`diss`	logický znak: je-li `T`, pak argument `x` se posuzuje jako matice vzdáleností, jinak `x` je považováno za matici dat,
`metric`	typ metriky použitý pro výpočet vzdáleností mezi objekty,
`medoids`	zadání vektoru počátečních medoidů (lze vynechat),
`stand`	logický znak: je-li `T`, pak se provede standardizace dat odečtením průměru $\overline{x}_j$ a vydělením odchylkou sm_j, kde

$$\overline{x}_j := \frac{1}{N} \cdot \sum_{i=1}^{N} x_{ij}, \quad j = 1, \ldots, p,$$

$$sm_j := \frac{1}{N} \cdot \sum_{i=1}^{N} |x_{ij} - \overline{x}_j|, \quad j = 1, \ldots, p.$$

Procedura `pam` spočte

- `medoids` - medoidy $\mathbf{x}_{i_1^*}, \ldots, \mathbf{x}_{i_K^*}$,
- `clustering vector` - vektor délky N značící, v jakém shluku jsou jednotlivá pozorování, tedy vektor $(C(1), C(2), \ldots, C(N))$, kde C je definována v (2.1)
- numerické informace o shluku S_k, $k = 1, \ldots, K$:
  - `size` - počet pozorování N_k,
  - `max_diss` a `av_diss` - maximální a průměrná vzdálenost objektů v k-tém shluku od medoidu daného shluku,
  - `diameter` - diametr k-tého shluku podle vzorce

$$diam_k = \max_{\{i,i':C(i)=k,C(i')=k\}} d(\mathbf{x}_i, \mathbf{x}_{i'}),$$

  - `separation` - minimální vzdálenost mezi pozorováním ve shluku S_k a pozorováním z jiného shluku podle vzorce

$$sep_k = \min_{\{i,i':C(i)=k,C(i')\neq k\}} d(\mathbf{x}_i, \mathbf{x}_{i'}),$$

[1] Partition Around Medoids; procedura navržená Kaufmanem a Rousseeuwem

- pro každé pozorování $i = 1, \ldots, N$:
  - `neighbor` - nejbližší sousední shluk,
  - hodnoty `silhouette width` $s(i)$ podle vzorce (2.9), které indikují kvalitu rozkladu; více bude o silhouette pojednáno na straně 27.
- `average silhouette width` pro k-tý shluk podle vzorce

$$sa_k = \frac{1}{N_k} \cdot \sum_{i=1}^{N} s(i) \cdot \mathbb{I}_{[\mathbf{x}_i \in S_k]}, \quad k = 1, \ldots, K,$$

- `average silhouette width` pro celý datový soubor podle vzorce

$$\frac{1}{N} \cdot \sum_{k=1}^{K} sa_k \cdot N_k = \frac{1}{N} \cdot \sum_{i=1}^{N} s(i). \tag{2.8}$$

2.3 Kritéria pro určení optimálního počtu shluků

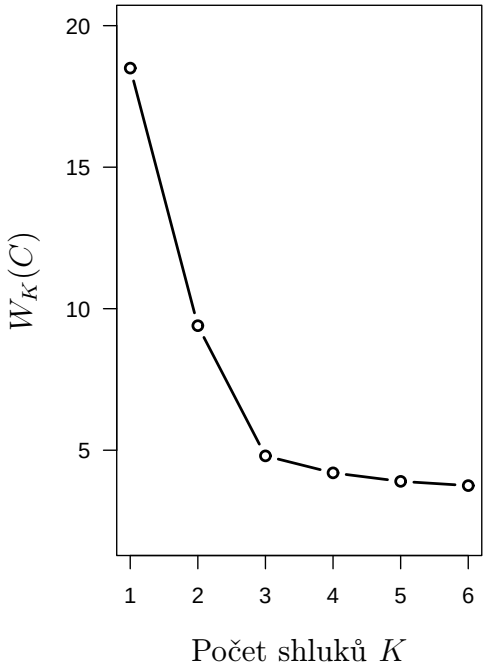
Jak již bylo zdůrazněno, nevýhodou metod k-means a k-medoids je nutnost zadat do algoritmu celkový počet shluků K. Někdy je z povahy úlohy hodnota K předem dána. Tato situace nastává například tehdy, když se firma rozhodne zaměstnat K lidí vyškolených pro plnění různých požadavků a je potřeba rozdělit okruh možných zákazníků firmy do těchto K skupin tak, aby se v každé skupině vyskytovali klienti s co možná nejbližšími požadavky. Často je však zapotřebí počet shluků K také předem odhadnout. Nyní si ukážeme přístupy, které mohou být použity k odhadu celkového počtu shluků.

2.3.1 Přístup pomocí ztrátové funkce

Následující přístup je popsán v knize [Tibshirani a kol., 2001], str. 471. Algoritmus k-means i k-medoids je založen na minimalizaci ztrátové funkce $W(C)$. Tato ztrátová funkce závisí na počtu shluků K, označme ji proto $W_K(C)$. Ztrátová funkce obyčejně s rostoucím K klesá. Nyní popíšeme subjektivní kritérium pro posouzení kvality rozkladu N objektů do K shluků. Na obrázku 2.2 je znázorněn graf závislosti součtů vzdáleností objektů ve shlucích podle vzorců (2.4) a (2.5) na počtu shluků. Všimněme si, že ztrátová funkce výrazně klesá, zvětší-li se počet shluků z jednoho na dva a ze dvou na tři. Zvětšujeme-li dále počet shluků, ztrátová funkce již významněji neklesá, a tedy další rozklady štěpí přirozeně vytvořené shluky na menší množiny, což již není žádoucí. Proto za optimální počet shluků K^* volíme ten, pro který

$$W_{K^*-1}(C) \gg W_{K^*}(C) \quad \& \quad W_{K^*}(C) \sim W_{K^*+1}(C).$$

V našem případě by tedy bylo $K^* = 3$.



Obrázek 2.2: Posouzení optimálního počtu shluků.

2.3.2 Silhouette

Jiný pohled na posouzení kvality rozkladu je publikován v [Kaufman a Rousseeuw, 1990], str. 83. Nechť máme N objektů rozřazených do K neprázdných shluků $S_1, \ldots, S_K$ pomocí funkce příslušnosti C zavedené v (2.1) a při zvolené metrice d. Pro každé $i = 1, \ldots, N$ provádíme následující postup:

1. Jestliže $\mathsf{card}\, S_{C(i)} = 1$, tedy pozorování $\mathbf{x}_i$ tvoří samostatný shluk, položíme $s(i) := 0$ a ukončíme proces výpočtu. V opačném případě provedeme kroky 2, 3, 4 a 5.

2. Vypočteme

$$a(i) := \frac{1}{\mathsf{card}\, S_{C(i)} - 1} \cdot \sum_{i'=1}^{N} \mathbb{I}_{[C(i')=C(i)]} \cdot d\left(\mathbf{x}_i, \mathbf{x}_{i'}\right).$$

   Jedná se o průměrnou vzdálenost mezi pozorováním $\mathbf{x}_i$ a všemi dalšími objekty $\mathbf{x}_{i'}$, se kterými je $\mathbf{x}_i$ ve stejném shluku.

3. Pro všechna $k = 1, \ldots, K$ taková, že $k \neq C(i)$ vypočteme

$$d(i,k) := \frac{1}{\mathsf{card}\, S_k} \cdot \sum_{\{i' \colon C(i')=k\}} d\left(\mathbf{x}_i, \mathbf{x}_{i'}\right).$$

Číslo $d(i,k)$ vyjadřuje průměrnou vzdálenost pozorování $\mathbf{x}_i$ od všech pozorování ve shluku S_k, $k = 1, \ldots, K$, $k \neq C(i)$.

4. Vypočteme

$$b(i) := \min_{\{k \colon k \neq C(i)\}} d(i,k).$$

Hodnota $b(i)$ udává průměrnou vzdálenost pozorování $\mathbf{x}_i$ od objektů v nejbližším sousedním shluku.

5. Položíme

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

Shrneme-li možné hodnoty funkce s, dostaneme:

$$s(i) := \begin{cases} \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}, & \mathsf{card}\, S_{C(i)} > 1, \\ 0, & \mathsf{card}\, S_{C(i)} = 1, \end{cases} \quad i = 1, \ldots, N. \tag{2.9}$$

Funkce s definovaná pomocí (2.9) se nazývá *siluetová funkce*. Z definice je ihned vidět, že $-1 \leq s(i) \leq 1$, pro všechna $i = 1, \ldots, N$.

Nyní rozebereme význam statistiky $s(i)$: Pozorování $\mathbf{x}_i$ s hodnotami $s(i)$ blízké jedné jsou považovány za dobře zařazené do shluku, neboť pak $b(i) \gg a(i)$ a tedy průměrná vzdálenost i-tého pozorování od ostatních objektů zařazených do stejného shluku je o mnoho menší než nejmenší průměrná vzdálenost objektu $\mathbf{x}_i$ od všech pozorování kteréhokoliv jiného shluku. Hodnoty kolem nuly vyjadřují, že $a(i)$ a $b(i)$ jsou přibližně stejné a tedy není jisté, zda by i-té pozorování mělo být zařazeno do jednoho nebo druhého shluku. Záporné hodnoty $s(i)$ indikují, že pozorování $\mathbf{x}_i$ je pravděpodobně zařazeno do špatného shluku. V [Kaufman a Rousseeuw, 1990], str. 88, je sestavena tabulka pro interpretaci hodnot $s(i)$, viz tabulka 2.1.

Hlavní klad siluetové funkce spočívá v tom, že pomáhá při interpretaci výsledků shlukové analýzy. Předpokládejme, že se data skládají z několika těsných shluků, které jsou navzájem dobře separované. Jestliže zvolíme K příliš malé, pak algoritmus sloučí některé přirozeně oddělené shluky do jednoho. Spojení těchto jinak dobře oddělených shluků vede k nárůstu vnitroshlukové variability a tedy k velkým hodnotám $a(i)$ a nízké hodnotě $s(i)$. Jestliže zvolíme K příliš velké, potom některé „přirozené“ shluky se rozpadnou na dva menší, které jsou velmi blízko u sebe, což vede k nízkým hodnotám $b(i)$ a tedy i k nízkým hodnotám $s(i)$. Z této úvahy vyplývá, že hodnoty siluetové funkce jsou nejvyšší pro „přirozenou“ volbu K.

Tabulka 2.1: Interpretace hodnot siluetové funkce.

rozmezí $s(i)$	Interpretace
0,71 až 1,00	Pozorování $\mathbf{x}_i$ se silně váže k danému shluku.
0,51 až 0,70	Pozorování $\mathbf{x}_i$ je dobře zařazeno do shluku.
0,26 až 0,50	Pozorování $\mathbf{x}_i$ se slabě váže k danému shluku.
−1,00 až 0,25	Pozorování $\mathbf{x}_i$ je pravděpodobně chybně zařazeno.

Definujme *průměrnou šířku rozkladu* (= average silhouette width) podle (2.8) jako

$$w_K := \frac{1}{N} \cdot \sum_{i=1}^{N} s(i). \tag{2.10}$$

Indexem K je vyznačena závislost průměrné šířky rozkladu na celkovém počtu shluků. Pak optimální počet shluků K^* se stanoví jako

$$K^* = \underset{\{K=2,3,\ldots,N-1\}}{\operatorname{argmax}} w_K. \tag{2.11}$$

2.3.3 Přístup pomocí indexů

Tento přístup pro měření kvality shlukování se poprvé objevuje v jednom z průkopnických článků o indexech v [Friedman a Rubin, 1967]. *Indexem* rozumíme funkci $\mathbb{N} \to \mathbb{R}$, která předem zvolenému počtu shluků přiřadí reálné číslo. Pomocí hodnot indexů lze stanovit optimální počet shluků. V nejjednodušších případech se volí počet shluků, kde příslušný index nabývá své maximální popř. minimální hodnoty. Nicméně častěji se při vyšetřování optimálního počtu shluků vizuálně vyšetřuje graf závislosti spočteného indexu na daném celkovém počtu shluků, jak je uvedeno v [Weingessel a Dimitriadou, 2002]. V takovém grafu se hledají největší, popř. nejmenší skoky.

Označme H předem stanovenou horní mez pro optimální počet shluků. Nechť i_K je hodnota daného indexu při počtu shluků K, $1 < K \leq H$. Jako rozhodovací kritéria se používají

- maximální rozdíl pro levý shluk, tj.

$$\max_{\{3 \leq K \leq H\}} \left\{ i_K - i_{K-1} \right\}, \tag{2.12}$$

- minimální/maximální hodnota druhých diferencí, tj.

$$\min_{\{3 \leq K \leq H-1\}} \Big\{ (i_{K+1} - i_K) - (i_K - i_{K-1}) \Big\}, \tag{2.13}$$

$$\max_{\{3 \leq K \leq H-1\}} \Big\{ (i_{K+1} - i_K) - (i_K - i_{K-1}) \Big\}. \tag{2.14}$$

Indexy rozdělíme podle [Weingessel a Dimitriadou, 2002] do dvou skupin podle užití statistik, které se používají k jejich výpočtu. Vedle názvu indexu bude vždy v závorce uvedeno, jaké rozhodovací kritérium se používá ke stanovení optimálního počtu shluků K.

První skupina indexů je založena na statistikách $WSS_K(C)$ (= Within Sum of Squares), $BSS_K(C)$ (= Between Sum of Squares) a TSS (= Total Sum of Squares).

Definice 2.3: *Mějme N pozorování $\mathbf{x}_1, \ldots, \mathbf{x}_N$, která byla rozřazena pomocí funkce $C : \{1, \ldots, N\} \to \{1, \ldots, K\}$ do K shluků s průměry $\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_K$. Nechť $\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$. Definujme statistiky*

$$\begin{aligned} WSS_K(C) &:= \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} \|\mathbf{x}_i - \overline{\mathbf{x}}_k\|^2, \\ TSS &:= \sum_{i=1}^{N} \|\mathbf{x}_i - \overline{\mathbf{x}}\|^2, \\ BSS_K(C) &:= TSS - WSS_K(C). \end{aligned} \tag{2.15}$$

Z definice 2.3 je zřejmé, že $WSS_K(C) = \sum_{k=1}^{K} WSS^{(k)}$, kde symbol $WSS^{(k)}$ jsme zavedli v (2.7). Všimněme si dále, že předpis právě zavedené statistiky $WSS_K(C)$ se podobá vzorci (2.4), který jsme odvozovali v souvislosti s funkcí $W(C)$. Obě funkce se liší ve vzorcích (2.4) a (2.15) pouze o člen N_k. Hlavní rozdíl obou funkcí spočívá v tom, že $W(C)$ sčítá odlišnosti jednotlivých pozorování uvnitř shluků mezi sebou, kdežto statistika $WSS_K(C)$ udává míru odlišnosti jednotlivých pozorování od příslušných center shluků. Podobné jsou obě funkce tím, že posuzují kvalitu shlukování na objektech uvnitř shluků. Oba funkcionály se tedy snažíme minimalizovat. TSS udává míru variability celého souboru dat a nezávisí vůbec na shlukovacích algoritmech.

Podobně jako u funkce $W(C)$ spočítáme i pro (2.15) vnitroshlukovou vzdálenost v případě $K = 1$. Potom $C(i) = 1$, $\forall i = 1, \ldots, N$ a

$$\begin{aligned} WSS_1 &= \sum_{i=1}^{N} \|\mathbf{x}_i - \overline{\mathbf{x}}\|^2 = TSS = \sum_{i=1}^{N}\sum_{j=1}^{p} (x_{ij} - \overline{x}_j)^2 = \\ &= (N-1) \cdot \sum_{j=1}^{p} \frac{1}{N-1} \sum_{i=1}^{N} (x_{ij} - \overline{x}_j)^2 = (N-1) \cdot \sum_{j=1}^{p} \mathsf{var}\, \mathbf{X}_j. \end{aligned} \tag{2.16}$$

Nyní už se věnujme první kategorii indexů, které jsou založeny na statistikách v definici 2.3:

- Calinski-Harabaszův index (optimální index řeší úlohu (2.13))

$$i_K := \frac{\frac{BSS_K(C)}{K-1}}{\frac{WSS_K(C)}{N-K}}, \tag{2.17}$$

- Hartiganův index (optimální index řeší úlohu (2.13))

$$i_K := \ln \frac{BSS_K(C)}{WSS_K(C)}, \tag{2.18}$$

- Ball-Hallův index (optimální index řeší úlohu (2.14))

$$i_K := \frac{WSS_K(C)}{K}. \tag{2.19}$$

Seřaďme všech N pozorování $\mathbf{x}_i$ do matice $\mathbf{X}$ typu $N \times p$, tedy

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_N^\top \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Np} \end{pmatrix}.$$

Nechť v k-tém shluku je N_k pozorování $\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_{N_k}^{(k)}$ s průměrem $\overline{\mathbf{x}}_k$, $k = 1, \dots, K$, $N = \sum_{k=1}^K N_k$. Definujme matice

$$\begin{aligned} \mathbf{W}_k &:= \sum_{i=1}^{N_k} \left(\mathbf{x}_i^{(k)} - \overline{\mathbf{x}}_k\right)\left(\mathbf{x}_i^{(k)} - \overline{\mathbf{x}}_k\right)^\top, \quad k = 1, \dots, K, \\ \mathbf{W}_K &:= \sum_{k=1}^{K} \mathbf{W}_k. \end{aligned}$$

Definujme dále matici

$$\mathbf{T} := \sum_{i=1}^{N} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^\top.$$

Povšimněme si, že matice $\mathbf{T}$ nezávisí na počtu shluků K. Udává míru celkové variability souboru pozorování. Matice $\mathbf{W}_K$ závisí na počtu shluků K prostřednictvím matic $\mathbf{W}_k$. Druhá skupina indexů je založena na výrazech s těmito maticemi:

- Scott-Symonsův index (optimální index řeší úlohu (2.12))

$$i_K := N \cdot \ln \frac{|\mathbf{T}|}{|\mathbf{W}_K|}, \tag{2.20}$$

- Edwards-Cavalliho index (optimální index řeší úlohu (2.14))

$$i_K := \mathrm{tr}\mathbf{W}_K, \tag{2.21}$$

- Friedman-Rubinův index (optimální index řeší úlohu (2.12), resp. (2.13))

$$i_K := \mathrm{tr}\left(\mathbf{W}_K^{-1} \cdot (\mathbf{T} - \mathbf{W}_K)\right), \text{ resp.} \tag{2.22}$$

$$i_K := \frac{|\mathbf{T}|}{|\mathbf{W}_K|}. \tag{2.23}$$

K právě zavedeným indexům z druhé kategorie je třeba poznamenat, že Scott-Symonsův index nelze spočítat, jestliže aspoň jedna z matic $\mathbf{T}$ a $\mathbf{W}_K$ je singulární. Friedmanův a Rubinův index nelze spočítat, je-li $\mathbf{W}_K$ singulární pro nějaké K. Rubinův index nemá smysl počítat, jestliže $\mathbf{T}$ je singulární, protože pak se rovná nule pro jakékoliv hodnoty K, pro které je $\mathbf{W}_K$ regulární.

Samotné číselné hodnoty funkcionálů nijak nevypovídají o kvalitě shlukování, protože když provedeme např. standardizaci dat (odečtení průměru a vydělení směrodatnou odchylkou) nebo změnu měřítka, můžeme dostat jiné rozklady a tedy i jiné hodnoty funkcionálů. Význam má ale porovnání hodnot jednoho funkcionálu rozkladu do určitého počtu shluků pro více shlukovacích metod. Podle nich pak lze vzájemně porovnávat kvalitu výsledného shlukování. Q. H. Nguyen navrhl v [Nguyen a Rayward-Smith, 2007] další funkcionály pro měření kvality shlukování a rozdělil je na ty funkce, které měří „vnitroshlukovou" variabilitu (intra-cluster measures) a ty, které měří vzdálenosti mezi shluky (inter-cluster measures). Zatímco hodnoty funkcionálů z první skupiny chceme minimalizovat, abychom dostávali kompaktnější shluky, hodnoty funkcí z druhé kategorie maximalizujeme, abychom vytvořené shluky co nejvíce separovali jeden od druhého.

2.4 Fuzzy shlukování

V této části se krátce zmíníme o jednom postupu shlukovacích metod, který se označuje jako *fuzzy shlukování*.

V rozdělovacích algoritmech k-means a k-medoids se každý objekt zařazuje do právě jednoho shluku, a proto se tyto metody označují jako tzv. *hard clustering* (přiřazení „natvrdo"). Fuzzy přístup namísto jednoznačného přiřazení objektů do shluků vypočítá tzv. *koeficienty příslušnosti* k těmto shlukům. Jedná se tedy o zobecnění přiřazovacích metod.

Koeficient (nebo také stupeň či míra) příslušnosti je číslo $u_{i,k}$, které musí vyhovovat podmínkám $u_{i,k} \geq 0$, $i = 1, \ldots, N$, $k = 1, \ldots, K$ a $\sum_{k=1}^{K} u_{i,k} = 1$, $i = 1, \ldots, N$, kde N je počet pozorování a K je počet shluků. Koeficient $u_{i,k}$ odpovídá pravděpodobnosti zařazení pozorování $\mathbf{x}_i$ do shluku S_k.

Hlavní výhoda fuzzy shlukování spočívá v tom, že poskytuje detailnější informaci o struktuře dat. Touto metodou lze například rozeznat pozorování, které leží zhruba uprostřed mezi vytvořenými shluky a má tedy koeficienty příslušnosti ke všem shlukům přibližně stejné. Nevýhodou fuzzy shlukování je absence reprezentativních objektů, jako jsou centra u metody k-means a medoidy u metody k-medoids.

Hlavní snahou algoritmu je spočítat hodnoty $u_{i,k}$, které se získají minimalizací účelové funkce

$$\sum_{k=1}^{K} \frac{\sum_{i=1}^{N} \sum_{i'=1}^{N} u_{i,k}^r u_{i',k}^r \, d(\mathbf{x}_i, \mathbf{x}_{i'})}{2 \cdot \sum_{i'=1}^{N} u_{i',k}^r} \tag{2.24}$$

za podmínek

$$u_{i,k} \geq 0, \quad i = 1, \ldots, N, \; k = 1, \ldots, K, \tag{2.25}$$

$$\sum_{k=1}^{K} u_{i,k} = 1, \quad i = 1, \ldots, N, \tag{2.26}$$

kde $r > 1$ je tzv. *exponent příslušnosti* (anglicky *membership exponent*). Jedná se o volitelný parametr, kterým lze ovlivnit výsledek procedury. Z numerických experimentů vyplývá, že v případě $r \to \infty$ vede řešení úlohy (2.24) za podmínek (2.25) a (2.26) k jevu zvaném *complete fuzziness* – tedy k situaci, kdy $u_{i,k} = \frac{1}{K}$ pro všechna $i = 1, \ldots, N$ a $k = 1, \ldots, K$. V takovém případě, kdy mají všechna pozorování stejné koeficienty příslušnosti ke všem shlukům, nelze výsledek shlukování interpretovat. Literatura [Kaufman a Rousseeuw, 1990] uvádí, že naopak hodnoty r blízké jedné zpomalují konvergenci algoritmu a doporučuje volit $r = 2$.

Algoritmus na minimalizaci (2.24) za podmínek (2.25) a (2.26) je založen na iterativní metodě, která se odvodí na základě výpočtu Lagrangeových multiplikátorů. Algoritmus spolu s jeho odvozením lze najít v [Kaufman a Rousseeuw, 1990], str. 182–186. Oproti nehierarchickým metodám je algoritmus výpočetně náročnější.

Na následujících řádcích uvedeme syntaxi pro použití algoritmu v praxi. V softwaru `R` se fuzzy shluková analýza volá příkazem `fanny`, který se nachází podobně jako `pam` v balíku `cluster`.

```
fanny(x, k, diss = ..., memb.exp = ..., metric = "...", stand = ...)
```

`x`	datová matice nebo matice vzdáleností, podle argumentu `diss`,
`k`	přirozené číslo specifikující počet shluků,
`diss`	logický znak: je-li `T`, pak argument `x` se posuzuje jako matice vzdáleností, jinak `x` je považováno za matici dat,
`memb.exp`	koeficient příslušnosti $r > 1$; viz nápověda v softwaru `R`.
`metric`	typ metriky použitý pro výpočet vzdáleností mezi objekty,
`stand`	logický znak: je-li `T`, pak se provede standardizace dat odečtením průměru $\overline{x}_j$ a vydělením odchylkou sm_j jako v proceduře `pam`.

Procedura `fanny` spočte

- `membership` - matice $\{u_{i,k}\}$ typu $N \times K$ udávající procentuální příslušnost jednotlivých pozorování k jednotlivým shlukům.

- `coeff` - Dunnův rozdělovací koeficient s jeho normovanou variantou podle vzorců

$$F_K := \sum_{i=1}^{N}\sum_{k=1}^{K}\frac{u_{i,k}^2}{N}$$
$$F_{\mathrm{norm},K} := \frac{F_K - \frac{1}{K}}{1-\frac{1}{K}} = \frac{KF_K - 1}{K-1}$$

  Hodnoty F_K blízké $\frac{1}{K}$ indikují jev complete fuzziness, hodnoty blízké jedné značí, že příslušnosti jednotlivých objektů ke shlukům jsou skoro stoprocentní.

- `objective` - minimální hodnota účelové funkce (2.24),

- `clustering` - vektor délky N; pro každé pozorování přiřazeno číslo shluku, pro které je $u_{i,k}$ největší, tedy

$$\left(\operatorname*{argmax}_k(u_{1,k}), \operatorname*{argmax}_k(u_{2,k}), \ldots, \operatorname*{argmax}_k(u_{N,k})\right).$$

Použití nehierarchických metod a fuzzy shlukování včetně prostředků k určení optimálního počtu shluků si ukážeme na následujícím příkladě.

2.5 Ilustrační příklad

K dispozici máme data v tabulce 2.2 o relativním zastoupení vybraných náboženství v 72 státech světa (zdroj: [Paldam, 2000]). Vysvětlivky k popisům proměnných jsou uvedeny v tabulce 2.3.

Zkusíme postupně aplikovat metody shlukové analýzy zmíněné v předchozích odstavcích. Protože všechna data jsou stejného kvantitativního charakteru v desetinných jednotkách, není třeba je standardizovat a můžeme použít k-means i k-medoids. K použití nehierarchických metod je potřeba zadat počet shluků K. Na základě vyložené teorie použijeme postupně subjektivní metody, siluetovou funkci i indexy.

Jak již bylo řečeno, procedura `kmeans` v programu `R` z balíku `stats` udává na výstupu vnitroshlukové vzdálenosti pro každý shluk. Následujícími příkazy uložíme na K-tou souřadnici vektoru **wss** součet vnitroshlukové variability pro rozklad objektů do K shluků podle vzorce (2.15) pro $K = 2, \ldots, 10$. Pro $K = 1$ ji spočteme podle vzorce (2.16).

```
wss<-vector(mode="numeric",length=10)
wss[1]<-(n-1)*sum(apply(Data1,2,var))
for (k in 2:10) {
     wss[k]<-sum(kmeans(Data1,k,iter.max=100,nstart=50)$withinss)}
```

Tabulka 2.2: Zastoupení jednotlivých náboženství.

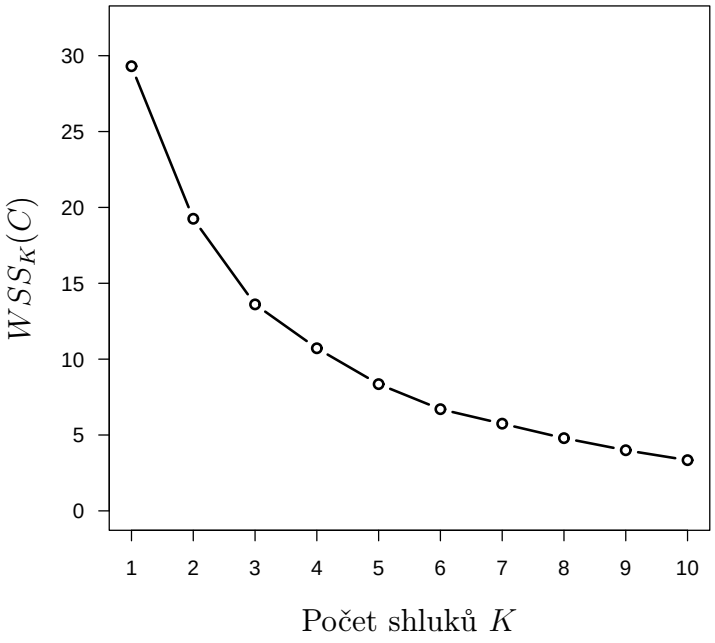
Country	Pro	Cat	Ang	Old	Isl	Bud	Chi	Hin	Tri	Ath	Res
Den	0,952	0,006	0,001	0,000	0,002	0,000	0,000	0,000	0,000	0,036	0,003
Fin	0,931	0,001	0,001	0,012	0,000	0,000	0,000	0,000	0,000	0,055	0,000
Swe	0,685	0,014	0,000	0,011	0,000	0,000	0,000	0,000	0,000	0,287	0,003
NZe	0,379	0,187	0,327	0,002	0,000	0,000	0,003	0,001	0,002	0,080	0,019
Ice	0,966	0,007	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,021	0,006
Can	0,296	0,476	0,105	0,028	0,006	0,001	0,001	0,002	0,000	0,063	0,022
Sin	0,026	0,047	0,006	0,000	0,174	0,086	0,546	0,067	0,000	0,042	0,006
Ned	0,424	0,431	0,001	0,001	0,010	0,000	0,001	0,007	0,000	0,121	0,004
Sui	0,432	0,536	0,002	0,003	0,003	0,000	0,000	0,000	0,000	0,019	0,005
Nor	0,978	0,003	0,001	0,000	0,001	0,000	0,000	0,000	0,000	0,017	0,000
Lux	0,030	0,970	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Aus	0,235	0,297	0,278	0,030	0,002	0,001	0,000	0,000	0,001	0,149	0,007
UK	0,235	0,297	0,278	0,030	0,002	0,001	0,000	0,000	0,000	0,149	0,008
Ger	0,463	0,360	0,000	0,008	0,019	0,000	0,000	0,000	0,000	0,116	0,034
Ire	0,011	0,953	0,029	0,001	0,000	0,000	0,000	0,000	0,000	0,004	0,002
HKo	0,075	0,079	0,006	0,000	0,000	0,172	0,506	0,007	0,000	0,135	0,020
Aut	0,065	0,893	0,000	0,008	0,006	0,000	0,000	0,000	0,000	0,027	0,001
USA	0,436	0,302	0,024	0,022	0,008	0,001	0,003	0,002	0,000	0,069	0,133
Chi	0,019	0,821	0,000	0,003	0,000	0,000	0,000	0,000	0,009	0,064	0,084
Isr	0,002	0,010	0,000	0,004	0,080	0,000	0,000	0,000	0,000	0,013	0,891
Por	0,011	0,942	0,001	0,000	0,000	0,000	0,000	0,000	0,000	0,046	0,000
Spa	0,001	0,969	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,029	0,001
Fra	0,024	0,768	0,000	0,008	0,030	0,001	0,002	0,000	0,000	0,156	0,011
Bot	0,268	0,094	0,022	0,000	0,000	0,000	0,000	0,000	0,492	0,000	0,124
Slo	0,000	0,718	0,000	0,000	0,010	0,000	0,000	0,000	0,000	0,043	0,229
Jap	0,009	0,006	0,002	0,000	0,000	0,596	0,224	0,000	0,000	0,120	0,043
Est	0,400	0,010	0,000	0,090	0,020	0,000	0,000	0,000	0,000	0,420	0,060
Tai	0,020	0,014	0,000	0,000	0,000	0,231	0,183	0,000	0,000	0,552	0,000
Bel	0,004	0,900	0,000	0,004	0,011	0,000	0,000	0,000	0,000	0,075	0,006
Nam	0,550	0,300	0,000	0,000	0,000	0,000	0,000	0,000	0,150	0,000	0,000
Hun	0,216	0,539	0,000	0,005	0,000	0,000	0,000	0,000	0,000	0,159	0,081
CRi	0,068	0,905	0,002	0,000	0,000	0,001	0,002	0,000	0,000	0,009	0,013
Mal	0,014	0,028	0,006	0,000	0,494	0,064	0,252	0,076	0,045	0,003	0,018
SAf	0,390	0,112	0,069	0,001	0,013	0,000	0,000	0,020	0,159	0,008	0,228
Tun	0,000	0,001	0,000	0,000	0,994	0,000	0,000	0,000	0,000	0,001	0,004
Mau	0,009	0,312	0,007	0,000	0,164	0,006	0,001	0,461	0,000	0,005	0,035
Gre	0,001	0,004	0,000	0,976	0,015	0,000	0,000	0,000	0,000	0,003	0,001
Ita	0,004	0,832	0,000	0,001	0,001	0,000	0,000	0,000	0,000	0,162	0,000
Cze	0,043	0,390	0,000	0,003	0,000	0,000	0,000	0,000	0,000	0,561	0,003
Per	0,003	0,951	0,000	0,000	0,000	0,000	0,002	0,000	0,010	0,005	0,029
Uru	0,019	0,602	0,000	0,007	0,000	0,000	0,000	0,000	0,000	0,351	0,021
Jor	0,003	0,017	0,002	0,019	0,930	0,000	0,000	0,000	0,000	0,020	0,009
Mon	0,000	0,000	0,000	0,000	0,040	0,960	0,000	0,000	0,000	0,000	0,000
Pol	0,001	0,812	0,000	0,012	0,000	0,000	0,000	0,000	0,000	0,095	0,080
Mlw	0,315	0,022	0,029	0,003	0,162	0,000	0,000	0,001	0,190	0,000	0,278
Bra	0,040	0,878	0,000	0,001	0,001	0,003	0,002	0,000	0,000	0,014	0,061
Mor	0,000	0,002	0,000	0,000	0,994	0,000	0,000	0,000	0,000	0,000	0,004
Zim	0,214	0,156	0,049	0,002	0,009	0,000	0,000	0,000	0,405	0,002	0,163
Sal	0,028	0,962	0,000	0,000	0,000	0,000	0,000	0,000	0,002	0,003	0,005
Lit	0,050	0,800	0,000	0,090	0,000	0,000	0,000	0,000	0,000	0,060	0,000
Cot	0,053	0,208	0,000	0,000	0,387	0,000	0,000	0,000	0,170	0,134	0,048
Bol	0,023	0,925	0,000	0,000	0,000	0,000	0,000	0,000	0,011	0,014	0,027
Arm	0,000	0,000	0,000	0,940	0,000	0,000	0,000	0,000	0,000	0,000	0,060
Rus	0,009	0,000	0,000	0,163	0,100	0,000	0,000	0,000	0,000	0,724	0,004
Ecu	0,019	0,964	0,000	0,000	0,000	0,000	0,000	0,000	0,006	0,006	0,005
Geo	0,000	0,000	0,000	0,750	0,110	0,000	0,000	0,000	0,000	0,140	0,000
Alb	0,000	0,100	0,000	0,200	0,700	0,000	0,000	0,000	0,000	0,000	0,000
Ban	0,000	0,000	0,000	0,000	0,883	0,000	0,000	0,105	0,000	0,000	0,012
Kaz	0,020	0,000	0,000	0,440	0,470	0,000	0,000	0,000	0,000	0,070	0,000
Pak	0,008	0,005	0,000	0,000	0,968	0,000	0,000	0,013	0,000	0,000	0,006
Kyr	0,000	0,000	0,000	0,200	0,750	0,000	0,000	0,000	0,000	0,050	0,000
Uga	0,019	0,496	0,262	0,001	0,066	0,000	0,000	0,000	0,126	0,000	0,030
Par	0,019	0,960	0,001	0,002	0,000	0,001	0,001	0,000	0,008	0,005	0,003
Ken	0,193	0,264	0,072	0,025	0,060	0,000	0,000	0,009	0,189	0,000	0,188
You	0,010	0,040	0,000	0,650	0,190	0,000	0,000	0,000	0,000	0,110	0,000
Tan	0,112	0,282	0,040	0,001	0,325	0,000	0,000	0,001	0,228	0,002	0,009
Uzb	0,000	0,000	0,000	0,090	0,880	0,000	0,000	0,000	0,000	0,000	0,030
Hon	0,026	0,958	0,000	0,000	0,001	0,000	0,000	0,000	0,001	0,004	0,010
Aze	0,000	0,000	0,000	0,048	0,934	0,000	0,000	0,000	0,000	0,000	0,018
Ino	0,048	0,027	0,000	0,000	0,434	0,010	0,362	0,021	0,047	0,016	0,035
Nig	0,158	0,121	0,105	0,000	0,450	0,000	0,000	0,000	0,056	0,002	0,108
Cam	0,181	0,350	0,000	0,000	0,220	0,000	0,000	0,000	0,216	0,001	0,032

Tabulka 2.3: Uvažovaná náboženství.

zkratka	náboženství
`Pro`	protestantské
`Cat`	katolické
`Ang`	anglikánské
`Old`	staré (např. pravoslavné)
`Isl`	islám

zkratka	náboženství
`Bud`	buddhismus
`Chi`	čínské
`Hin`	hinduismus
`Tri`	domorodé
`Ath`	bez vyznání
`Res`	jiné

Při výpočtu WSS_K jsme zvolili horní hranici $K = 10$, protože předpokládáme, že shlukování proběhne podle nízkého počtu dominantních náboženství. Rozklady do vysokého počtu shluků jsou navíc obtížně interpretovatelné. Na obrázku 2.3 vidíme závislost statistiky WSS na počtu shluků. Hodnoty $WSS_K(C)$ klesají přibližně kvadraticky, takže vizuálně lze těžko stanovit optimální počet shluků. Nepatrně větší změny v poklesu jsou vidět v bodech $K = 2$ a $K = 3$.



Obrázek 2.3: Závislost statistiky WSS na počtu shluků v příkladě náboženství.

Zkusíme tedy rozhodnout o optimálním celkovém počtu shluků K pomocí indexů. V softwaru `R` je k dispozici procedura `clustIndex` z balíku `cclust`, která

vypočte výše uvedené indexy pro zadaný rozklad do příslušného počtu shluků. Volání `clustIndex` je podmíněno zavoláním procedury `cclust`, která provede vlastní shlukování. Argumenty funkce `cclust` jsou datová matice, volba metody (k-means, aj.), předem zvolený počet shluků K nebo vektory inicializačních center. Často však uživatel nemá představu o prostorovém rozložení dat − zvláště, když se jedná o více jak třírozměrné vektory pozorování. Proto se zadává namísto počátečních center spíše požadovaný počet shluků. Pak ale `cclust` volí počáteční centra náhodně a tedy pokaždé jinak, takže po opětovném zavolání této procedury můžeme dostat jiné rozřazení objektů do shluků, a tedy po zavolání `clustIndex` i jiné hodnoty indexů. U `cclust` totiž nelze nastavit opakované inicializace jako u k-means z balíku `stats`. Tato nepříjemnost nastává zvláště tehdy, máme-li shlukovat velké množství dat. Proto funkce, které počítají hodnoty indexů, naprogramujeme mechanicky na základě procedury `kmeans`. Procedura `kmeans` v základním balíku `stats` již problém náhodných inicializací řeší zadáním parametru `nstart`, který provede vícekrát náhodný výběr inicializačních center a ze všech voleb vybere nejlepší výsledek shlukování ve smyslu minimální hodnoty WSS. Dostatečným počtem inicializací tedy zvětšujeme pravděpodobnost, že výsledky procedury budou po každém jejím zavolání stejné.

Vypočteme hodnoty indexů (2.17), (2.18) a (2.19) z první skupiny

```
x.prum<-1/n*apply(Data,2,sum)
TSS<-0
for (i in 1:n) {
     TSS<-TSS+sum( (Data[i,]-x.prum)^2)}
for (k in 2:10) {
     rozklad<-kmeans(Data,k,nstart=50)
     WSS<-sum(rozklad$withinss)
     BSS<-TSS-WSS
     Cal.ind[k]<-(BSS/(k-1))/(WSS/(n-k))
     Har.ind[k]<-log(BSS/WSS)
     Bal.ind[k]<-WSS/k}
```

a na nalezené indexy použijeme optimalizační kritéria (2.12), (2.13) a (2.14) podle typu indexu, viz strana 28.

Při výpočtu indexů z 2. kategorie se objeví problém se singularitou matice $\mathbf{T}$, neboť $\mathsf{rank}(\mathbf{T}) = 10 < 11$. Potom tedy $|\mathbf{T}| = 0$, a proto výraz pro Scottův index (2.20) nemá smysl. Friedmannův index (2.22) ani Rubinův index (2.23) rovněž v našem případě nelze vypočítat, protože pro všechna uvažovaná $K = 2, 3, \ldots, 10$ vyšlo rovněž $\mathsf{rank}(\mathbf{W}_K) = 10 < 11$ a tedy ani $\mathbf{W}_K^{-1}$ neexistuje. Lze vypočítat pouze Edwardsův index (2.21), který nepracuje ani s determinantem matice $\mathbf{W}_K$ ve jmenovateli ani s inverzní maticí $\mathbf{W}_K^{-1}$.

```
T<-matrix(0,nrow=p,ncol=p)
for (i in 1:n) {
     T<-T+(Data[i,]-x.prum)%*%t(Data[i,]-x.prum)}
```

```
for (k in 2:10) {
     r<-kmeans(Data,k,nstart=50)
     W<-matrix(0,nrow=p,ncol=p)
     for (l in 1:k) {
     for (i in 1:n) {
         if (r$cluster[i]==l) {
     W<-W+(Data[i,]-r$centers[l,])%*%t(Data[i,]-r$centers[l,])}}}
     for (j in 1:p) {
          Edw.ind[k]<-Edw.ind[k]+W[j,j]}}
```

Optimální počty shluků pro jednotlivé indexy nalezené na základě kritérií ze strany 28 uvádí tabulka 2.4. Vidíme, že v případě procedury k-means bychom nejspíše zvolili 3 shluky.

Tabulka 2.4: Optimální počet shluků na základě indexů.

Index	Calinski	Hartigan	Ball	Edwards
opt. počet shluků	3	3	3	3

Nyní provedeme vlastní shlukování objektů do třech shluků pomocí procedury `kmeans` v programu `R`. Dostaneme $N_1 = 36$, $N_2 = 23$, $N_3 = 13$ a průměry shluků uvedené v tabulce 2.5.

Tabulka 2.5: Průměrná zastoupení náboženství v jednotlivých shlucích procedury k-means v %.

shluk	Pro	Cat	Ang	Old	Isl	Bud	Chi	Hin	Tri	Ath	Res
1	28,6	15,6	3,7	10,4	5,9	5,7	4,1	1,6	6,1	11,5	6,8
2	3,0	84,9	1,3	0,6	0,5	0,0	0,0	0,0	0,8	5,8	3,0
3	1,9	2,3	0,9	7,7	76,0	0,6	4,7	1,7	1,1	1,2	1,9

Vnitroshlukové vzdálenosti ve shlucích jsou $WSS^{(1)} \doteq 11{,}76$, $WSS^{(2)} \doteq 0{,}78$, $WSS^{(3)} \doteq 1{,}07$. Výsledek shlukování metodou k-means je v tabulce 2.6.

Tabulka 2.5 uvádí průměrná procentuální zastoupení náboženství v jednotlivých shlucích. Vidíme v ní, že došlo k rozřazení států podle třech velmi vlivných náboženství světa – protestantského, katolického a islámu. V 1. shluku jsou soustředěny státy s častým zastoupením protestantského a katolického náboženství. Typickým příkladem může být Dánsko nebo Velká Británie. V 2. shluku převažují katolické státy a třetí shluk obsahuje státy s výrazným zastoupením islámu. Povšimněme si, že zařazení některých států nekoresponduje s jejich náboženskou skladbou. Např. v Arménii, která byla zařazena do 1. shluku, má téměř výhradní zastoupení staré náboženství. To tvoří průměrně v 1. shluku až třetí nejsilnější kategorii, kdežto mezi

Tabulka 2.6: Výsledek shlukování (metoda: k-means).

1. shluk
Arménie, Austrálie, Botswana, Česká republika, Dánsko, Estonsko, Finsko, Gruzie, Hong-Kong, Island, Izrael, Japonsko, Jižní Afrika, Jugoslávie, Kamerun, Kanada, Keňa, Malawi, Mauricius, Mongolsko, Namíbie, Německo, Nizozemsko, Norsko, Nový Zéland, Pobřeží Slonoviny, Rusko, Řecko, Singapur, Švédsko, Švýcarsko, Taiwan, Tanzánie, USA, Velká Británie, Zimbabwe,
2. shluk
Belgie, Bolívie, Brazílie, Costa Rica, Ekvádor, Francie, Honduras, Chile, Irsko, Itálie, Lotyšsko, Lucembursko, Maďarsko, Paraguay, Peru, Polsko, Portugalsko, Rakousko, Salvádor, Slovinsko, Španělsko, Uganda, Uruguay,
3. shluk
Albánie, Ázerbajdžán, Bangladéš, Indonésie, Jordánsko, Kazachstán, Kyrgystán, Malajsie, Maroko, Nigérie, Pákistán, Tunisko, Uzbekistán.

státy ve 3. shluku je staré náboženství průměrně druhé nejvíce zastoupené. Důvodem, proč byla Arménie zařazena právě do 1. shluku a ne do 3. shluku, může být fakt, že ve 3. shluku dominují státy s výrazným zastoupením islámu, které v Arménii není zastoupeno vůbec. Obecně lze říci, že 1. shluk „pohltil“ i státy, které mají výrazně zastoupená i jiná náboženství (např. v případě buddhismu je to Hong-Kong, Japonsko nebo Mongolsko). Tento fakt je ostatně vidět z tabulky 2.5, kde 1. shluk má v průměru jako jediný zastoupený všechna náboženství výrazněji než jen 1 %. Tím, že 1. shluk obsahuje i státy s výrazným zastoupením jiných náboženství než jen katolické a protestantské, lze vysvětlit vysokou hodnotu $WSS^{(1)}$ oproti $WSS^{(2)}$ a $WSS^{(3)}$. To možná nasvědčuje tomu, že bychom měli provést rozklad do většího počtu shluků.

Zkusíme ještě použít kritérium (2.11). Spočítáme postupně hodnoty průměrné šířky rozkladu (2.10) pro rozklady do dvou, třech, až deseti shluků a zjišťujeme, že hodnoty w_K s rostoucím K také rostou. Při rozkladu do více než sedmi shluků již existuje shluk obsahující jen 2 pozorování a při rozkladu do více než devíti shluků existuje dokonce jednoprvkový shluk. Shlukování do vyššího počtu shluků tedy některé shluky očistí od pozorování, které se k danému shluku váží jen slabě, na druhou stranu není výhodné mít příliš mnoho nízkoprvkových segmentů, neboť takový rozklad pak působí nevyváženě. Nicméně jsme se rozhodli provést shlukování pomocí k-means ještě do šesti shluků. Podrobné výsledky pro 6 shluků uvádět nebudeme (učiníme to až v případě metody k-medoids), nicméně alespoň naznačíme, jak se změnily vytvořené rozklady oproti rozkladu do třech segmentů, viz tabulka 2.7.

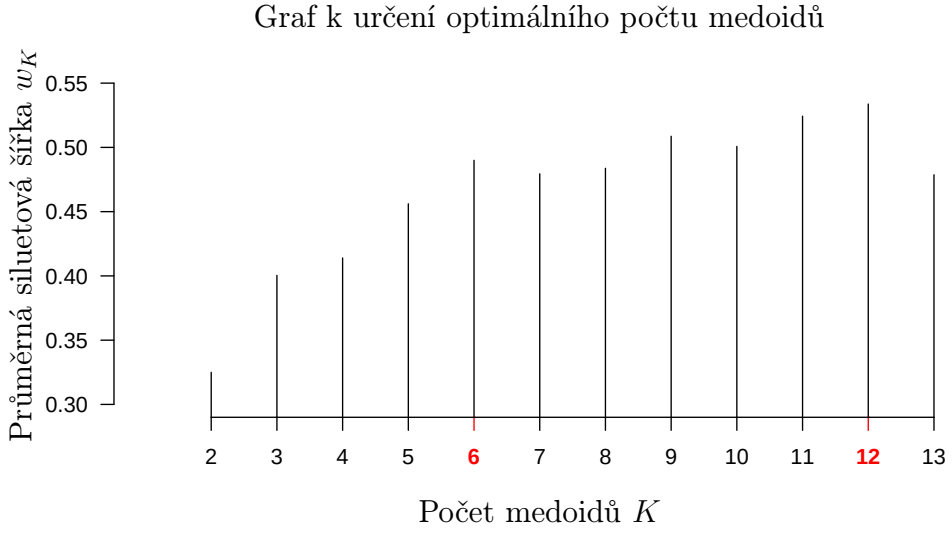
Z porovnání rozkladů do třech a šesti shluků zjistíme, že shluk z tabulky 2.6

Tabulka 2.7: Znázornění vlivu změny počtu shluků K v metodě k-means.

č. shluku	**1**	**2**	**3**	**4**	**5**	**6**	$\sum$
1	21	0	0	5	4	6	36
2	2	21	0	0	0	0	23
3	0	0	13	0	0	0	13
$\sum$	23	21	13	5	4	6	72

označený jako třetí obsahující 13 států s výrazným zastoupením islámu zůstal nezměněný i tehdy, když se data shlukovala do šesti shluků. Z 2. shluku se při rozkladu do šesti shluků oddělila Uganda a Maďarsko, které přešly do 1. shluku. Jinak však tento shluk s výrazným zastoupením katolického náboženství zůstal zachován. Z tabulky 2.7 je patrné, že zvětšení počtu shluků K ovlivnilo zejména podobu 1. shluku z tabulky 2.6. Tento shluk se při rozkladu do šesti segmentů rozštěpil a z jeho části se vytvořily další 3 shluky. Podobný výsledek však bylo možné očekávat, neboť při rozkladu do třech shluků jsme se již zmínili o vysoké hodnotě vnitroshlukové variability u 1. shluku.

V případě použití procedury k-medoids s eukleidovskou metrikou rozhodneme o optimálním počtu medoidů využitím siluetové funkce, resp. kritéria (2.11). Dostáváme výsledek dostáváme na obrázku 2.4.



Obrázek 2.4: Závislost statistiky w_K na počtu medoidů v příkladě náboženství.

V softwaru `R` jsme použili příkazy

```
onm<-numeric(70) #optimal number of medoids
```

```
for (k in 2:71) {
    onm[k]<-pam(Data,k,diss=F,metric="euclidean")$silinfo$avg.width}
k.best<-which.max(onm)
```

Vyjde $K^* = 12$, přičemž Izrael tvoří samostatný shluk. Tento rozklad se nám zdál příliš jemný a obtížně interpretovatelný, a proto jsme rozdělili data jen do šesti shluků. Na obrázku 2.4 je vidět, že 6 shluků může být považováno za optimální počet pro maximální počet 8 shluků. Výsledek shlukování uvádí tabulka 2.8. Na prvním místě je tučně vyznačen reprezentativní objekt každého shluku (medoid).

Tabulka 2.8: Výsledek shlukování (metoda: PAM, Kaufman-Rousseeuw).

1. shluk
Dánsko, Island, Finsko, Norsko, Švédsko
2. shluk
Taiwan, Česká republika, Estonsko, Hong-Kong, Japonsko, Mongolsko, Rusko, Singapur
3. shluk
Bolívie, Belgie, Brazílie, Costa Rica, Ekvádor, Francie, Honduras, Chile, Irsko, Itálie, Lotyšsko, Lucembursko, Paraguay, Peru, Polsko, Portugalsko, Rakousko, Salvádor, Slovinsko, Španělsko, Uruguay
4. shluk
Gruzie, Arménie, Jugoslávie, Kazachstán, Řecko
5. shluk
Keňa, Austrálie, Botswana, Izrael, Jižní Afrika, Kamerun, Kanada, Maďarsko, Malawi, Mauritius, Namíbie, Německo, Nigérie, Nizozemsko, Nový Zéland, Pobřeží Slonoviny, Švýcarsko, Tanzánie, Uganda, USA, Velká Británie, Zimbabwe
6. shluk
Ázerbajdžán, Albánie, Bangladéš, Indonésie, Jordánsko, Kyrgystán, Malajsie, Maroko, Pákistán, Tunisko, Uzbekistán.

Po provedení procedury `pam` vidíme, že ve stejném shluku se například ocitly státy severní Evropy Dánsko, Finsko, Norsko, Švédsko a Island se silně protestantským obyvatelstvem. Tyto státy tvoří jeden samostatný shluk i v případě `kmeans`, jestliže předem zadáme 6 shluků. V dalším shluku se ocitají státy středního východu a východní Evropy Gruzie, Arménie, Řecko, Jugoslávie a Kazachstán s výraznou převahou starého náboženství. Další shluk čítající 8 států se vyznačuje tím, že jsou výrazně zastoupeny dva postoje k náboženství. V případě České republiky jsou to ateisté a křesťané, v případě Taiwanu ateisté a buddhisté. Dále se zde vyskytují státy Japonsko (buddhismus a čínské náboženství), Rusko (ateismus a pravoslaví)

a Hong-Kong (čínské, buddhisté). Tabulka 2.9 uvádí průměrná procentuální zastoupení náboženství v jednotlivých shlucích.

Tabulka 2.9: Průměrná zastoupení náboženství v jednotlivých shlucích procedury k-medoids v %.

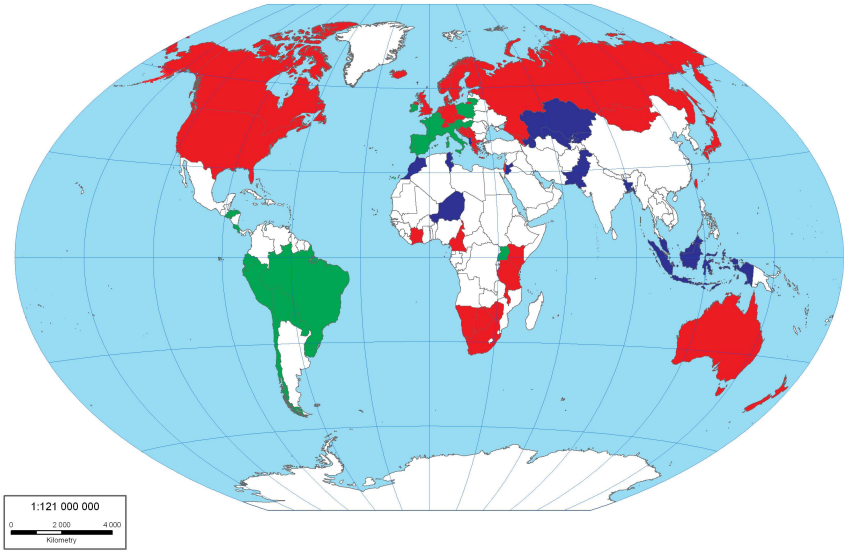
shluk	Pro	Cat	Ang	Old	Isl	Bud	Chi	Hin	Tri	Ath	Res
1	90,2	0,6	0,1	0,5	0,1	0,0	0,0	0,0	0,0	8,3	0,2
2	7,3	6,8	0,2	3,2	4,2	25,6	18,2	0,9	0,0	31,9	1,7
3	2,2	88,0	0,2	0,7	0,3	0,0	0,0	0,0	0,2	5,6	2,8
4	0,6	0,9	0,0	75,1	15,7	0,0	0,0	0,0	0,0	6,5	1,2
5	25,4	28,0	7,6	0,8	9,0	0,0	0,0	2,3	10,8	5,0	11,1
6	0,7	1,6	0,1	5,1	81,5	0,7	5,6	2,0	0,8	0,8	1,2

Na obrázcích 2.5 a 2.6 je shlukování graficky znázorněno na mapě světa. Porovnáním těchto obrázků si můžeme všimnout, jak se z většího červeného shluku na obrázku 2.5 (označený v tabulce 2.6 jako první) oddělilo 8 států s průměrnou převahou ateistů (na obrázku 2.6 jsou vyznačeny žlutou barvou a v tabulce 2.8 tvoří 2. shluk).

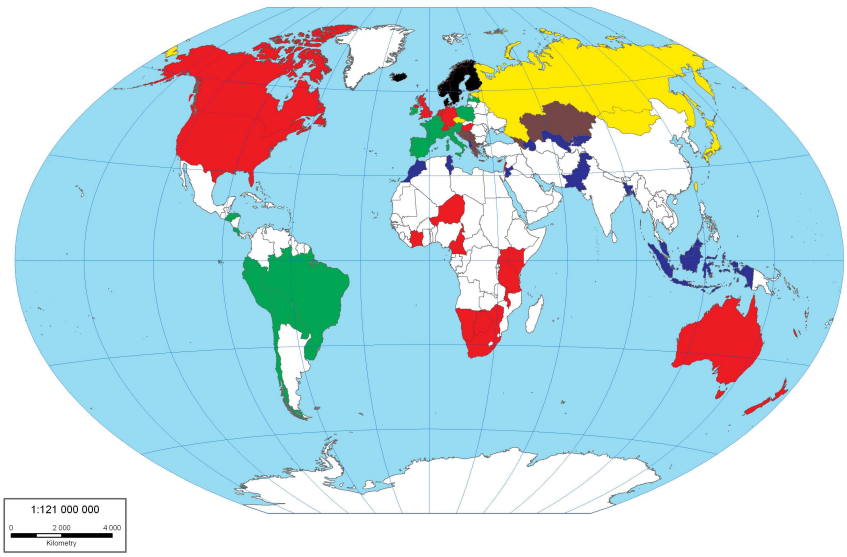
Shluky vyznačené zelenou barvou (v tabulce 2.6 je zelený shluk označený jako druhý a v tabulce 2.8 označený jako třetí) zobrazující převážně státy se silně katolickým obyvatelstvem zůstaly stejné až na Maďarsko a Ugandu, které se na obrázku 2.6 zabarvily červeně. Tyto státy mají sice nejsilněji zastoupené katolické náboženství, ale v Maďarsku je významné také protestantské náboženství a v Ugandě anglikánské. V případě metody k-medoids jsou v červeném shluku (označený v tabulce 2.8 jako pátý) průměrně zhruba stejně zastoupeny jak katolické tak protestantské náboženství.

Konečně z modrého shluku (v tabulce 2.6 označen jako třetí a v tabulce 2.8 označen jako šestý) se oddělily 2 státy: Kazachstán přešel do hnědého shluku (označený v tabulce 2.8 jako čtvrtý), neboť je v tomto státě výrazně zastoupeno kromě islámu také pravoslavné náboženství. Nigérie přešla do pátého (červeného) shluku, protože v ní sice převažuje islám, ale významně jsou zde zastoupeny také přibližně stejnou měrou katolické a protestantské náboženství.

Pro úplnost jsme provedli ještě fuzzy shlukování s rozkladem do 3 shluků. Podle doporučení v [Kaufman a Rousseeuw, 1990] jsme volili exponent příslušnosti $r = 2$. Výslednou matici rozměrů 72×3 uvádět nebudeme, ale podle výsledku algoritmu jsou nejsilnější koeficienty příslušnosti u pozorování zařazených do 2. shluku, který obsahuje státy s výrazným zastoupením katolického náboženství. Žádné pozorování nemá příslušnost k 1. shluku nebo k 3. shluku větší než 60 %. Výsledek shlukování programem `fanny` je dosti podobný rozkladu do třech shluků procedurou k-means, jak je vidět z tabulky 2.10.



Obrázek 2.5: k-means, 3 shluky, eukleidovská metrika.



Obrázek 2.6: k-medoids, 6 shluků, eukleidovská metrika.

Tabulka 2.10: Porovnání rozkladu do třech shluků u metody k-means a fuzzy shlukování.

		fanny			
	č. shluku	**1**	**2**	**3**	$\sum$
kmeans	**1**	29	0	7	36
	2	2	21	0	23
	3	0	0	13	13
	$\sum$	31	21	20	72

V 3. shluku převažují islámské státy. 1. shluk má pak dosti podobné složení, jako 1. shluk v případě rozkladu do třech shluků procedurou k-means.

Kapitola 3

Hierarchické postupy

V této kapitole se budeme zabývat algoritmy hierarchického shlukování a podobně jako v kapitole 2 zmíníme funkci, která měří kvalitu těchto metod. V závěru pak srovnáme veškeré popsané metody pomocí některých funkcionálů kvality rozkladu.

Jak napovídá už samotný název, výsledkem hierarchických shlukovacích metod je tzv. *hierarchická reprezentace*, kde shluk v každé úrovni vzniká spojením shluků o úroveň níže. Zatímco na nejnižší úrovni se jedná o triviální shluky (tj. jednoprvkové shluky obsahující původní pozorování), na nejvyšší úrovni existuje pouze jeden shluk obsahující všechny objekty. Nehierarchické metody shlukové analýzy závisely na volbě počtu shluků a inicializačních centrech, kdežto hierarchické metody tyto specifikace nepotřebují. Namísto toho pracují s maticí nepodobnosti $\mathbf{D}$ mezi shluky. V případě triviálních shluků má matice nepodobnosti tvar

$$\mathbf{D} = \begin{pmatrix} 0 & d(\mathbf{x}_1, \mathbf{x}_2) & \cdots & d(\mathbf{x}_1, \mathbf{x}_N) \\ d(\mathbf{x}_2, \mathbf{x}_1) & 0 & \cdots & d(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ d(\mathbf{x}_N, \mathbf{x}_1) & d(\mathbf{x}_N, \mathbf{x}_2) & \cdots & 0 \end{pmatrix},$$

tudíž její velikost je $N \times N$. V operační paměti počítače lze samozřejmě uchovávat jenom prvky nad horní diagonálou, neboť matice $\mathbf{D}$ je symetrická. Ani tak se ale nedoporučuje hierarchické metody používat pro velké datové soubory.

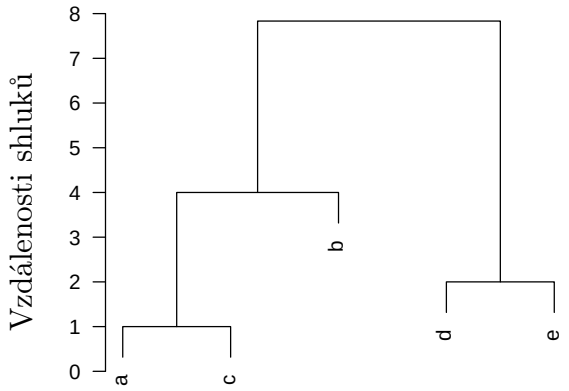
3.1 Algoritmy hierarchické shlukové analýzy

Hierarchické *aglomerativní* postupy konstruují shluky tak, že v každé iteraci spojují dva nejvíce si odpovídající shluky (metoda tzv. merging), přičemž na začátku má každé pozorování svůj vlastní shluk. Aglomerativním metodám se říká „bottom-up“ metody.

Divizivní metody postupují opačným způsobem. Z jednoho shluku všech dat se postupně vytvoří dva menší a to způsobem, že dva nové shluky mají největší mezishlukovou vzdálenost. Dalším postupem se shluky rozpadají až na jednotlivé

objekty. Divizivní algoritmy tedy nejprve pracují s menším počtem shluků a lépe tak vystihují strukturu dat. Vzhledem k tomu, že aglomerativní metody na začátku shlukují detailní objekty a postupují k menšímu počtu shluků, je zde větší pravděpodobnost, že velké shluky jsou ovlivněny náhodnými chybami v prvních krocích algoritmu. Naproti tomu zásadní nevýhodou divizivních algoritmů je početní náročnost: Počet způsobů rozdělení N pozorování do 2 neprázdných shluků je totiž $2^{N-1} - 1$, což je exponenciální složitost. Kdežto v 1. kroku aglomerativního postupu je počet způsobů všech možných shluknutí 2 objektů jen $N(N-1)/2$, tedy kvadratická složitost. Proto dosáhly větší obliby v praktickém použití aglomerativní techniky, kterými se v této práci budeme věnovat. Pro bližší seznámení s divizivními technikami hierarchického shlukování doporučujeme literaturu [Kaufman a Rousseeuw, 1990].

Jak aglomerativní tak divizivní metody mají $N - 1$ iterací a jejich průběh lze znázornit na tzv. *dendrogramu*, viz obrázek 3.1.



Obrázek 3.1: Dendrogram pro příklad 3.1.

Svislé linky reprezentují mezishlukové vzdálenosti mezi dvěma shluky. Čím delší je linka, tím více lze na základě zvolené metriky označit shluky jako rozdílné. Jednotlivé větve se rozdělují na menší a „nesou“ na sobě všechny menší shluky, které byly postupně spojeny.

V následujícím výčtu jsou uvedeny nejpoužívanější aglomerativní postupy spolu s nevýhodami spjatými s jejich použitím:

1. Nearest Neighbor (= Metoda nejbližšího souseda)
   Mezishluková vzdálenost se definuje jako vzdálenost nejbližších dvou prvků

z každého z uvažovaných shluků podle vzorce jednoduchého spojení (viz single linkage, vzorec (1.10) na straně 16). Hlavní slabina algoritmu jednoduchého spojení spočívá v tom, že stačí, aby dva zřetelně oddělené shluky měly k sobě blízko jedinou dvojicí pozorování a už je tato metoda nedokáže udržet oddělené. Při aplikaci tohoto algoritmu se tedy často i značně vzdálené objekty mohou brzy sejít v jediném shluku. Tento fenomén se nazývá *chaining* (= řetězení). Výsledné shluky pak často nesplňují přirozený požadavek spočívající v tom, že pozorování uvnitř shluku mají menší míru odlišnosti v porovnání s pozorováními mezi těmito shluky. Jestliže definujeme diametr shluku G jako

$$D_G := \max_{\substack{\mathbf{x} \in G \\ \mathbf{y} \in G}} d(\mathbf{x}, \mathbf{y}),$$

pak výsledkem metody Nearest Neighbor mohou být shluky s velkými diametry.

2. Furthest Neighbor (= Metoda nejvzdálenějšího souseda)
   Spojení shluků v tomto případě probíhá podle vzorce (1.11). Jestliže vzniknou shluky G a H metodou Furthest Neighbor, pak nastává opačný extrém než v případě metody nejbližšího souseda. G a H jsou totiž podle metody blízké pouze tehdy, jestliže všechna pozorování uvnitř těchto shluků jsou relativně podobné. Vznikají tak shluky s velmi malými diametry.

3. Group Average
   Tato metoda reprezentuje kompromis mezi výše uvedenými metodami Nearest a Furthest Neighbor. Daří se konstruovat shluky, které jsou „relativně kompaktní“ a poměrně vzdálené od ostatních. Nevýhodou je pouze závislost metody na měřítku, od kterého se odvíjí velikost míry nepodobnosti mezi pozorováními. Mezishluková vzdálenost se počítá podle (1.12).

4. Centroidní metoda
   Výpočet probíhá podle vzorce (1.13). Nevýhoda této metody spočívá v tom, že nepodobnosti mezi shluky již nejsou monotónní a to ani v případě, kdy se použije eukleidovská metrika. Je tedy často obtížné interpretovat správně dosažené výsledky.

Popíšeme nyní podrobněji algoritmus aglomerativního shlukování, který se v praxi používá nejčastěji:

AGLOMERATIVNÍ POSTUPY:

0. **Inicializace:**
   Vytvoříme N jednoprvkových shluků.

1. **Kritérium podobnosti:**
   Najdeme nejpodobnější dvojici shluků ve smyslu (SL), (CL) atd. a spojíme je do jediného shluku.

2. Vypočteme vzdálenost mezi nově vzniklým shlukem a ostatními shluky ve smyslu (SL), (CL), atd.

3. Zopakujme kroky 1. a 2., dokud nevznikne 1 shluk.

Pro výše popsaný algoritmus existují v softwaru `R` dva programy – `hclust` ze základního balíku `stats` a procedura `agnes` z balíku `cluster`. Obě funkce počítají prakticky úplně totéž, `hclust` se hodí více pro následné grafické znázornění výsledku, `agnes` má tu výhodu, že vypočítá i koeficient kvality rozkladu.

```
hclust(d, method = "...")
```

`d`	matice vzdáleností vypočtená na základě funkce `dist`,
`method`	metoda použitá k určení vzdálenosti mezi shluky; lze `ward`, `single`, `complete`, `average`, `mcquitty`, `median` nebo `centroid`.

`hclust` umožňuje nastavit i další parametry, jejich specifikace je uvedena v nápovědě programu `R`. Výstup z procedury `hclust` je podobný výstupu z procedury `agnes`, proto uvedeme až tento.

```
agnes(x, diss = ..., metric = "...", stand = ..., method = "...")
```

`x`	datová matice nebo matice vzdáleností, podle argumentu `diss`,
`diss`	logický znak: je-li `T`, pak argument `x` se posuzuje jako matice vzdáleností, jinak `x` je považováno za matici dat,
`metric`	typ metriky použitý pro výpočet vzdáleností mezi objekty, lze jen `euclidean` nebo `manhattan`,
`stand`	logický znak: je-li `T`, pak se provede standardizace dat odečtením průměru $\overline{x}_j$ a vydělením odchylkou sm_j, jako u procedury `pam`,
`method`	metoda použitá k určení vzdálenosti mezi shluky, lze `single` (single linkage), `complete` (complete linkage), `ward` (Ward's method), `weighted` (weighted average linkage); jako defaultní je nastaveno `average` (average linkage).

Procedura `agnes` spočte

- `ac` - aglomerativní koeficient, který udává kvalitu rozkladu – více bude o něm pojednáno v následující části,
- `order` - takové pořadí pozorování, aby se větve dendrogramu vzájemně nekřížily,
- `merge` - matice typu $(N-1) \times 2$, každý řádek reprezentuje krok spojení 2 shluků; jestliže pro obě čísla s, t v k-tém řádku matice `merge` platí
  - $s < 0$ a $t < 0$, znamená to, že v k-tém kroku aglomerativního shlukování došlo ke sloučení pozorování $\mathbf{x}_s$ a $\mathbf{x}_t$,

- $s > 0$ a $t < 0$, znamená to, že pozorování $\mathbf{x}_t$ bylo v k-tém kroku algoritmu připojeno ke shluku, který vznikl v s-tém řádku matice `merge` (samozřejmě $s < k$),
- $s > 0$ a $t > 0$, znamená to, že došlo ke spojení dvou víceprvkových shluků, které vznikly v s-tém a t-tém řádku matice `merge`.

Poznamenejme, že první řádek matice `merge` tvoří vždy dvě záporná čísla.

- `height` - vzdálenosti mezi spojujícími se shluky (podle argumentů `metric` a `method`)

Použití funkce `agnes` si předvedeme na následujícím příkladě:

Příklad 3.1: Mějme 5 pozorování $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$ s danou maticí vzdáleností

$$\mathbf{A} = \begin{pmatrix} & \{a\} & \{b\} & \{c\} & \{d\} & \{e\} \\ \{a\} & 0 & 3 & 1 & 10 & 9 \\ \{b\} & 3 & 0 & 5 & 9 & 8 \\ \{c\} & 1 & 5 & 0 & 6 & 5 \\ \{d\} & 10 & 9 & 6 & 0 & 2 \\ \{e\} & 9 & 8 & 5 & 2 & 0 \end{pmatrix}.$$

Jestliže zadáme příkaz

```
agnes(A, diss=T, metric="euclidean", method="average"),
```

dostaneme

```
Agglomerative coefficient:  0.7446809
Order of objects:
[1] 1 3 2 4 5
Merge:
     [,1] [,2]
[1,]   -1   -3
[2,]   -4   -5
[3,]    1   -2
[4,]    3    2
Height:
[1] 1.000000 4.000000 7.833333 2.000000

10 dissimilarities, summarized :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    3.50    5.50    5.80    8.75   10.00

Number of objects : 5
```

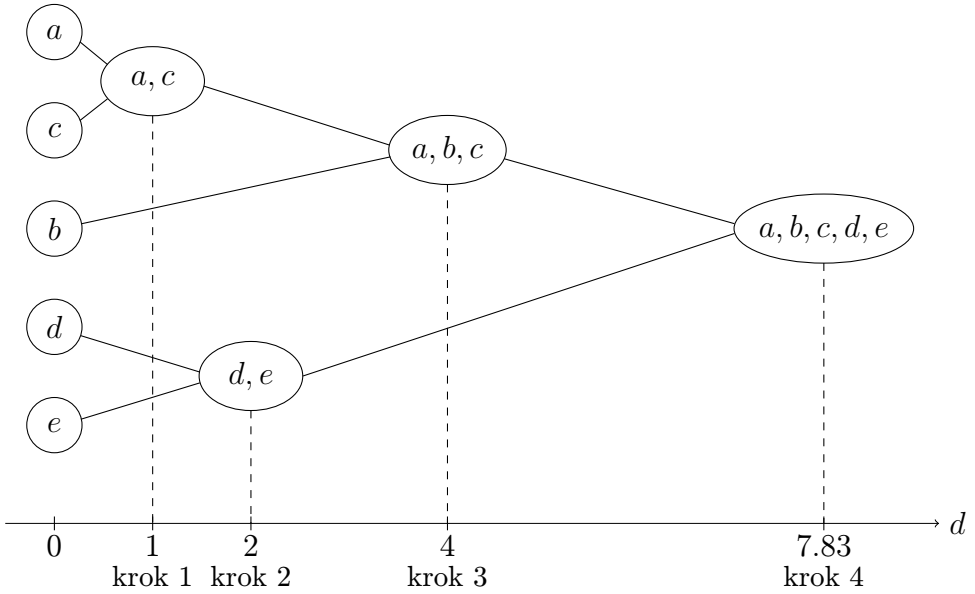
Poznamenejme, že pozorování $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$ jsou zakódována ve výstupu pod čísly 1 až 5. Z výstupu u položky `merge` vidíme, že jako první se spojovala pozorování $\{a\}$ a $\{c\}$ {viz čísla -1 a -3 v prvním řádku matice). Jejich vzdálenost je rovna 1, což lze přečíst přímo v zadané matici vzdáleností. Následně došlo ke spojení pozorování $\{d\}$ a $\{e\}$ (označených čísly 4 a 5, viz -4 a -5 v druhém řádku matice), jejich vzdálenost je podle zadané matice vzdáleností 2. Ve třetím kroku algoritmu se spojilo pozorování $\{b\}$ se shlukem $\{a, c\}$ zakódovaným pod číslem 1 (tedy viz první řádek matice `merge`). Vzdálenost $\{b\}$ a $\{a, c\}$ spočteme podle metody average linkage (1.12) jako

$$d\big(\{b\},\{a,c\}\big) = \frac{1}{1\cdot 2}\cdot\Big(d(\{b\},\{a\}) + d(\{b\},\{c\})\Big) = \frac{1}{2}\cdot(3+5) = 4. \tag{3.1}$$

Na závěr došlo ke spojení shluků $\{a, b, c\}$ a $\{d, e\}$. Pro vzdálenost těchto shluků opět podle (1.12) platí:

$$\begin{aligned} d\big(\{a,b,c\},\{d,e\}\big) \;&=\; \frac{1}{3\cdot 2}\cdot\Big(d(\{a\},\{d\}) + d(\{a\},\{e\}) + d(\{b\},\{d\}) + \\ &+\; d(\{b\},\{e\}) + d(\{c\},\{d\}) + d(\{c\},\{e\})\Big) = \\ &=\; \frac{1}{6}\cdot(10+9+9+8+6+5) \doteq 7{,}83. \end{aligned} \tag{3.2}$$

Postup shlukování v tomto případě vidíme na obrázku 3.2. Jedná se o 90 stupňů otočený dendrogram z obrázku 3.1.



Obrázek 3.2: Postup aglomerativního shlukování pro příklad 3.1.

Nyní se věnujme řádkům `Order` a `Height`, které si přepíšeme pod sebe:

```
Order of objects:  1        3        2        4        5
Height:             1.000000 4.000000 7.833333 2.000000
```

Ukážeme si, že pouze z těchto dvou řádků lze sestrojit graf jako na obrázku 3.2. Podívejme se na nejmenší číslo ve vektoru `Height`. Je to číslo 1,00 a nalézá se mezi objekty $\{a\}$ a $\{c\}$ (mezi čísly 1 a 3). Podle toho tedy víme, že tyto objekty se shlukovaly jako první. Druhé nejmenší číslo je 2,00, které svou polohou mezi číslicemi 4 a 5 indikuje, že v druhém kroku došlo ke spojení $\{d\}$ a $\{e\}$. Třetí nejmenší číslo je 4,00, které se nalézá mezi 3 a 2 a tedy mezi $\{c\}$ a $\{b\}$. Pozorování $\{c\}$ už je ale přiřazeno k $\{a\}$ a tedy ve třetím kroku algoritmu se $\{b\}$ se přiřadí k $\{a, c\}$. Vzdálenost shluků jsme již vypočetli v (3.1). Největší číslo 7,83 se nalézá mezi 2 a 4, (tedy mezi $\{b\}$ a $\{d\}$), ale $\{b\}$ je již spojeno s $\{a, c\}$ a $\{d\}$ je spojeno s $\{e\}$, takže dochází ke spojení $\{a, b, c\}$ a $\{d, e\}$. Vzdálenost mezi shluky jsme vypočetli v (3.2).

Co se týče grafického znázornění aglomerativních postupů, lze použít takzvaný *banner plot*. Starší verzi tohoto grafu pro příklad 3.1 můžeme vidět na obrázku 3.3.

```
------------------------------------------------------------
|      aaa+aaa+aaa+aaa+aaa+aaa+aaa+aaa+aaa+aaa+aaa+aaa+aaa+a|
|      *****************************************************|
|      ccc+ccc+ccc+ccc+ccc+ccc+ccc+ccc+ccc+ccc+ccc+ccc+ccc+c|
|                           ********************************|
|                           bbb+bbb+bbb+bbb+bbb+bbb+bbb+bbb+|
|                                                       ***|
|             ddd+ddd+ddd+ddd+ddd+ddd+ddd+ddd+ddd+ddd+ddd+dd|
|             *********************************************|
|             eee+eee+eee+eee+eee+eee+eee+eee+eee+eee+eee+ee|
0------1------2------3------4------5------6------7-----7.83-
```

Obrázek 3.3: Starší verze banner plotu pro příklad 3.1.

Graf se skládá z hvězdiček a řad objektů. Hvězdičky charakterizují spojení příslušných shluků. Graf čteme zleva doprava. Bílý prostor v levé části značí, že doposud žádná pozorování nebyla spojena. Posléze pro $d = 1$ dochází ke spojení $\{a\}$ a $\{c\}$, pro $d = 2$ ke spojení $\{d\}$ a $\{e\}$ a pro $d = 4$ dochází k připojení $\{b\}$ ke shluku $\{a, c\}$. Graficky vylepšenou verzi vidíme na obrázku 3.4. Tento graf se skládá z barevných obdélníčků, které indikují spojení dvou shluků. Na vodorovnou osu nanášíme vzdálenosti mezi spojenými shluky, svislá osa reprezentuje pořadí pozorování, jak byla vypočtena ve výstupu `order`. K banner plotu v závěru poznamenejme, že tento graf nelze sestrojit v případě metody centroidního spojení, neboť vzdálenosti mezi spojujícími se shluky nejsou monotónní. Jak jsme již říkali, hlavním grafickým nástrojem hierarchických metod je dendrogram. Pro příklad 3.1 je znázorněn na obrázku 3.1. ♢

Obrázek 3.4: Banner plot pro příklad 3.1.

3.2 Posouzení kvality shlukování

Hlavním nástrojem posouzení kvality shlukování v aglomerativních algoritmech je *aglomerativní koeficient*. Poskytuje informaci o množství struktury v datech, která byla nalezena v průběhu shlukování. Označme $m(i) =$ vzdálenost i-tého pozorování od prvního shluku, se kterým se i spojilo, vydělenou vzdáleností 2 shluků v posledním $(N-1)$-kroku algoritmu pro $i = 1, \ldots, N$. Pak aglomerativní koeficient definujeme jako

$$AC = \frac{1}{N} \cdot \sum_{i=1}^{N} \Big(1 - m(i)\Big). \tag{3.3}$$

V příkladě 3.1 je $N = 5$ a naposledy se shlukovaly $\{a, b, c\}$ a $\{d, e\}$, mezi kterými byla vzdálenost 7,83, což je dělitel v definici $m(i)$. Pro koeficient AC dostáváme

$$\begin{aligned} AC \;&=\; \frac{1}{5} \cdot \sum_{i=1}^{5} \Big(1 - m(i)\Big) = 1 - \frac{1}{5} \cdot \sum_{i=1}^{5} m(i) = \\ &=\; 1 - \frac{1}{5} \cdot \frac{1+4+1+2+2}{7{,}83} \doteq 0{,}7446. \end{aligned}$$

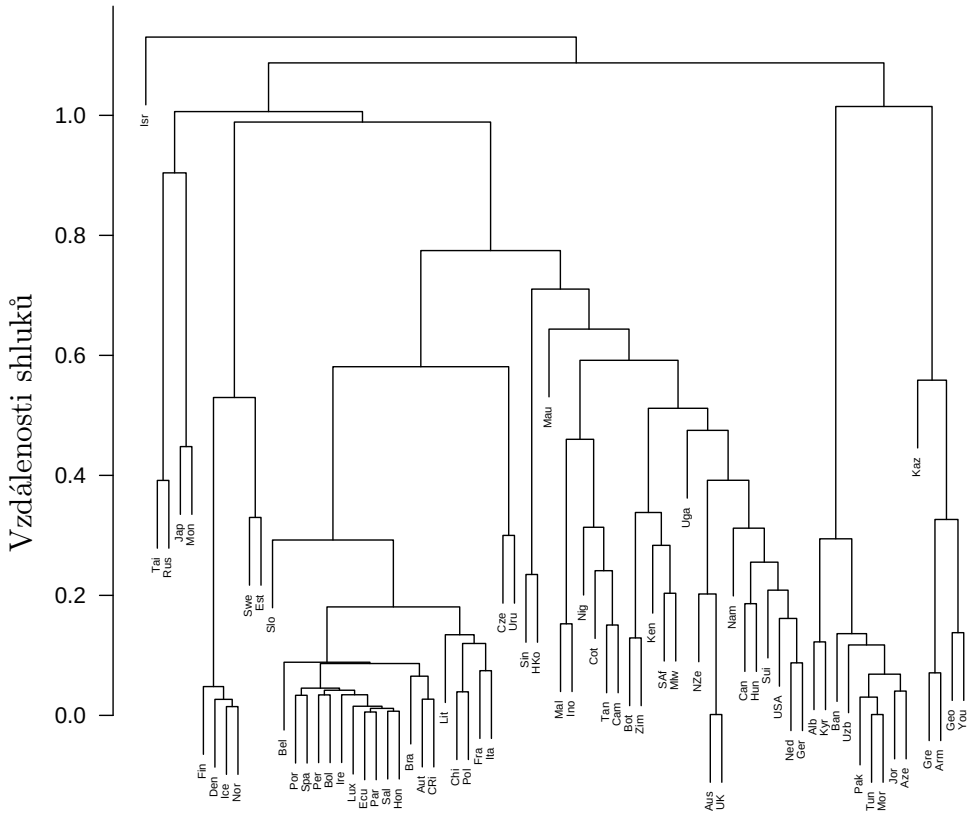
Jednotlivé sčítance v součtu $1 + 4 + 1 + 2 + 2$ dostaneme tak, že na obrázku 3.3 přečteme na vodorovné ose, odkud začínají sekvence `aaa`, `bbb`, atd. Čím vyšší je aglomerativní koeficient, tím více struktury se v datech podařilo nalézt. Některé extrémní příklady hodnot koeficientu AC spolu s grafickým znázorněním na banner plotu lze nalézt v [Kaufman a Rousseeuw, 1990] na str. 217–220.

3.3 Ilustrační příklad – pokračování

Připomeňme, že máme k dispozici data o procentuálním zastoupení náboženství v 72 státech světa. Nejprve musíme vybrat jeden z výše uvedených aglomerativních

postupů. O nevýhodách při použití algoritmů single linkage, complete linkage a centroidní metody jsme se již zmiňovali, některé další argumenty hovořící v neprospěch těchto metod lze nalézt v [Kaufman a Rousseeuw, 1990], str. 226–230.

Budeme tedy aplikovat metodu average linkage s eukleidovskou metriku. Po výpočtu dostaneme dendrogram na obrázku 3.5.



Group Average

Obrázek 3.5: Dendrogram pro ilustrační příklad.

Vidíme, že se shlukovaly velmi brzy státy severní Evropy Dánsko, Norsko, Island, Finsko a také státy, které byly při kolonizaci Ameriky v 16. století pod španělským a portugalským vlivem (Ekvádor, Paraguay, Salvádor, Honduras). V případě metod k-means a k-medoids jsou tyto státy rovněž v jednom shluku, což je vidět z tabulek 2.6 a 2.8. Z těchto tabulek ale nevyčteme, jak moc jsou si tyto státy blízké ve smyslu vhodné metriky. Vzájemná blízkost jednotlivých států z hlediska náboženské

struktury je patrná právě až z dendrogramu na obrázku 3.5, který zachycuje proces shlukování. Na tomto obrázku si všimněme ještě jednoho faktu: Izrael se připojil až v samém závěru aglomerativního shlukování. To může indikovat dvě věci: jednak je možné, že se jedná o chybné pozorování, nebo je Izrael skladbou náboženské struktury zcela specifický od všech ostatních států. Z tabulky 2.2 vidíme, že se jedná o druhou alternativu, neboť Izrael má nejvyšší hodnotu ve sloupci `Res` − a to 0,89. Je to z toho důvodu, že v Izraeli dominuje judaismus, který v datech [Paldam, 2000] nefiguruje jako samostatné náboženství, ale je přiřazen do sloupce „Ostatní".

Porovnáme ještě jednotlivé aglomerativní postupy v závislosti na použité metrice a metodě. Ke srovnání použijeme aglomerativní koeficient (3.3). Výsledné hodnoty koeficientů AC pro různé metriky a hierarchické algoritmy obsahuje tabulka 3.1.

Tabulka 3.1: Srovnání aglomerativních postupů pomocí AC.

	metrika	
	manhattan	euclidean
single linkage	0,774	0,797
average linkage	0,826	0,853
complete linkage	0,835	0,877

Z tabulky 3.1 vidíme, že nejvyšší hodnotu aglomerativního koeficientu docílíme při použití metody complete linkage a eukleidovské metriky. Výsledek použití algoritmu complete linkage za použití eukleidovské vzdálenosti uvádí obrázek 3.6.

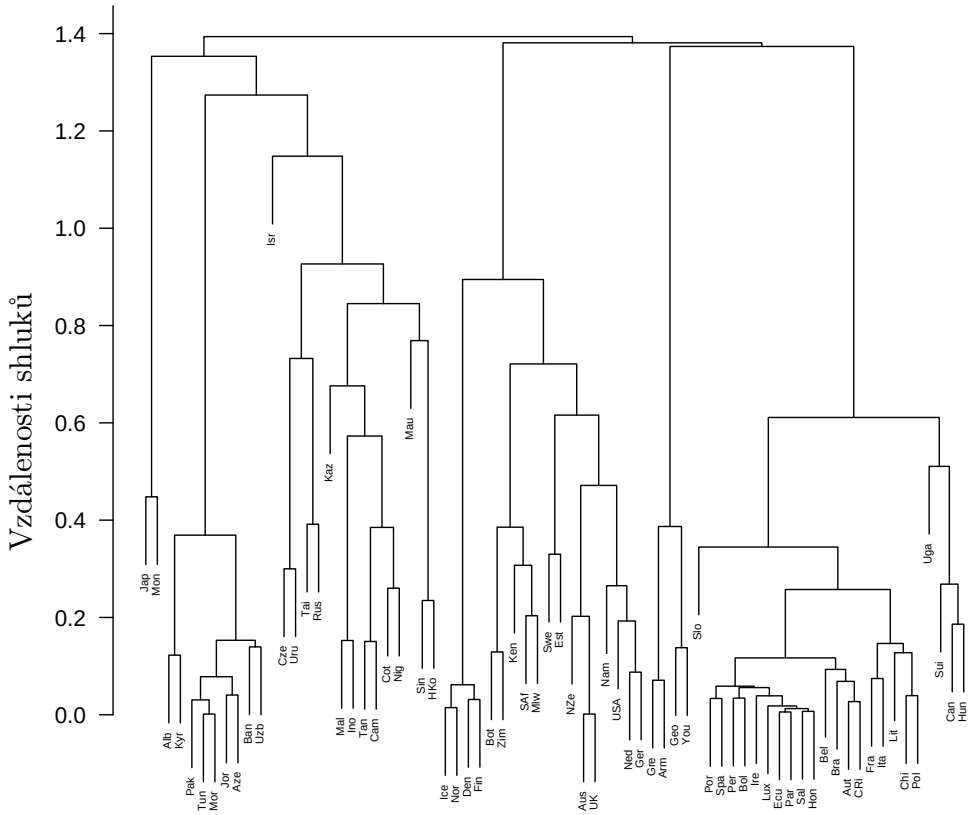
Nyní provedeme porovnání všech výše uvedených shlukovacích postupů, které jsme použili v předchozím textu. Porovnáme algoritmus k-means, k-medoids s eukleidovskou metrikou, fuzzy shlukování a aglomerativní metody average linkage a complete linkage s eukleidovskou metrikou. Jako funkcionály hodnotící kvalitu shlukování použijeme WSS (viz str. 29), w_K (viz str. 28) a dále

$$MSS_K(C) := \sum_{k=1}^{K} \sum_{\{i:\, C(i)=k\}} \|\mathbf{x}_i - \mathbf{m}_k\|^2,$$

kde $\mathbf{m}_k$ je medoid k-tého shluku.

Připomeňme, že vyšší kvalitu shlukování dávají nižší hodnoty WSS_K a MSS_K a vyšší hodnoty w_K. Shrnutí dosažených výsledků vidíme v tabulce 3.2 pro rozklad do $K = 3$, 6 a 9 shluků za použití eukleidovské metriky. V každém řádku je vždy červeně vyznačena nejlepší hodnota.

Z tabulky 3.2 si všimněme několika skutečností: S rostoucím celkovým počtem shluků dochází ke zmenšování hodnot vnitroshlukových funkcionálů rozkladu a ke zvětšování hodnot funkcionálu založeným na siluetové funkci. Je tedy $WSS_3 > WSS_6 > WSS_9$ a také $MSS_3 > MSS_6 > MSS_9$ u všech shlukovacích metod, což odpovídá tomu, že s rostoucím počtem shluků K klesá součet vzdáleností mezi



Complete Linkage

Obrázek 3.6: Dendrogram pro ilustrační příklad.

objekty a jimi příslušnými reprezentativními objekty. Dalším zajímavým poznatkem z tabulky 3.2 je skutečnost, že s rostoucím počtem shluků se začínají hodnoty WSS a MSS pro metodu average linkage a complete linkage přibližovat hodnotám těchto funkcionálů u zbývajících metod. Jedním z možných vysvětlení tohoto jevu může být již zmíněné konstatování z úvodu kapitoly 3, že aglomerativní techniky kumulují náhodné chyby s postupným spojováním do menšího počtu shluků.

Nyní můžeme podle funkcionálů rozkladu srovnat jednotlivé metody. Vidíme, že metoda k-means dopadla ve většině vyšetřovaných případech nejlépe. Dokonce pro rozklad do 9 shluků vyšla u této metody nejnižší hodnota funkcionálů WSS, MSS a nejvyšší hodnota funkcionálu w. Metoda k-medoids se ukázala jako vhodná pro rozklad do šesti shluků, neboť byla za k-means druhá nejlepší v optimalizaci WSS

Tabulka 3.2: Srovnání shlukovacích postupů (eukleidovská metrika).

K	fcionály	k-means	k-medoids	fuzzy	average linkage	complete linkage
$K = 3$	WSS_3	13,609	13,631	13,995	21,102	16,962
	w_3	0,494	0,400	0,392	0,332	0,319
	MSS_3	16,400	16,399	15,872	24,069	20,422
$K = 6$	WSS_6	6,702	6,732	7,112	11,722	7,855
	w_6	0,606	0,490	0,450	0,392	0,452
	MSS_6	8,213	8,033	9,300	13,621	10,005
$K = 9$	WSS_9	4,000	4,235	4,902	5,071	4,681
	w_9	0,652	0,509	0,475	0,489	0,486
	MSS_9	4,974	5,014	5,960	6,417	6,339

a w a hodnotu MSS má na této hladině shluků nejnižší. Metody average linkage a complete linkage vychází algoritmicky z jiného postupu, takže se dalo očekávat, že při optimalizaci navržených funkcionálů rozkladu nedopadnou dobře.

Hodnoty v tabulce 3.2 byly vypočteny na základě procedur, které jsou uvedeny v příloze A. U hierarchických metod bylo potřeba před výpočtem funkcionálů useknout dendrogram na zvolené hladině K a průchodem binárního stromu postupně vypsat skupiny pozorování patřících do stejného shluku.

Kapitola 4

Archetypální analýza

Archetypální analýza představuje jeden z nástrojů data miningu, který se v dané množině dat pokouší najít reprezentativní prvky, tzv. *archetypy*. Vznik archetypální analýzy se datuje rokem 1994, kdy Adele Cutler a Leo Breiman sestavili algoritmus pro hledání archetypů a publikovali ho v [Cutler a Breiman, 1994]. Cílem archetypální analýzy je najít reprezentativní objekty jako smysluplné lineární kombinace původních dat. Výhodou archetypů je jejich snadná logická interpretace, jak vyplyne z jejich tvaru, který nyní uvedeme.

Definice 4.1: *Nechť* $\mathbf{x}_1, \ldots, \mathbf{x}_n$ *jsou* p*-rozměrné objekty a nechť* $m \in \mathbb{N}$, $m < n$. *Objekty* $\mathbf{z}_1, \ldots, \mathbf{z}_m$ *tvaru*

$$\mathbf{z}_k := \sum_{j=1}^{n} \beta_{kj} \mathbf{x}_j, \quad k = 1, \ldots, m, \tag{4.1}$$

takové, že minimalizují

$$\min_{\{\alpha_{ik}\}} \left\{ \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{k=1}^{m} \alpha_{ik} \mathbf{z}_k \right\|^2 \right\} \tag{4.2}$$

za podmínek

$$\alpha_{ik} \geq 0, \quad i = 1, \ldots, n, \; k = 1, \ldots, m, \tag{4.3}$$

$$\sum_{k=1}^{m} \alpha_{ik} = 1, \quad i = 1, \ldots, n, \tag{4.4}$$

$$\beta_{kj} \geq 0, \quad k = 1, \ldots, m, \; j = 1, \ldots, n, \tag{4.5}$$

$$\sum_{j=1}^{n} \beta_{kj} = 1, \quad k = 1, \ldots, m \tag{4.6}$$

se nazývají archetypy.

Označme

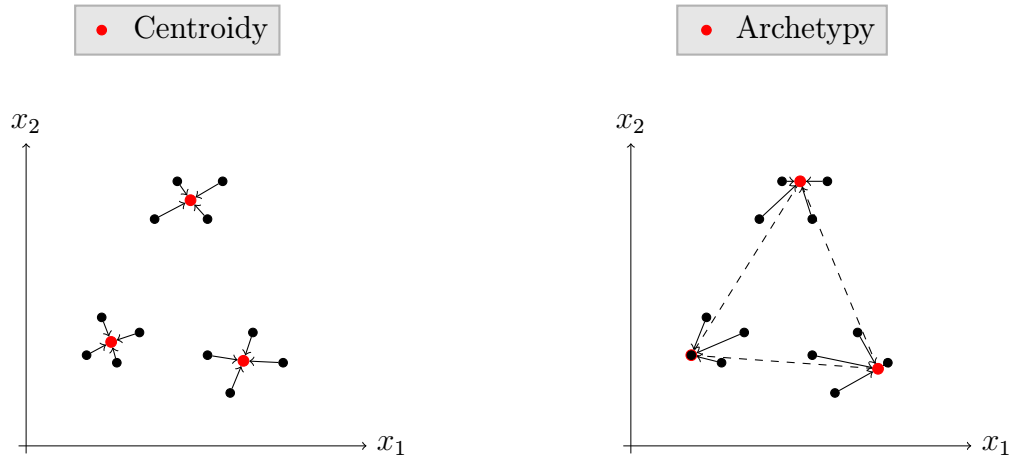
$$RSS := \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{k=1}^{m} \alpha_{ik} \mathbf{z}_k \right\|^2 . \tag{4.7}$$

Dosadíme-li v definici 4.1 výraz (4.1) do (4.2), dostaneme

$$\min_{\{\alpha_{ik}\},\{\beta_{kj}\}} \left\{ \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{k=1}^{m} \alpha_{ik} \sum_{j=1}^{n} \beta_{kj} \mathbf{x}_j \right\|^2 \right\}, \tag{4.8}$$

takže úloha nalezení archetypů $\mathbf{z}_1, \ldots, \mathbf{z}_m$ byla převedena na problém najít $\{\alpha_{ik}\}$ a $\{\beta_{kj}\}$ splňující (4.3) až (4.6), které minimalizují RSS. V případě, že $m = 1$, je nalezený archetyp průměrem ze všech pozorování.

Archetypální analýza i metody k-means a k-medoids využívají k popisu dat reprezentativní objekty. Rozdíl mezi přístupem archetypální analýzy a nehierarchickými metodami je patrný z obrázku 4.1.



Obrázek 4.1: Rozdíl mezi centry u metody k-means a archetypy.

Zatímco nehierarchické metody analyzují data „zevnitř“ a hledají reprezentativní objekty „uprostřed“ množiny dat, archetypy jakožto nezáporné lineární kombinace se součtem vah 1 popisují data „zvenku“. Vágně řečeno, vytvořené shluky se v případě nehierarchických metod formují symetricky okolo medoidů (centroidů), u archetypů se shluky vytváří pouze z určité strany. Zatímco centroidy a medoidy spíše vyvažují množinu dat, archetypy působí jako do jisté míry extrémní hodnoty, neboť jak je dokázáno v [Cutler a Breiman, 1994], leží v konvexním obalu dat. Díky požadavku (4.5) jsou archetypy „podobné“ původním datům a díky požadavku (4.6) jsou také směsí původních dat. Podmínka (4.3) vyjadřuje, že každý objekt je smysluplnou kombinací archetypů a podmínka (4.4) značí, že data jsou směsí archetypů.

Označme

$$\mathbf{A} := \begin{pmatrix} \alpha_{11} & \dots & \alpha_{1m} \\ \vdots & \ddots & \vdots \\ \alpha_{n1} & \dots & \alpha_{nm} \end{pmatrix}, \quad \mathbf{X} := \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{1,\bullet} \\ \vdots \\ \mathbf{x}_{n,\bullet} \end{pmatrix},$$

$$\mathbf{B} := \begin{pmatrix} \beta_{11} & \dots & \beta_{1n} \\ \vdots & \ddots & \vdots \\ \beta_{m1} & \dots & \beta_{mn} \end{pmatrix}, \quad \mathbf{Z} := \begin{pmatrix} z_{11} & \dots & z_{1p} \\ \vdots & \ddots & \vdots \\ z_{m1} & \dots & z_{mp} \end{pmatrix} = \begin{pmatrix} \mathbf{z}_{1,\bullet} \\ \vdots \\ \mathbf{z}_{m,\bullet} \end{pmatrix}.$$

Potom podle (4.1) platí $\mathbf{Z} = \mathbf{BX}$. Archetypy pak mohou být nalezeny tak, že najdeme matice $\mathbf{A}$ a $\mathbf{B}$ takové, že $\|\mathbf{X} - \mathbf{AZ}\| = \|\mathbf{X} - \mathbf{ABX}\|$ je minimální, kde $\|\cdot\|$ je vhodně zvolená norma matice a kde matice $\mathbf{A}$ a $\mathbf{B}$ jsou nezáporné s řádkovými součty 1. Tato úloha je součástí širší třídy úloh označovaných jako *Nonnegative Matrix Factorization*. Více např. ve [Finesso a Spreij, 2004].

4.1 Řešení úloh nejmenších čtverců

Problém nalezení archetypů $\mathbf{z}_1, \dots, \mathbf{z}_m$ budeme řešit jako úlohu nejmenších čtverců s okrajovými podmínkami. Nejprve si připomeneme úlohy nejmenších čtverců. V následujících odstavcích budeme uvažovat standardní eukleidovskou normu vektoru $\mathbf{u}$

$$\|\mathbf{u}\| := \sqrt{\mathbf{u}^\top \mathbf{u}} = \left(\sum_{i=1}^{n} u_i^2 \right)^{\frac{1}{2}}.$$

Nechť $\mathbf{E}$ je matice typu $m_2 \times n$, $\mathbf{f}$ je m_2-složkový vektor. Uvažujme dále matici $\mathbf{G}$ typu $m \times n$ a m-složkový vektor $\mathbf{h}$. V tomto odstavci budeme hledat n-složkový vektor $\mathbf{x}$, který řeší jednotlivě následující problémy nejmenších čtverců:

- ULS (= Unconstrained Least Squares problem)

$$\min \|\mathbf{Ex} - \mathbf{f}\|$$

- LSI (= Least Squares with Inequality Constraints)

$$\min \|\mathbf{Ex} - \mathbf{f}\| \text{ za podmínky } \mathbf{Gx} \geq \mathbf{h} \tag{4.9}$$

- NNLS (= Non-Negative Least Squares)

$$\min \|\mathbf{Ex} - \mathbf{f}\| \text{ za podmínky } \mathbf{x} \geq \mathbf{0} \tag{4.10}$$

- LDP (= Linear Distance Programming)

$$\min \|\mathbf{x}\| \text{ za podmínky } \mathbf{Gx} \geq \mathbf{h} \tag{4.11}$$

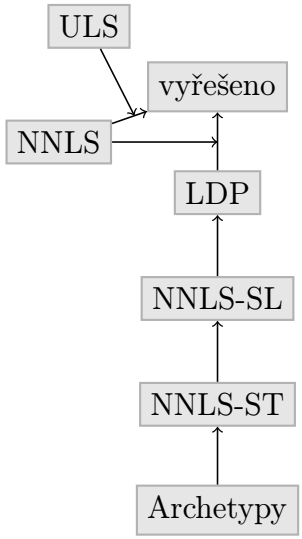
- NNLS-SL (= Non-Negative Least Squares with Sum Less or Equal to One)

$$\min \|\mathbf{E}\mathbf{x} - \mathbf{f}\| \text{ za podmínky } \mathbf{x} \geq \mathbf{0} \text{ a } \mathbf{1}^\top \mathbf{x} \leq \mathbf{1} \tag{4.12}$$

- NNLS-ST (= Non-Negative Least Squares with Sum To One)

$$\min \|\mathbf{E}\mathbf{x} - \mathbf{f}\| \text{ za podmínky } \mathbf{x} \geq \mathbf{0} \text{ a } \mathbf{1}^\top \mathbf{x} = \mathbf{1} \tag{4.13}$$

Poslední zmiňovaný problém nejmenších čtverců NNLS-ST povede k nalezení archetypů. Na obrázku 4.2 vidíme, jak řešení jednotlivých obtížnějších úloh nejmenších čtverců bude využívat ty jednodušší.



Obrázek 4.2: Postup řešení algoritmů nejmenších čtverců.

ULS

Klasický problém nejmenších čtverců může být formulován následujícím způsobem:

Problém ULS:
Pro zadanou reálnou matici $\mathbf{A}$ typu $m \times n$ hodnosti $k \leq \min\{m, n\}$ a m-rozměrný reálný vektor $\mathbf{b}$ nalezněme reálný n-rozměrný vektor $\mathbf{x}$, který minimalizuje $\|\mathbf{A}\mathbf{x}-\mathbf{b}\|$.

K řešení problému ULS využijeme QR rozklad matice $\mathbf{A}$, tedy nalezneme ortogonální matici $\mathbf{Q}$ a horní trojúhelníkovou matici $\mathbf{R}$, takové, že $\mathbf{A} = \mathbf{QR}$.

Učiníme ještě několik poznámek ke QR rozkladu matice $\mathbf{A}$ typu $m \times n$ týkající se výpočtu v softwaru `R`.

Poznámky 4.2:
Nalezení QR rozkladu se v softwaru `R` provádí pomocí příkazů

```
Q<-qr.Q(qr(A))
R<-qr.R(qr(A)).
```

Mohou nastat tyto případy:

1. Jestliže $m > n$ a $\mathsf{rank}\,\mathbf{A} = n$, pak $\mathbf{Q}$ je typu $m \times n$ a $\mathbf{R}$ je regulární typu $n \times n$.
2. Jestliže $m > n$ a $\mathsf{rank}\,\mathbf{A} < n$, pak $\mathbf{Q}$ je typu $m \times n$ a $\mathbf{R}$ je singulární typu $n \times n$.
3. Jestliže $m \leq n$ a $\mathsf{rank}\,\mathbf{A} \leq m$, pak $\mathbf{Q}$ je ortogonální typu $m \times m$ a $\mathbf{R}$ je typu $m \times n$.

Pro další výpočty budeme potřebovat, aby matice $\mathbf{Q}$ byla vždy *ortogonální*, tedy aby byla čtvercová a platilo $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$, kde $\mathbf{I}$ je jednotková matice řádu m. Abychom toho docílili pro jakýkoliv rozměr matice $\mathbf{A}$, budeme modifikovat výše uvedené případy 1 a 2 (tj. situaci, kdy $m > n$), neboť v těchto případech matice $\mathbf{Q}$ není čtvercová: V prostředí `R` použijeme příkazů

```
Q<-qr.Q(qr(A),complete=TRUE)
R<-qr.R(qr(A),complete=TRUE)
```

Dostaneme ortogonální matici $\widetilde{\mathbf{Q}}$ $m \times m$ a matici $\widetilde{\mathbf{R}}$ typu $m \times n$. Matice $\widetilde{\mathbf{R}}$ bude mít prvních n řádků shodných s maticí $\mathbf{R}$, která vznikla pomocí příkazu `qr.R(qr(A))` a v posledních $m - n$ řádcích přibudou nuly. Bude platit $\mathbf{A} = \widetilde{\mathbf{Q}}\widetilde{\mathbf{R}}$.

NNLS

V této části bude uveden algoritmus nezáporných nejmenších čtverců, k jehož pochopení budeme potřebovat jednu fundamentální větu z teorie nelineárního programování známou jako Kuhn-Tuckerova věta. My budeme tuto větu formulovat v podobě, uvedené v [Norstad, 2005], která bude dávat nutnou i postačující podmínkou nalezení optimálního řešení.

Věta 4.3: *Definujme* $M := \{\mathbf{x} \in \mathbb{R}^n : h_j(\mathbf{x}) = 0,\ j = 1, \ldots, p, \quad g_k(\mathbf{x}) \leq 0,\ k = 1, \ldots, m\} \neq \emptyset$, *kde* $h_j(\cdot)$, $j = 1, \ldots, p$ *a* $g_k(\cdot)$, $k = 1, \ldots, m$ *jsou lineární funkce. Nechť* $f : \mathbb{R}^n \to \mathbb{R}$ *je konvexní spojitě diferencovatelná funkce na* $\mathbb{R}^n$. *Pak* $\mathbf{x}^*$ *je optimální řešení úlohy*

$$\min_{\mathbf{x} \in M} f(\mathbf{x}),$$

právě když existují Lagrangeovy multiplikátory $\boldsymbol{\lambda} \in \mathbb{R}^p$ *a* $\boldsymbol{\mu} \in \mathbb{R}^m$, *které splňují:*

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^{p} \lambda_j \nabla h_j(\mathbf{x}^*) + \sum_{k=1}^{m} \mu_k \nabla g_k(\mathbf{x}^*) \ = \ \mathbf{0}, \tag{4.14}$$

$$\boldsymbol{\mu} \ \geq \ \mathbf{0}, \tag{4.15}$$

$$\sum_{k=1}^{m} \mu_k \, g_k(\mathbf{x}^*) \ = \ 0. \tag{4.16}$$

Důkaz: viz [McLennan, 1999]. ∎

Zabývejme se tedy problémem NNLS (4.10). Potom ve větě 4.3 je $m := n$, $p := 0$, $g_k(\mathbf{x}) := -x_k$, $k = 1, \ldots, n$, $M := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq \mathbf{0}\}$.

Za účelovou funkci vezměme $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{E}\mathbf{x}-\mathbf{f}\|^2$, tím se zřejmě nezmění optimální řešení $\mathbf{x}^*$ pro problém NNLS.

Jsou splněny předpoklady věty 4.3, neboť omezení $g_k(\mathbf{x})$, $k = 1, \ldots, n$ jsou zřejmě lineární, $M \neq \emptyset$, účelová funkce je spojitě diferencovatelná a konvexní v celém $\mathbb{R}^n$.

Utvořme Lagrangeovu funkci

$$L(\mathbf{x}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{E}\mathbf{x} - \mathbf{f}\|^2 - \boldsymbol{\mu}^\top \mathbf{x}$$

a přepišme postupně podmínky (4.14), (4.15) a (4.16). Dostaneme vztahy

$$\begin{aligned} \boldsymbol{\mu} &= \mathbf{E}^\top(\mathbf{E}\mathbf{x}^* - \mathbf{f}), \\ \boldsymbol{\mu} &\geq \mathbf{0}, \\ \boldsymbol{\mu}^\top \mathbf{x}^* &= 0. \end{aligned}$$

Položme

$$\mathbf{w} := \mathbf{E}^\top(\mathbf{f} - \mathbf{E}\mathbf{x}^*) = -\boldsymbol{\mu}.$$

Označme $\mathcal{P} := \{j = 1, \ldots, n : x_j^* > 0\}$ a $\mathcal{Z} := \{j = 1, \ldots, n : x_j^* = 0\}$. Potom pro vektor řešení $\mathbf{x}$ a vektor $\mathbf{w}$, který představuje duální řešení úlohy NNLS, platí:

$$\begin{array}{llll} x_j > 0 & \text{pro} & j \in \mathcal{P} & \qquad w_j = 0 \quad \text{pro} \quad j \in \mathcal{P} \\ x_j = 0 & \text{pro} & j \in \mathcal{Z} & \qquad w_j \leq 0 \quad \text{pro} \quad j \in \mathcal{Z}. \end{array}$$

Nyní uvedeme algoritmus na řešení úlohy NNLS uvedený v [Lawson a Hanson, 1974], do něhož vstupuje matice $\mathbf{E}$ typu $m \times n$ a m-složkový vektor $\mathbf{f}$.

ALGORITMUS NNLS:

0. **Inicializace:**
   Položme $\mathbf{x} := \mathbf{0}$, $\mathcal{P} := \emptyset$, $\mathcal{Z} := \{1, 2, \ldots, n\}$.

1. **Vstup do hlavního cyklu:**
   Vypočteme vektor $\mathbf{w} := \mathbf{E}^\top(\mathbf{f} - \mathbf{E}\mathbf{x})$.

2. **Výstup z hlavního cyklu i celého algoritmu:**
   Jestliže $\mathcal{Z} = \emptyset$ nebo $w_j \leq 0$ $\forall j \in \mathcal{Z}$ jdeme na krok 10.

3. Nalezneme index $t \in \mathcal{Z}$ takový, že $w_t = \max\{w_j,\, j \in \mathcal{Z}\}$,
   tedy $t = \text{argmax}\{w_j, j \in \mathcal{Z}\}$.

4. Přesuneme index t z množiny $\mathcal{Z}$ do množiny $\mathcal{P}$.

5. **Vstup do vnitřního cyklu:**
   Označme $\mathbf{E}_{\mathcal{P}}$ matici typu $m \times n$ definovanou jako

$$j\text{-tý sloupec matice } \mathbf{E}_{\mathcal{P}} := \begin{cases} j\text{-tý sloupec matice } \mathbf{E} & \text{pro } j \in \mathcal{P} \\ \mathbf{0} & \text{pro } j \in \mathcal{Z}. \end{cases}$$

   Vypočteme vektor $\mathbf{z} \in \mathbb{R}^n$ jako řešení problému nejmenších čtverců

$$\mathbf{z} := \underset{\mathbf{u} \in \mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{f} - \mathbf{E}_{\mathcal{P}}\mathbf{u}\|.$$

   Pro $j \in \mathcal{Z}$ definujeme $z_j := 0$.

6. **Výstup z vnitřního cyklu:**
   Jestliže $z_j > 0 \; \forall j \in \mathcal{P}$, položíme $\mathbf{x} := \mathbf{z}$ a jdeme na krok 1.

7. Nalezneme index $p \in \mathcal{P}$ takový, že

$$\alpha := \frac{x_p}{x_p - z_p} = \min_{z_j \leq 0, j \in \mathcal{P}} \left\{ \frac{x_j}{x_j - z_j} \right\}.$$

8. Položíme $\mathbf{x} := \mathbf{x} + \alpha(\mathbf{z} - \mathbf{x})$.

9. Přesuneme z množiny $\mathcal{P}$ do množiny $\mathcal{Z}$ všechny indexy $j \in \mathcal{P}$, pro které $x_j = 0$ a jdeme na krok 5.

10. Algoritmus je u konce, $\mathbf{x}$ je optimální řešení.

Než budeme charakterizovat jednotlivé kroky algoritmu NNLS, uvedeme si následující tvrzení:

Lemma 4.4: *Nechť* $\mathbf{A}$ *je matice typu* $m \times n$ *hodnosti* n *a nechť* $\mathbf{b}$ *je* m*-rozměrný vektor splňující*

$$\mathbf{A}^\top \mathbf{b} = (\overbrace{0, 0, \ldots, 0}^{n-1}, \omega)^\top, \quad kde\, \omega > 0.$$

Jestliže

$$\widehat{\mathbf{x}} = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|,$$

pak $\widehat{x}_n > 0$.

Důkaz: [Lawson a Hanson, 1974], str. 162, lemma (23.17). ∎

Nyní budeme charakterizovat jednotlivé kroky algoritmu. Algoritmus NNLS se skládá ze dvou cyklů: První hlavní obsahuje kroky 1 až 4 a jeho začátek je v kroku 1 a výstup v kroku 2. Druhý vnitřní cyklus se skládá z kroků 5 až 10 se vstupem v kroku 5 a s výstupem v kroku 6.

V kroku 1 množina $\mathcal{P}$ obsahuje komponenty vektoru $\mathbf{x}$, které jsou momentálně kladné; v množině $\mathcal{Z}$ jsou indexy nulových složek vektoru $\mathbf{x}$. V kroku 3 se zvolí index t, který zatím není v množině $\mathcal{P}$. Jestliže bude zahrnutý do množiny $\mathcal{P}$, pak bude podle lemmatu 4.4 $x_t > 0$. Tento koeficient pak bude zahrnut do prozatímního řešení vektoru $\mathbf{z}$ v kroku 5. Jestliže všechny další komponenty vektoru $\mathbf{z}$ indexované množinou $\mathcal{P}$ zůstanou kladné, pak v kroku 6 se položí $\mathbf{x} := \mathbf{z}$ a algoritmus se přesune na začátek vnějšího cyklu. V právě popsaném procesu se množina $\mathcal{P}$ zvětšuje a množina $\mathcal{Z}$ zmenšuje přesouváním indexu t.

Jestliže se stane, že pro nějaký koeficient j z množiny $\mathcal{P}$ bude po kroku 5 $z_j \leq 0$, potom krok 6 pozdrží algoritmus uvnitř vnitřního cyklu a proběhne oprava $\mathbf{x} := \mathbf{x} + \alpha(\mathbf{z} - \mathbf{x})$, kde $0 < \alpha \leq 1$ je vybráno tak, aby aby nové $\mathbf{x}$ zůstalo nezáporné. Vnitřní cyklus se pak opakuje do té doby, než je eventuálně splněna podmínka v kroku 6.

Vnitřní i vnější cyklus je vždy konečný a algoritmus NNLS konverguje, jak je ukázáno v knize [Lawson a Hanson, 1974] s využitím lemmatu 4.4.

LDP

V této části uvedeme algoritmus LDP, který je publikován v [Lawson a Hanson, 1974] spolu s důkazem platnosti. Vstupem do algoritmu je matice $\mathbf{G}$ typu $m \times n$ a m-složkový vektor $\mathbf{h}$.

ALGORITMUS LDP:

1. **Inicializace:**
   Definujme matici $\mathbf{E}$ typu $(n+1) \times m$ a $(n+1)$-složkový vektor $\mathbf{f}$ jako
   $$\mathbf{E} := \begin{pmatrix} \mathbf{G}^\top \\ \mathbf{h}^\top \end{pmatrix}, \mathbf{f} := (\overbrace{0, 0, \ldots, 0}^{n}, 1)^\top.$$

2. **Použití NNLS:**
   Vypočteme $\widehat{\mathbf{u}} \in \mathbb{R}^m$ jako řešení úlohy $\min \|\mathbf{E}\mathbf{u} - \mathbf{f}\|$ za podmínky $\mathbf{u} \geq 0$.

3. Vypočteme $(n+1)$-rozměrný vektor $\mathbf{r} := \mathbf{E}\widehat{\mathbf{u}} - \mathbf{f}$.

4. Jestliže $\|\mathbf{r}\| = 0$, pak $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{G}\mathbf{x} \geq \mathbf{h}\} = \emptyset$ a jdeme na krok 6.

5. Pro $j = 1, \ldots, n$ vypočteme
   $$\widehat{x}_j := -\frac{r_j}{r_{n+1}}.$$

6. Výpočet je u konce.

V případě, že v kroku 4 platí $\|\mathbf{r}\| = 0$, je množina přípustných řešení pro úlohu LDP prázdná, proto nemá úloha LDP ani optimální řešení. To je dokázáno v [Lawson a Hanson, 1974], str. 167.

NNLS-SL

Úlohu (4.12) převedeme na (4.11) podle postupu v [Lawson a Hanson, 1974]. Nechť E je matice $m_2 \times n$, $\mathsf{rank}\,\mathsf{E} = n$. Označme

$$\mathsf{G} := \begin{pmatrix} \mathsf{I} \\ -\mathbf{1}^\top \end{pmatrix}, \quad \mathbf{h} := \begin{pmatrix} \mathbf{0} \\ -1 \end{pmatrix},$$

kde I je jednotková matice řádu n, G je typu $(n+1) \times n$ a $\mathbf{h}$ je $(n+1)$-složkový vektor. Pak úlohu (4.12) lze zapsat jako úlohu LSI (4.9) ve tvaru

$$\min \|\mathsf{E}\mathbf{x} - \mathbf{f}\| \text{ za podmínky } \mathsf{G}\mathbf{x} \geq \mathbf{h}. \tag{4.17}$$

Nalezneme ortogonální rozklad matice E:

$$\mathsf{E} = \mathsf{Q} \begin{pmatrix} \mathsf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathsf{K}^\top = (\mathsf{Q}_1\, \mathsf{Q}_2) \begin{pmatrix} \mathsf{R} \\ \mathbf{0} \end{pmatrix} \mathsf{K}^\top,$$

kde Q je ortogonální typu $m_2 \times m_2$, Q_1 je typu $m_2 \times n$, Q_2 je typu $m_2 \times (m_2 - n)$, K je ortogonální typu $n \times n$ a R je regulární typu $n \times n$. Položme

$$\mathbf{x} := \mathsf{K}\mathbf{y}, \tag{4.18}$$
$$\mathbf{f}_1 := \mathsf{Q}_1^\top \mathbf{f},$$
$$\mathbf{z} := \mathsf{R}\mathbf{y} - \mathbf{f}_1. \tag{4.19}$$

Místo úlohy (4.17) pak lze řešit ekvivalentní úlohu LDP ve tvaru

$$\min \|\mathbf{z}\| \text{ za podmínky } \mathsf{G}\mathsf{K}\mathsf{R}^{-1}\mathbf{z} \geq \mathbf{h} - \mathsf{G}\mathsf{K}\mathsf{R}^{-1}\mathbf{f}_1.$$

Řešení $\widehat{\mathbf{x}}$ úlohy NNLS-SL pak lze vypočítat z rovnic (4.19) a (4.18).

NNLS-ST

Řešení této úlohy se bude hodit pro archetypální analýzu. Úlohu (4.13) si vyjádříme ve tvaru:

$$\min \|\mathsf{E}\mathbf{x} - \mathbf{f}\| \text{ za podmínky } \mathbf{x} \geq \mathbf{0} \text{ a } x_1 = 1 - \sum_{i=2}^{n} x_i. \tag{4.20}$$

Úlohu (4.20) převedeme na (4.12):

$$\min \|\widetilde{\mathbf{E}}\widetilde{\mathbf{x}} - \widetilde{\mathbf{f}}\| \text{ za podmínky } \widetilde{\mathbf{x}} \geq \mathbf{0} \text{ a } \mathbf{1}^\top \widetilde{\mathbf{x}} \leq 1, \tag{4.21}$$

kde

$$\widetilde{\mathbf{x}} = (x_2, x_3, \ldots, x_n)^\top, \quad \widetilde{\mathbf{E}} = (\mathbf{e}_2 - \mathbf{e}_1, \ldots, \mathbf{e}_n - \mathbf{e}_1), \quad \widetilde{\mathbf{f}} = \mathbf{f} - \mathbf{e}_1.$$

Tím jsme úlohu NNLS-ST převedli na úlohu NNLS-SL. Jestliže $\widehat{\widetilde{\mathbf{x}}}$ je řešením úlohy (4.21), pak zbývá vypočítat $\widehat{x}_1$ jako

$$\widehat{x}_1 = 1 - \mathbf{1}^\top \widehat{\widetilde{\mathbf{x}}}.$$

4.2 Algoritmy pro nalezení archetypů

Připomeňme, že se pokoušíme pro danou matici dat $\mathbf{X}$ nalézt čísla α_{ik}, $i = 1, \ldots, n$, $k = 1, \ldots, m$ a β_{kj}, $k = 1, \ldots, m$, $j = 1, \ldots, n$ takové, že řeší úlohu (4.8) za podmínek (4.3) až (4.6). V „tečkovacím“ zápise platí $\mathbf{x}_i^\top = \mathbf{x}_{i,\bullet}$ pro každé $i = 1, \ldots, n$, kde $\mathbf{x}_i$ vyjadřuje i-té pozorování, které se umístí jako řádkový vektor $\mathbf{x}_{i,\bullet}$ do matice $\mathbf{X}$.

Myšlenka hledání archetypů je vzhledem k povaze úlohy založena na alternující optimalizaci. Střídavě se řeší dvě podúlohy. Pro pevné hodnoty β_{kj} se hledá optimální α_{ik} a naopak: Pro daná α_{ik} se hledají optimální β_{kj}. Obě tyto dílčí podúlohy − jak bude níže odvozeno − vyžadují řešení problému NNLS-ST.

Zabývejme se nejprve hledáním hodnot α_{ik}, $i = 1, \ldots, n$, $k = 1, \ldots, m$. Pro pevná $\mathbf{z}_1, \ldots, \mathbf{z}_m$ řešme úlohu

$$\min_{\{\alpha_{ik}\}} \left\{ \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{k=1}^{m} \alpha_{ik} \mathbf{z}_k \right\|^2 \right\} \tag{4.22}$$

za podmínek (4.3) a (4.4). Vzhledem k nezápornosti normy se minimum výrazu v (4.22) nalezne tak, že se pro každé $i = 1, \ldots, n$ zvlášť (tedy n-krát) řeší úloha

$$\min_{\{\alpha_{ik}\}} \left\{ \left\| \mathbf{x}_i - \sum_{k=1}^{m} \alpha_{ik} \mathbf{z}_k \right\|^2 \right\} \tag{4.23}$$

za podmínek

$$\alpha_{ik} \geq 0, \quad k = 1, \ldots, m, \quad \sum_{k=1}^{m} \alpha_{ik} = 1. \tag{4.24}$$

Úloha (4.23) s podmínkami (4.24) lze pro pevné $i = 1, \ldots, n$ zapsat jako úloha nejmenších čtverců v maticovém tvaru ve formě

$$\min_{\{\boldsymbol{\alpha}_{i,\bullet}\}} \left\| \mathbf{Z}^\top \boldsymbol{\alpha}_{i,\bullet}^\top - \mathbf{x}_i \right\|^2, \quad \text{za podmínek} \quad \boldsymbol{\alpha}_{i,\bullet}^\top \geq \mathbf{0}, \quad \mathbf{1}^\top \boldsymbol{\alpha}_{i,\bullet}^\top = 1, \tag{4.25}$$

kde $\boldsymbol{\alpha}_{i,\bullet} = (\alpha_{i1}, \ldots, \alpha_{im})$ je i-tý řádek matice $\mathbf{A}$ (braný jako řádkový vektor), $i = 1, \ldots, n$. Vidíme, že úloha (4.25) je úloha nejmenších čtverců NNLS-ST.

Analogicky se pokusíme zapsat problém nalezení β_{kj}, $k = 1, \ldots, m$, $j = 1, \ldots, n$, pro zadaná α_{ik}. Podle přístupu uvedeném v [Cutler a Breiman, 1994] spočítáme 1 archetyp, řekněme $\mathbf{z}_\ell$, ostatní archetypy podržíme konstantní. Ukážeme, že pro pevné $\ell = 1, \ldots, m$ je nalezení hodnot $\beta_{\ell j}$, $j = 1, \ldots, n$, rovněž problémem NNLS-ST. Položme

$$\begin{aligned} \mathbf{v}_i &:= \begin{cases} \frac{\mathbf{x}_i - \sum_{k \neq \ell} \alpha_{ik}\mathbf{z}_k}{\alpha_{i\ell}}, & \alpha_{i\ell} \neq 0 \\ \mathbf{0}, & \alpha_{i\ell} = 0 \end{cases}, \quad i = 1, \ldots, n. \\ \overline{\mathbf{v}} &:= \frac{\sum_{i=1}^n \alpha_{i\ell}^2 \mathbf{v}_i}{\sum_{i=1}^n \alpha_{i\ell}^2}, \quad \text{jestliže} \quad \sum_{i=1}^n \alpha_{i\ell}^2 > 0. \end{aligned}$$

Poznámka 4.5:
Jestliže se stane, že $\sum_{i=1}^n \alpha_{i\ell}^2 = 0$, pak ℓ-tý archetyp je nadbytečný, neboť platí, že

$$RSS = \sum_{i=1}^n \left\| \mathbf{x}_i - \sum_k \alpha_{ik}\mathbf{z}_k \right\|^2 = \sum_{i=1}^n \left\| \mathbf{x}_i - \sum_{k \neq \ell} \alpha_{ik}\mathbf{z}_k \right\|^2 .$$

V takovém případě můžeme nahradit tento archetyp pozorováním $\mathbf{x}_i$, pro které je v tu chvíli $\|\mathbf{x}_i - \sum_k \alpha_{ik}\mathbf{z}_k\|^2$ největší.

Jestliže $\sum_{i=1}^n \alpha_{i\ell}^2 > 0$, pak RSS definované v (4.7) lze napsat ve tvaru

$$RSS = \sum_{i=1}^n \alpha_{i\ell}^2 \|\mathbf{v}_i - \overline{\mathbf{v}}\|^2 + \sum_{i=1}^n \alpha_{i\ell}^2 \|\overline{\mathbf{v}} - \mathbf{z}_\ell\|^2. \tag{4.26}$$

Protože první člen v (4.26) nezávisí na $\mathbf{z}_\ell$, je minimalizace RSS ekvivalentní úloze

$$\min_{\{\mathbf{z}_\ell\}} \|\overline{\mathbf{v}} - \mathbf{z}_\ell\|^2 = \min_{\{\beta_{\ell j}\}} \left\| \overline{\mathbf{v}} - \sum_{j=1}^n \beta_{\ell j}\mathbf{x}_j \right\|^2$$

za podmínek $\beta_{\ell j} \geq 0$, $j = 1, \ldots, n$ a $\sum_{j=1}^n \beta_{\ell j} = 1$. Pomocí zápisu ve formě nejmenších čtverců můžeme tuto úlohu psát jako

$$\min_{\{\boldsymbol{\beta}_{\ell,\bullet}\}} \left\| \mathbf{X}^\top \boldsymbol{\beta}_{\ell,\bullet}^\top - \overline{\mathbf{v}} \right\|^2, \quad \text{za podmínek} \quad \boldsymbol{\beta}_{\ell,\bullet}^\top \geq \mathbf{0}, \quad \mathbf{1}^\top \boldsymbol{\beta}_{\ell,\bullet}^\top = 1, \tag{4.27}$$

kde $\boldsymbol{\beta}_{\ell,\bullet} = (\beta_{\ell 1}, \ldots, \beta_{\ell m})$ je ℓ-tý řádek matice $\mathbf{B}$ (braný jako řádkový vektor), $\ell = 1, \ldots, m$. Dostáváme tedy opět úlohu NNLS-ST.

Hlavní rozdíl v algoritmech na výpočet archetypů spočívá právě v odlišném způsobu řešení NNLS-ST. Úlohu NNLS-ST jsme vyřešili výše pomocí dalších úloh nejmenších čtverců, jak je naznačeno na obrázku 4.2. Nyní se zmíníme ještě o dvou

možnostech řešení. Nejprve uvedeme přístup z článku [Cutler a Breiman, 1994]. NNLS-ST převedeme na NNLS tak, že pro daná $\mathbf{u}, \mathbf{t}_1, \ldots, \mathbf{t}_q$ řešíme úlohu

$$\min_{\{w_k\}} \left\{ \left\| \mathbf{u} - \sum_{k=1}^{q} w_k \mathbf{t}_k \right\|^2 + M^2 \left\| 1 - \sum_{k=1}^{q} w_k \right\|^2 \right\} \tag{4.28}$$

za podmínky $w_k \geq 0$, $k = 1, \ldots, q$. Volbou dostatečně velkého M druhý člen v (4.28) dominuje a nutí tím při minimalizaci funkce, aby podmínka na rovnost byla přibližně splněna. Při hledání archetypů založených na této myšlence tedy k matici, která vstupuje do algoritmu NNLS, přidáme jeden řádek složený z čísel M a k vektoru pravých stran přidáme jednu dimenzi obsahující rovněž číslo M. Řešíme tedy pro každé $i = 1, \ldots, n$ podúlohu

$$\min_{\{\boldsymbol{\alpha}_{i,\bullet}\}} \left\| \widetilde{\mathbf{Z}}^\top \boldsymbol{\alpha}_{i,\bullet}^\top - \widetilde{\mathbf{x}}_i \right\|^2, \quad \text{za podmínek} \quad \boldsymbol{\alpha}_{i,\bullet}^\top \geq \mathbf{0}, \tag{4.29}$$

kde

$$\widetilde{\mathbf{Z}}^\top = \begin{pmatrix} \mathbf{Z}^\top \\ \mathbf{M}^\top \end{pmatrix}, \quad \widetilde{\mathbf{x}}_i = \begin{pmatrix} \mathbf{x}_i \\ M \end{pmatrix},$$

a pro každé $\ell = 1, \ldots, m$ podúlohu

$$\min_{\{\boldsymbol{\beta}_{\ell,\bullet}\}} \left\| \widetilde{\mathbf{X}}^\top \boldsymbol{\beta}_{\ell,\bullet}^\top - \widetilde{\overline{\mathbf{v}}} \right\|^2, \quad \text{za podmínek} \quad \boldsymbol{\beta}_{\ell,\bullet}^\top \geq \mathbf{0}, \tag{4.30}$$

kde

$$\widetilde{\mathbf{X}}^\top = \begin{pmatrix} \mathbf{X}^\top \\ \mathbf{M}^\top \end{pmatrix}, \quad \widetilde{\overline{\mathbf{v}}} = \begin{pmatrix} \overline{\mathbf{v}} \\ M \end{pmatrix}.$$

Řešení úloh (4.29) a (4.30) je časově méně náročné než výpočet NNLS-ST v úlohách (4.25) a (4.27).

Další algoritmus na řešení NNLS-ST je založený na myšlence průchodu binárním stromem. Protože však výpočetní složitost tohoto problému je exponenciální, byla nalezena ořezávací pravidla založená na Kuhn-Tuckerových podmínkách optimality s využitím konvexity účelové funkce, viz [Cutler, 1993], která dokáží uvedený postup výrazně urychlit.

Nyní již uvedeme algoritmus pro nalezení archetypů založený na myšlence střídavé minimalizace. Provedeme metodu uvedenou v [Cutler a Breiman, 1994], kde se provádí penalizovaná metoda NNLS. Jinou možností by bylo řešit úlohu NNLS-ST, což je ale výpočetně náročnější.

Do algoritmu vstupuje matice $\mathbf{X}$ typu $n \times p$, kde řádky matice odpovídají pozorováním a sloupce proměnným.

ALGORITMUS HLEDÁNÍ ARCHETYPŮ:

0. **Inicializace:**
   Určíme počet archetypů m; zpravidla $m \leq p$. Utvoříme matici $\mathbf{Z}$ typu $m \times p$, jejíž řádky budou hledané archetypy. Do matice $\mathbf{Z}$ uložíme m náhodně vybraných pozorování z matice $\mathbf{X}$.

1. **Hledání čísel $\boldsymbol{\alpha}$**:
   Pro každé $i = 1, \ldots, n$ řešíme úlohu (4.29) s $M = 10^{12}$.

2. **Hledání čísel $\boldsymbol{\beta}$**:
   Pro každé $\ell = 1, \ldots, m$ provádíme následující postup:

   (a) Jestliže $\sum_{i=1}^{n} \alpha_{i\ell}^2 = 0$, pak $\mathbf{z}_{\ell,\bullet} := \mathbf{x}_{i^*,\bullet}$, kde

   $$i^* = \underset{i=1,\ldots,n}{\operatorname{argmax}} \left\| \mathbf{x}_{i,\bullet} - \sum_{k=1}^{m} \alpha_{ik} \mathbf{z}_{k,\bullet} \right\|^2 .$$

   (b) Jestliže $\sum_{i=1}^{n} \alpha_{i\ell}^2 > 0$, pak řešíme úlohu (4.30) s $M = 10^{12}$.

3. Pro každé $k = 1, \ldots, m$ takové, že $\sum_{i=1}^{n} \alpha_{ik}^2 > 0$, vypočteme

   $$\mathbf{z}_{k,\bullet} = \sum_{j=1}^{n} \beta_{kj} \mathbf{x}_{j,\bullet}.$$

4. Vypočteme hodnotu RSS podle vzorce (4.7).

5. Jestliže snížení hodnoty RSS oproti předchozí iteraci přesahuje předem stanovenou mez, přejdeme na krok 1.

6. Algoritmus končí.

Důkaz konvergence metod založených na střídavém řešení úloh NNLS, resp. NNLS-ST je proveden v [Cutler a Breiman, 1994]. K algoritmu je třeba poznamenat, že vzhledem ke střídavé minimalizaci není zaručena konvergence ke globálnímu minimu RSS, proto je vhodné provést volbu inicializačních archetypů v kroku 0 vícekrát a ze všech voleb vybrat tu, která vede k nejnižší hodnotě RSS. Zdrojový kód algoritmu pro prostředí `R` je uveden v příloze B. Algoritmus NNLS, který je potřebný pro algoritmus ke hledání archetypů, jsme rovněž programovali, ale v průběhu psaní diplomové práce se objevil zpracovaný v podobě balíku `nnls` pro prostředí `R`. Provedli jsme srovnání s námi naprogramovanou procedurou s konstatováním, že jsme obdrželi stejné výsledky.

4.3 Ilustrační příklad – pokračování

Mezi 72 státy budeme hledat 3, resp. 6 archetypů podle výše uvedeného algoritmu. Na začátku zvolíme vždy několik náhodných inicializací, ze kterých vybereme takovou, která vede k nejnižší hodnotě RSS. Algoritmus přejde k další náhodné inicializaci (tedy k opakování postupu pro jinou inicializační matici $\mathbf{Z}$) tehdy, když rozdíl mezi nově vypočtenou a předchozí hodnotou RSS bude menší než 10^{-4}. Pro 3 archetypy dostáváme výsledky v tabulce 4.1. V dolní části tabulky jsou uvedeny státy, které mají nejblíže k vypočteným archetypům ve smyslu eukleidovské metriky.

Tabulka 4.1: Nalezené 100-násobky archetypů, $m = 3$, $RSS = 10{,}75$.

arch.	Pro	Cat	Ang	Old	Isl	Bud	Chi	Hin	Tri	Ath	Res
$100\mathbf{z}_1$	43,8	2,4	1,3	11,7	0,5	7,0	4,9	0,0	6,1	14,9	7,5
$100\mathbf{z}_2$	0,2	92,2	1,0	0,0	0,3	0,0	0,0	0,0	0,5	3,9	1,9
$100\mathbf{z}_3$	0,1	0,3	0,0	10,3	84,5	0,3	1,8	1,6	0,0	0,3	0,9

$\mathbf{z}_1 \sim$ Jižní Afrika, $\mathbf{z}_2 \sim$ Portugalsko, $\mathbf{z}_3 \sim$ Uzbekistán.

V případě 3 archetypů se nejsilněji projevilo katolické náboženství zastoupené v druhém archetypu z 92 %. Typickým státem složeným převážně z katolického obyvatelstva je například Portugalsko. Ve třetím archetypu již není tak jednoznačně zastoupeno jediné náboženství jako u druhého archetypu, ale převažuje islám doprovázený větším podílem pravoslavného náboženství. Další náboženství jsou v tomto archetypu zastoupena spíše okrajově. První archetyp nemá tak jednoznačnou skladbu jako předchozí dva, převažuje v něm protestantské náboženství v kombinaci s ateismem.

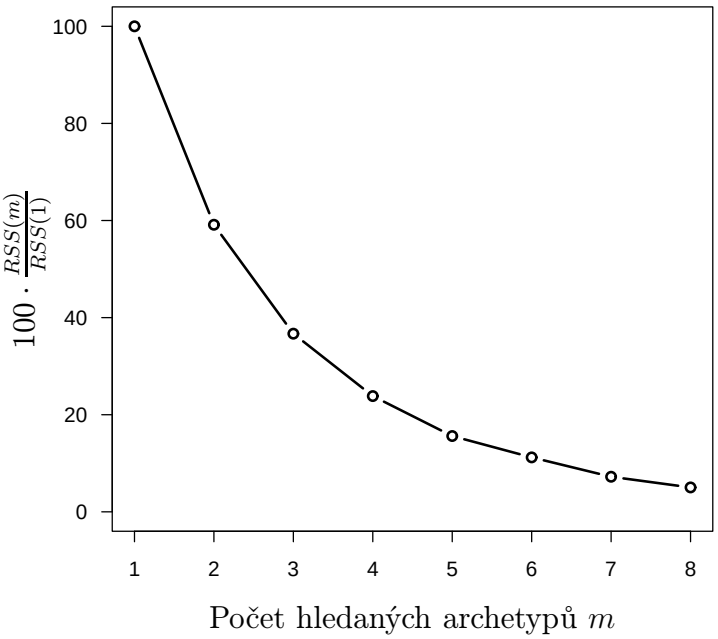
V případě 6 archetypů znázorněných v tabulce 4.2 je dominantních více různých náboženství v jednotlivých archetypech. Čtyři ze šesti uvedených archetypů mají zastoupeno právě jedno náboženství z více jak 90 %. Všimněme si, že z tabulky 4.2 lze hůře vyčíst, která náboženství jsou ve světě nejvíce rozšířená, neboť větší počet archetypů způsobil, že u více náboženství jsou vysoká čísla. Nalezené archetypy ale na druhou stranu více korespondují s původními daty ve smyslu nižší hodnoty RSS. Z tabulky 4.1 sice hůře vyčteme, které státy jsou charakteristické určitým náboženstvím, ale podle třech nejvyšších hodnot v této tabulce lze usoudit, že nejextrémněji je v některých státech zastoupeno katolické náboženství, islám a protestantské náboženství.

Na obrázku 4.3 je znázorněn procentuální pokles hodnoty RSS v závislosti na počtu archetypů m použitých k reprezentaci původních dat.

Tabulka 4.2: Nalezené 100-násobky archetypů, $m = 6$, $RSS = 3{,}48$.

arch.	Pro	Cat	Ang	Old	Isl	Bud	Chi	Hin	Tri	Ath	Res
$100\mathbf{z}_1$	0,2	0,3	0,0	0,0	95,2	0,0	1,4	2,0	0,2	0,1	0,7
$100\mathbf{z}_2$	1,7	1,3	0,1	0,0	1,0	33,9	21,8	0,4	0,0	38,3	1,4
$100\mathbf{z}_3$	14,9	5,7	1,2	0,2	3,6	0,0	0,0	0,0	27,2	0,6	46,6
$100\mathbf{z}_4$	0,1	0,4	0,0	92,6	2,0	0,0	0,0	0,0	0,0	4,7	0,1
$100\mathbf{z}_5$	0,6	94,4	1,3	0,1	0,0	0,0	0,0	0,0	0,0	3,6	0,1
$100\mathbf{z}_6$	91,6	1,4	1,8	0,1	0,1	0,0	0,0	0,0	0,0	4,8	0,1

$\mathbf{z}_1 \sim$ Pákistán, $\mathbf{z}_2 \sim$ Taiwan, $\mathbf{z}_3 \sim$ Malawi, $\mathbf{z}_4 \sim$ Řecko, $\mathbf{z}_5 \sim$ Portugalsko, $\mathbf{z}_6 \sim$ Finsko.



Obrázek 4.3: Procenta RSS pro archetypy v příkladě náboženství.

Kapitola 5

Aplikace shlukové analýzy ve výzkumu životního stylu v ČR

Některé z výše uvedených postupů budeme nyní aplikovat na data z průzkumu životního stylu v České republice, který prováděla v roce 1999 společnost TNS-AISA, s.r.o. Zúčastnilo se ho 1 327 respondentů ve věku mezi 14 a 70 lety, kteří odpovídali na 75 otázek týkajících se jejich preferencí, názorů, jejich ekonomické situace, rodiny, zaměstnání, demografie atp. Vzhledem k téměř desetiletému stáří prováděného průzkumu zanikly některé názvy používané v dotazníku (např. neexistence některých periodik či rozhlasových stanic). V datech se pokusíme najít vlastnosti, které segmentují českou společnost.

Užitečnost takové analýzy uvádí následující příklad: Firma, která uvažuje o reklamní kampani v Praze na prodej svého produktu, potřebuje vědět, čím se vyznačují potenciální kupci či zájemci o tento produkt. Na ty potom zaměří svou kampaň. V tomto případě se tedy pro shlukovou analýzu vyberou všichni respondenti žijící v Praze a provede se rozklad do několika shluků. Následně se zkoumá, jaké proměnné se zejména podílely na segmentaci dat a v jakých vlastnostech se vzájemně liší jednotlivé shluky. Na základě apriorních informací (např. o typu a ceně produktu) se vybere cílový shluk, který by svým profilem mohl nejpravděpodobněji reagovat na nabídku firmy.

5.1 Proměnné

Máme k dispozici předzpracovaná data ve formátu excelovského souboru `data.xls`, který uvažuje 64 proměnných (64 dotazů) pro všechny respondenty. Tyto proměnné byly z původních 75 vybrány expertním posudkem a na základě analýzy chybějících pozorování. Proměnné jsou charakterizovány v tabulce 5.1. Zkratky „O“, resp. „N“, resp. „B“ jsou použity pro ordinální, resp. nominální, resp. binární proměnnou. Hodnotám, které proměnné mohou nabývat, budeme říkat *znaky* nebo *vlastnosti*, neboť se jedná o charakteristiky respondenta.

Tabulka 5.1: Proměnné, jejich popis s počtem nabývajících hodnot a základní ukazatele.

č.	**proměnná**	**typ**	**popis**	ø	**s. o.**
1	sex	B	pohlaví	0,42	0,49
2	age	O	věk (57)	42,8	15,44
3	agecat	O	věková kategorie (3)	2,33	0,71
4	marst	N	rodinný stav (5)	–	–
5	chil18	B	dítě do 18 let	0,46	0,50
6	pers	B	odpovědný za nákupy, každodenní potřeby	0,46	0,49
7	red	O	nejvyšší dosažené vzdělání (4)	2,40	0,85
8	empst	N	zaměstnanecké postavení (9)	–	–
9	hhincat	O	rozmezí měsíčních příjmů (3)	1,96	0,86
10	dis	N	okres bydliště (85)	–	–
11	sizec	O	rozmezí počtu obyvatel v okrese bydliště (5)	3,03	1,42
12	q3	O	frekvence soukr. dovolených v zahraničí (5)	3,82	1,34
13	q8	O	stupně spokojenosti se souč. prací/studiem (7)	5,19	1,20
14	q12	O	jak často tráví večer mimo domov (5)	3,16	1,29
15	q13a	O	frekvence soukr. telefonátů (9)	3,56	2,13
16	q14a	O	frekvence soukr. cest do zahraničí (8)	6,65	1,54
17	q15	N	vlastnický vztah k bytu, kde bydlí (4)	–	–
18	q18a	B	soukromé (ne firemní) auto v domácnosti	0,63	0,48
19	q20	N	místo v bytě, kde tráví nejvíce volného času (9)	–	–
20	q32	O	významnost kvality oblékání a stylu (4)	2,44	0,77
21	q38	O	kouření (3)	2,48	0,84
22	q44	B	věřící nebo ateista	0,36	0,48
23	q46	B	optimista nebo pesimista	0,68	0,47
24	q47	O	porovnání finanční situace s loňským rokem (5)	3,14	0,85
25	q2a	B	sporty: lezení, vysokohorská turistika	0,11	0,31
26	q2b	B	sporty: cyklistika	0,45	0,50
27	q2c	B	sporty: běhání	0,14	0,34
28	q2d	B	sporty: plavání	0,44	0,50
29	q2e	B	sporty: fotbal	0,12	0,33
30	q2f	B	sporty: volejbal	0,11	0,31
31	q2g	B	sporty: basketbal	0,04	0,19
32	q2h	B	sporty: tenis	0,09	0,28
33	q2i	B	sporty: stolní tenis	0,11	0,32
34	q2j	B	sporty: squash	0,02	0,14
35	q2k	B	sporty: aerobik	0,19	0,39
36	q2l	B	sporty: fitness aerobik	0,11	0,31
37	q2m	B	sporty: běžky	0,14	0,35
38	q2n	B	sporty: sjezdové lyžování	0,15	0,35
39	q2o	B	sporty: bruslení	0,18	0,38
40	q2p	B	sporty: hokej	0,05	0,22
41	q2q	B	sporty: jízda na koleč. bruslích	0,03	0,16
42	q2r	B	sporty: skateboard, snowboard	0,01	0,11
43	rad3	B	rádia (poslouchaná včera): Radiožurnál	0,14	0,34
44	rad9	B	rádia (poslouchaná včera): Frekvence 1	0,13	0,33
45	rad32	B	rádia (poslouchaná včera): Impuls	0,12	0,32
46	rad6	B	rádia (poslouchaná včera): Český rozhlas-region	0,05	0,21
47	rad4	B	rádia (poslouchaná včera): Český rozhlas 2 Praha	0,04	0,21
48	rad33	B	rádia (poslouchaná včera): Blaník	0,04	0,20
49	rad8	B	rádia (poslouchaná včera): Evropa 2	0,04	0,19
50	rad25	B	rádia (poslouchaná včera): O.K.	0,03	0,18
51	rad10	B	rádia (poslouchaná včera): Kiss	0,03	0,18
52	m1a	B	noviny (čtené minulý týden): Blesk	0,50	0,50
53	m1i	B	noviny (čtené minulý týden): Mladá Fronta DNES	0,42	0,49
54	m1m	B	noviny (čtené minulý týden): Právo	0,25	0,44

55	m1s	B	noviny (čtené minulý týden): Zemské noviny	0,14	0,34
56	m1p	B	noviny (čtené minulý týden): Sport	0,13	0,33
57	m1d	B	noviny (čtené minulý týden): Hospodářské noviny	0,11	0,32
58	m1g	B	noviny (čtené minulý týden): Lidové noviny	0,11	0,32
59	m1h	B	noviny (čtené minulý týden): Metro	0,08	0,27
60	m1k	B	noviny (čtené minulý týden): Moravskoslezský den	0,06	0,25
61	m1r	B	noviny (čtené minulý týden): Večerník Praha	0,06	0,24
62	m1l	B	noviny (čtené minulý týden): Plzeňský deník	0,06	0,24
63	m1n	B	noviny (čtené minulý týden): Severočeské noviny	0,06	0,24
64	m1j	B	noviny (čtené minulý týden): Moravské noviny Rovnost	0,06	0,23

U proměnných, které nejsou binární, je vždy v závorce uvedeno, kolika hodnot můžou nabývat.

Protože proměnné jsou různého typu, budeme raději pracovat s maticí vzdáleností vypočtenou na základě koeficientu nepodobnosti (1.8). Shlukovací algoritmy pak budeme aplikovat na tuto matici nepodobností.

Před použitím shlukovacích metod jsme vypočetli dva statistické ukazatele, kterými jsou průměr (ø) a směrodatná odchylka (s. o.). Průměr a směrodatná odchylka nemají smysl počítat pro nominální data s více jak dvěma stavy, proto jsou v tabulce 5.1 příslušná místa proškrtnutá. V případě binární proměnné udává vypočtený průměr relativní četnost výskytu daného znaku. Z tabulky 5.1 lze vyčíst, že do tohoto výzkumu bylo zahrnuto 42 % mužů a 58 % žen. Průměrný věk dotazovaných byl asi 43 let.

5.2 Shlukování

Zkusíme nejprve provést shlukování bez jakékoliv předběžné analýzy datového souboru. Začneme výběrem vhodného shlukovacího algoritmu. Protože datový soubor se skládá z mnoha typů proměnných, vypočteme matici nepodobností mezi pozorováními podle koeficientu nepodobnosti (1.8). Jak již bylo řečeno, tento koeficient dokáže pracovat s různými typy proměnných a vypořádá se i s chybějícími hodnotami. Matici nepodobností získáme příkazem

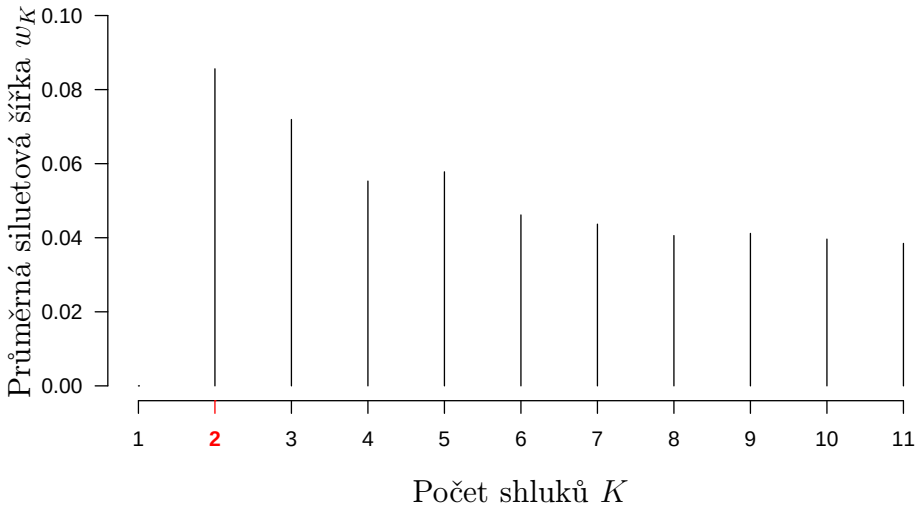
```
diff<-daisy(data,type="gower").
```

Vyzkoušíme výpočet optimálního počtu shluků pomocí siluetové funkce (2.9) a kritéria (2.11).

```
asw<-numeric(10)
for (k in 2:10) {
                 asw[k]<-pam(diff,k,diss=TRUE)$silinfo$avg.width
                }
```

Získáme výsledky na obrázku 5.1, ze kterého je vidět, že optimální počet shluků je pouze 2.

Nyní vyberme vhodný shlukovací algoritmus. Kdybychom se rozhodli použít aglomerativní shlukování, algoritmus by postupně tvořil z 1 327 shluků jediný shluk.



Obrázek 5.1: Graf k určení optimálního počtu shluků.

Nás bude zřejmě zajímat rozklad do nízkého počtu shluků, jak již naznačilo kritérium siluetové funkce. Proto by byl nalezený rozklad z velké části ovlivněn kumulujícími se chybami. Divizivní techniky mají vysokou výpočetní náročnost, proto pro analýzu tohoto velkého datového souboru nepřichází v úvahu. Metoda k-means lze aplikovat pouze na data, která jsou kvantitativní, nebo je za kvantitativní považujeme. Proto jsme se rozhodli použít algoritmus k-medoids. Vlastní shlukování provedeme pomocí procedury `pam`, kterou aplikujeme na matici nepodobností.

```
data.pam<-pam(diff,2,diss=T).
```

Dostaneme rozklad do 2 shluků s četnostmi $N_1 = 576$, $N_2 = 751$.

5.2.1 Určující proměnné

Nyní je třeba vyšetřit, které znaky, příp. proměnné se nejvýrazněji podílely na rozřazení do těchto 2 shluků. Budeme zkoumat, zda-li daný znak dostatečně charakterizuje vzniklý shluk. Uvažujme následující relativní četnosti:

$$\frac{\text{počet respondentů s daným znakem zařazených do 1. shluku}}{\text{počet respondentů s daným znakem}},$$

resp.

$$\frac{\text{počet respondentů s daným znakem zařazených do 2. shluku}}{\text{počet respondentů s daným znakem}}.$$

Tyto poměry vyjadřují, na kolik byl daný znak určující pro zařazení do jednoho či druhého shluku. Jestliže na nově dotazované osobě pozorujeme určitou vlastnost, pak výše uvedené poměry vyjadřují pravděpodobnost, že tuto osobu přiřadíme do prvního, resp. druhého shluku na základě přítomnosti daného znaku.

Po prozkoumání všech relativních četností jsme zjistili, že výraznou roli hraje věk. Všichni respondenti ve věku 16, 18 a 20 let byli zařazeni do 1. shluku. Do věku 32 let převažuje podmíněná pravděpodobnost zařazení do 1. shluku nad podmíněnou pravděpodobností zařazení do 2. shluku. Od věku 45 let je tomu naopak. Mezi 33 roky a 44 roky relativní četnosti kolísají. Odpovídá tomu i zúžení informace o věku do proměnné „věková kategorie“. V tabulce 5.2 jsou uvedeny významy jednotlivých znaků proměnných „věková kategorie“ a „rozmezí měsíčních příjmů“, relativní četnosti zařazení do obou shluků jsou uvedeny v tabulce 5.3 u proměnné `agecat`.

Tabulka 5.2: Významy proměnné „věková kategorie“ (`agecat`) a „rozmezí měsíčních příjmů“ (`hhincat`).

číslo	**význam** `agecat`
1	14 − 24 let
2	25 − 44 let
3	45 − 70 let

číslo	**význam** `hhincat`
1	podprůměrný plat
2	průměrný plat
3	nadprůměrný plat

Tabulka 5.3: Relativní četnosti v rozkladu u proměnných „věková kategorie“ (`agecat`) a „rozmezí měsíčních příjmů“ (`hhincat`).

	věk. kategorie			měs. příjem		
	1	2	3	1	2	3
shluk 1	0,84	0,52	0,24	0,42	0,33	0,52
shluk 2	0,16	0,48	0,76	0,58	0,67	0,48

Z tabulky 5.3 také vyčteme, že ze všech respondentů s průměrným platem vyšetřovaných v tomto průzkumu se jich dvě třetiny zařadilo do 2. shluku (viz proměnná `hhincat`, stav 2). Z tabulky 5.3 je dále vidět, že 52 % respondentů, kteří mají nadprůměrný plat, se zařadilo do 1. shluku. Taktéž zhruba polovina respondentů s podprůměrným platem připadla do 1. shluku a polovina do 2. shluku. Měsíční příjem občana tedy pro shlukování nebyl příliš podstatný ukazatel.

Prozkoumáme-li relativní četnosti v proměnné „rodinný stav“, zjišťujeme, že 87 % svobodných lidí bylo zařazeno do 1. shluku a stejné procento vdov a vdovců do 2. shluku. Přibližně dvě třetiny respondentů žijících v manželství bylo zařazeno do 2. shluku. Rozvedení lidé i občané žijící dlouhodobě s partnerem bez sňatku

se do obou shluků rozvrstvili víceméně rovnoměrně. Tabulku pro tuto proměnnou uvádět nebudeme.

S věkem souvisí podle průzkumu četností také proměnná „zaměstnanecké postavení“. V tabulce 5.4 jsou uvedeny významy hodnot 1 až 9 této proměnné. Údaje o četnostech můžeme vidět v tabulce 5.5.

Tabulka 5.4: Významy proměnné „zaměstnanecké postavení“.

číslo	význam
1	student
2	zaměstnanec
3	člen družstva
4	podnikatel bez zaměstnanců
5	podnikatel se zaměstnanci

číslo	význam
6	muž/žena v domácnosti
7	nezaměstnaný
8	pracující důchodce
9	nepracující důchodce

Tabulka 5.5: Relativní četnosti v rozkladu u proměnné „zaměstnanecké postavení“ (`empst`).

	zaměstnanecké postavení								
	1	2	3	4	5	6	7	8	9
shluk 1	0,91	0,55	0,27	0,57	0,70	0,15	0,40	0,15	0,08
shluk 2	0,09	0,45	0,73	0,43	0,30	0,85	0,60	0,85	0,92

Vidíme, že 91 % studentů bylo zařazeno do 1. shluku a 92 % nepracujících penzistů do 2. shluku. Přes 70 % respondentů, kteří se označili jako členové družstva, byli zařazeni do 2. shluku, naopak 70 % vyšetřovaných podnikatelů, kteří mají nějaké zaměstnance, byli zařazeni do 1. shluku. Významněji do rozkladu ještě vstoupila role muže a ženy v domácnosti a pracujících důchodců. Respondenti, kteří se takto označili, byli v 85 % případů zařazeni do 2. shluku.

Místo bydliště příliš nerozhoduje o rozřazení do vzniklých shluků. V 1. shluku je asi 60 % Pražanů a přibližně 50 % obyvatel z oblasti severozápadních Čech. Do 2. shluku se dostalo 66 % lidí z jižních Čech. V ostatních regionech je rozvrstvení do shluků rovnoměrnější.

V kategorii sportů dominují v tomto směru fotbal, basketbal, tenis, stolní tenis, squash, fitness aerobik, sjezdové lyžování, hokej, jízda na kolečkových bruslích a jízda na snowboardu a skateboardu. Tyto sporty tvoří zleva sloupce tabulky 5.6 v takovém pořadí, v jakém byly jmenovány. V tabulce 5.6 jsou pro vybraný sport vypočteny relativní četnosti respondentů, kteří vypověděli, že jej provozují. Přes 80 % respondentů, kteří označili, že provozují občas jeden z těchto sportů, bylo zařazeno do 1. shluku. V případě jízdy na skateboardu a snowboardu (sport č. 42)

Tabulka 5.6: Relativní četnosti zařazení respondentů, kteří občas provozují vybrané sporty.

	aktivně provozované sporty									
otázka č.	29	31	32	33	34	36	38	40	41	42
shluk 1	0,89	0,92	0,90	0,80	0,86	0,84	0,82	0,94	0,86	1,00
shluk 2	0,11	0,08	0,10	0,20	0,14	0,16	0,18	0,06	0,14	0,00

se dokonce všichni respondenti dostali do 1. shluku; je ale pravda, že ze všech respondentů tyto sporty občas provozovalo pouze 15 dotazovaných ($\sim$ 1 %). Sport hraje v procesu shlukování dominantní roli. Kdyby např. nový respondent označil, že aktivně hraje hokej, nemuseli bychom se ho ptát na dalších 63 otázek a jeho zařazením do 1. shluku bychom si mohli být z 94 % jisti. Na druhou stranu je třeba říci, že jestliže by respondent neprovozoval žádný ze sportů, rozhodnutí o jeho zařazení k již existujícím shlukům by zdaleka nebylo tak jednoznačné. Tabulku s relativními četnostmi respondentů, kteří neprovozují vybrané sporty uvádět nebudeme, protože z hlediska rozřazení do shluku není zajímavá.

Z rádií poslouchaných včera hrála výraznější roli Evropa 2. 83 % dotazovaných, kteří Evropu 2 poslouchali, se zařadilo do 1. shluku. 80 % respondentů, kteří poslouchali Český rozhlas 2, bylo zařazeno do 2. shluku. Rozhlasová stanice Evropa 2 hraje aktuální hudbu, která oslovuje především mladší generaci, kdežto Český rozhlas 2 vysílá především klasickou hudbu. Z průzkumu relativních četností se ukázalo, že čtenost deníků (proměnné č. 52 až 64) se výraznou měrou nepodílela na rozřazení do shluků.

5.2.2 Shrnutí

Obecně lze říci, že v procesu shlukování hrál významnou roli věk, zaměstnání a sport. Mladší sportovní jedinci bez víry, kteří studují nebo podnikají a svou budoucnost vidí více optimisticky, jsou vhodní kandidáti na zařazení spíše do 1. shluku. Starší lidé, často v penzi, kteří tráví volný čas jinak než sportem, by se spíše řadili do 2. shluku. Jsou k vývoji v ČR spíše skeptičtí, sází na stabilitu a jistoty (většina z nich má zajištěné soukromé bydlení).

Nyní se krátce zmíníme o charakterizaci shluků, tedy relativní četnosti výskytu daného znaku ve shluku vzhledem k velikosti tohoto shluku: 62 % respondentů v 1. shluku jsou muži, naopak v 2. shluku je 73 % žen. Větší rozdíl v zastoupení znaků ve shlucích je patrný už jen v proměnné „odpovědnost za nákupy“. V prvním shluku je odpovědných 39 %, ve druhém shluku je to zhruba 77 %. Další četnosti již nejsou tak rozdílné, takže vytvořené shluky nelze vzájemně příliš dobře vymezit jeden od druhého pomocí proměnných z tohoto dotazníku.

5.3 Příprava dat

Nyní se v procesu zacházení s těmito daty vrátíme zpět na začátek a budeme se je snažit nějakým způsobem upravit. Tyto úpravy jsou motivovány jednak tím, že se v datech mohou vyskytovat proměnné, které spolu úzce souvisí, a také tím, že bychom rádi vytvořili smysluplný pestřejší rozklad, který by poukázal na členitější diverzifikaci české společnosti.

5.3.1 Úprava hodnot proměnných

Zaměřme se nejprve na datový soubor jako na matici čísel. Vidíme, že všechny proměnné, až na proměnnou označující okres bydliště (`dis`), nabývají hodnot v řádu jednotek. Proměnnou `dis`, kódovanou v tisících, upravíme na nominální proměnnou, kde zohledníme níže uvedená širší území ČR, kterým budeme říkat *oblasti*. V některých případech se jedná o kraje ČR, v jiných oblastech se vyskytují rozdíly od krajů. Vzniklé seskupení zohledňuje číslování zadavatele. Jednotlivé okresy sdružíme do větších celků a přiřadíme jim následující hodnoty nominální proměnné.

1 – Praha

2 – Střední Čechy (okresy: Benešov, Beroun, Kladno, Kolín, Kutná Hora, Mělník, Mladá Boleslav, Nymburk, Praha-východ, Praha-západ, Příbram, Rakovník)

3 – Jižní Čechy (okresy: České Budějovice, Český Krumlov, Jindřichův Hradec, Pelhřimov, Písek, Prachatice, Strakonice, Tábor)

4 – Západní Čechy (okresy: Domažlice, Cheb, Karlovy Vary, Klatovy, Plzeň-město, Plzeň-jih, Plzeň-sever, Rokycany, Sokolov, Tachov)

5 – Severozápadní Čechy (okresy: Česká Lípa, Děčín, Chomutov, Jablonec nad Nisou, Liberec, Litoměřice, Louny, Most, Teplice, Ústí nad Labem)

6 – Severní a východní Čechy (okresy: Havlíčkův Brod, Hradec Králové, Chrudim, Jičín, Náchod, Pardubice, Rychnov nad Kněžnou, Semily, Svitavy, Trutnov, Ústí nad Orlicí)

7 – Střední a jižní Morava (okresy: Blansko, Brno-město, Brno-venkov, Břeclav, Zlín, Hodonín, Jihlava, Kroměříž, Prostějov, Třebíč, Uherské Hradiště, Vyškov, Znojmo, Žďár nad Sázavou)

8 – Severovýchodní Morava (okresy: Bruntál, Frýdek Místek, Karviná, Nový Jičín, Olomouc, Opava, Ostrava-město, Přerov, Šumperk, Vsetín)

5.3.2 Testy nezávislosti

V datovém souboru se zaměříme na to, zda existuje nějaká závislost mezi proměnnými. Vzhledem k tomu, že všechny proměnné jsou diskrétní povahy (nominální nebo ordinální), budeme předpokládat, že pochází z diskrétních rozdělení. K testování jejich nezávislosti použijeme test nezávislosti v kontingenčních tabulkách. V tabulce 5.7 je znázorněna kontingenční tabulka pro 2 proměnné – jedna nabývá hodnot $1, \ldots, r$, druhá nabývá hodnot $1, \ldots, c$. Po pravé straně tabulky, resp. dole jsou uvedené řádkové, resp. sloupcové součty v každé hodnotě této proměnné.

Tabulka 5.7: Kontingenční tabulka.

	proměnná 2			
proměnná 1	1	...	c	$\sum$
1	n_{11}	...	n_{1c}	$n_{1,\bullet}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
r	n_{r1}	...	n_{rc}	$n_{r,\bullet}$
$\sum$	$n_{\bullet,1}$	...	$n_{\bullet,c}$	n

K testování nezávislosti těchto dvou proměnných použijeme testovou statistiku

$$\chi^2 = n \sum_{i=1}^{r} \sum_{j=1}^{c} \left(\frac{n_{ij}^2}{n_{i,\bullet} n_{\bullet,j}} \right) - n.$$

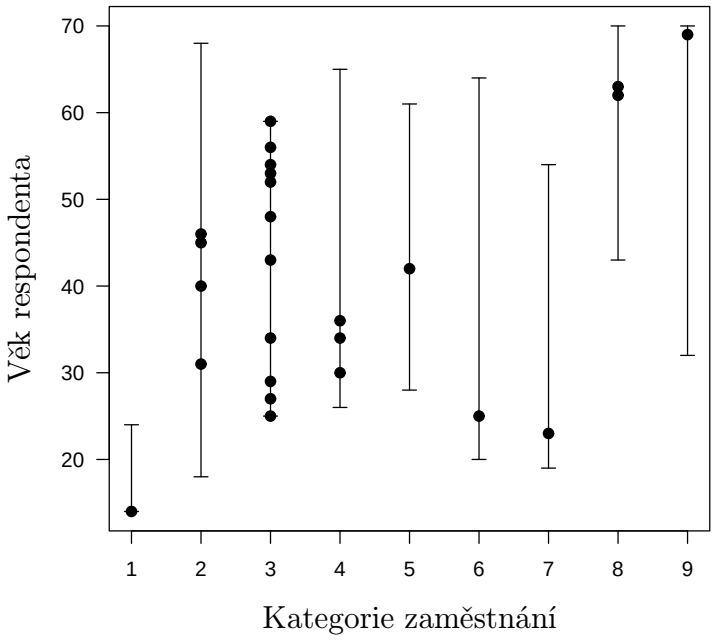
Za předpokladu, že

$$\frac{n_{i,\bullet} \cdot n_{\bullet,j}}{n} > 5, \quad \text{pro všechna} \quad i = 1, \ldots, r, \;\; j = 1, \ldots, c, \tag{5.1}$$

má podle [Anděl, 2005] statistika χ^2 asymptoticky rozdělení χ^2 s $(r-1)(c-1)$ stupni volnosti.

Zkoumáním jednotlivých kontingenčních tabulek jsme objevili závislosti mezi nejrůznějšími typy proměnných. Dobře patrná závislost byla zjištěna mezi proměnnými „zaměstnanecké postavení“ (`empst`) a „věk“ (`age`). Na obrázku 5.2 vidíme pro každou kategorii zaměstnání rozmezí věku respondentů. Černou tečkou je pak vyznačen věk, ve kterém respondent nejčastěji volil příslušnou variantu povolání. Z obrázku 5.2 je patrné, že zaměstnanci se ve vybraném vzorku populace objevují napříč věkovým spektrem. Nejužší věkové rozpětí mají studenti, protože omezení je dáno ze zákona dovršením 26 let.

Silná závislost vyšla mezi proměnnou „odpovědnost za nákupy“ (`pers`) a „pohlaví“ (`sex`). Absolutní četnosti v těchto případech jsou uvedeny v tabulce 5.8, ve které provedeme test nezávislosti. Budeme testovat hypotézu H_0, že binární proměnné „pohlaví“ a „odpovědnost za nákupy“ jsou nezávislé znaky. Vypočteme



Obrázek 5.2: Závislost věku na typu povolání.

Tabulka 5.8: Absolutní četnost proměnných „odpovědný za nákupy“ (`pers`) a „pohlaví“ (`sex`).

			`sex`		
			žena	muž	$\sum$
			0	1	
`pers`	ne	0	89	434	523
	ano	1	681	122	803
	$\sum$		770	556	1326

hodnotu statistiky

$$\chi^2 = n \sum_{i=1}^{2} \sum_{j=1}^{2} \left(\frac{n_{ij}^2}{n_{i,\bullet} n_{\bullet,j}} \right) - n,$$

kde n_{ij} je četnost případů, kdy se ve výběru vyskytly současně znaky (i, j), $n_{i,\bullet}$ je součet v i-tém řádku kontingenční tabulky a $n_{\bullet,j}$ je součet v j-tém sloupci kontingenční tabulky, $i = 1, 2$, $j = 1, 2$. Vyjde $\chi^2 \doteq 597{,}76$. Je splněn předpoklad (5.1), takže podle [Anděl, 2005] má statistika χ^2 asymptoticky rozdělení χ^2 s jedním

stupněm volnosti (označme toto rozdělení χ^2_1). Označme $\chi^2_1(\alpha)$ kritickou hodnotu rozdělení χ^2_1 na hladině α. Jedná se o číslo, které náhodná veličina s rozdělením χ^2_1 překročí s pravděpodobností α. Protože

$$597{,}76 = \chi^2 > \chi^2_1(0{,}05) = 3{,}84,$$

zamítáme hypotézu H_0 o nezávislosti proměnných „pohlaví“ a „odpovědnost za nákupy“ na hladině 5 %.

Následující tabulka 5.9 se zaměřuje na některé další případy, kdy jsme zamítali hypotézu nezávislosti pro 2 vybrané proměnné na hladině 5 %. Ve všech testech byl splněn předpoklad (5.1).

Tabulka 5.9: Testy nezávislosti na hladině 5 % v kontingenčních tabulkách.

popis proměnných		**st. volnosti** f	χ^2	$\chi^2_f(0,05)$
věk. kategorie	rodinný stav	$2 \cdot 4 = 8$	634,86	15,51
věk. kategorie	dítě do 18-ti let	$2 \cdot 1 = 2$	388,01	5,99
rod. stav	odpov. za nákupy	$4 \cdot 1 = 4$	180,01	9,49
rod. stav	měsíční příjem	$4 \cdot 2 = 8$	128,00	15,51
oblast	poč. obyv. u byd.	$7 \cdot 4 = 28$	579,12	41,34

Detailnější analýzou kontingenčních tabulek pro dvojice proměnných z tabulky 5.9 zjistíme, že mezi mladými lidmi do 24 let převládají svobodní, naproti tomu ve střední a starší věkové generaci jsou nejčastěji zastoupeny oddané páry. Ve střední generaci je podle očekávání největší výskyt rodin s alespoň jedním dítětem. Mezi svobodnými lidmi vyšetřovanými v průzkumu převládají ti, co nejsou zodpovědní za chod domácnosti a kteří mají podprůměrný příjem. Může se jednat o mladé studenty žijící dohromady se svými rodiči, kteří si případně pouze přivydělávají. Ve všech dalších kategoriích (oddaní, rozvedení, žijící dlouhodobě s partnerem bez sňatku a vdovy a vdovci) převládají odpovědní za chod domácnosti. Rozvedení lidé a vdovy a vdovci mají převážně podprůměrný příjem. V kategorii lidí žijících dlouhodobě s partnerem a lidí oddaných má většina respondentů nadprůměrný příjem. Tito lidé se totiž kromě sami sebe musí starat ještě o své nejbližší a k tomu musí mít dostatek finančních prostředků.

Nyní prozkoumáme závislost mezi vybraným celostátním deníkem (celkem je jich 7; jedná se o proměnné označené v tabulce 5.1 čísly 52 až 58) a oblastmi. Výsledky obdržíme v tabulce 5.10. Vzhledem k tomu, že proměnné týkající se deníků jsou binární proměnné a máme 8 regionů, je počet stupňů volnosti v každé z kontingenčních tabulek roven $f = 7$. Výsledky porovnáme s kritickou hodnotou $\chi^2_7(\alpha) = 14{,}07$. Poznamenejme opět, že byl splněn předpoklad (5.1).

Vidíme, že na zvolené 5% hladině můžeme zamítnout hypotézu o nezávislosti každého celostátního deníku na regionu.

Tabulka 5.10: χ^2-statistiky v testu nezávislosti mezi regiony a jednotlivými celostátními deníky.

č. otázky	**52**	**53**	**54**	**55**	**56**	**57**	**58**
χ^2	28,52	37,48	42,84	39,01	16,52	19,17	26,32

5.3.3 Chybějící pozorování

V dotaznících podobného typu se můžeme setkat s neúplnými daty. Respondent buď na danou otázku nemá jednoznačnou odpověď, nebo na ni z nějakého důvodu nechce reagovat, proto neoznačí žádnou z nabízených variant. V našem datovém souboru se vyskytuje 602 chybějících hodnot. Počty neúplných odpovědí jsou pro jednotlivé proměnné uvedeny v tabulce 5.11.

Tabulka 5.11: Počty chybějících odpovědí.

proměnná	**chybí**	**chybí** [%]	**proměnná**	**chybí**	**chybí** [%]
rodinný stav	2	0,2	soukr. telefonáty	4	0,3
dítě do 18-ti let	235	17,7	soukr. cesty	4	0,3
odpov. za nákupy	1	0,1	majet. vztah (byt)	35	2,6
dosažené vzdělání	5	0,4	oblíb. místo v bytě	40	3,0
okres bydliště	8	0,6	důlež. oblékání	11	0,8
poč. obyv. u byd.	2	0,2	kouření	8	0,6
soukr. dovolené	8	0,6	věřící/ateista	81	6,1
spokojenost s prací	29	2,2	optimista/pesimista	103	7,8
večer. kultur. akce	1	0,1	porov. fin. situace	25	1,9

Nejčastěji (ve 235 případech) respondent neodpověděl na otázku, zda má dítě mladší 18 let a 103 respondentů odmítalo odpovědět na otázku, zda jsou spíše optimisté nebo pesimisté. 81 dotazovaných se nechtělo vyjádřit k otázce víry. Chybějící hodnoty ve všech proměnných jsme se rozhodli doplnit způsobem doporučovaným v [Jain a Dubes, 1988]. Pro každou z uvedených proměnných v tabulce 5.11 provedeme následující postup:

1. Pro každé pozorování, které má v této proměnné chybějící hodnotu, nalezneme P nejbližších pozorování ve smyslu Gowerova koeficientu nepodobnosti (1.8).

2. V těchto „sousedech“ spočítáme četnost výskytu všech znaků v dané proměnné.

3. Vybereme znak s nejvyšší četností a ten doplníme místo chybějící hodnoty.

V našem případě jsme volili $P = 53$ (cca 4 % dat). Abychom se při doplňování chybějících hodnot nedopouštěli kumulujících chyb, provedeme doplnění znaku každé proměnné nezávisle na sobě. To znamená, že doplňujeme-li například hodnotu proměnné „optimista/pesimista“, pak nejbližší „sousedé“ mají ve všech ostatních proměnných hodnoty z původního datového souboru.

5.3.4 Metoda hlavních komponent

Po zjištění, že u některých dvojic proměnných jsme zamítali hypotézu nezávislosti, se nyní nabízí otázka, zda nelze datový soubor zredukovat bez výraznější ztráty původní variability. Přitom budeme vycházet z matice dat $\mathbf{X}$, ve které byly agregovány znaky proměnné „okres bydliště“ do oblastí a dopočteny chybějící hodnoty podle postupu v předešlé části. Jako nástroj k zodpovězení otázky vzájemné propojenosti proměnných použijeme metodu hlavních komponent, viz [Hebák a kol., 2005], kap. 18. Tato statistická metoda vyžaduje, aby data byla kvantitativního charakteru nebo alespoň ordinální. Vzhledem k tomu, že v datech od zadavatele jsou i nominální proměnné, rozhodli jsme se z analýzy komponent vyjmout všechny nominální proměnné s více jak dvěma stavy (tj. proměnné „rodinný stav“, „zaměstnanecké postavení“, „oblast bydliště“, „vlastnický vztah k bytu“ a „oblíbené místo v bytě“). Na binární proměnné budeme pohlížet jako na spojité proměnné.[1] Po odstranění nominálních proměnných s více jak dvěma stavy pracujeme s maticí $\mathbf{Y}$. Při analýze hlavních komponent budeme vycházet z výběrové korelační matice, protože máme k dispozici data s různými typy proměnných. Počet komponent r uvažovaných pro analýzu stanovíme jako počet vlastních čísel matice $\mathsf{cor}\,\mathbf{Y}$ větších nebo rovných jedné. Vyjde $r = 19$. Pro následnou shlukovou analýzu uvažujme prvních 19 vypočtených hlavních komponent, ke kterým přidáme 5 nominálních proměnných, které jsme nepoužili v komponentní analýze. Takto vzniklou matici označíme jako $\mathbf{U}$. Přidané nominální proměnné ještě příslušně označíme jako faktory. Zdrojový kód výše uvedeného postupu je na následujících řádcích.

```
set.nominal<-c(4,8,10,17,19)
cisla<-eigen(cor(Y))$values
r<-length(cisla[cisla>=1])
skore<-princomp(Y,cor=TRUE,scores=TRUE)$scores[,1:r]
p1<-length(set.nominal)+r
U<-matrix(0,nrow=n,ncol=p1)
U[1:n,1:r]<-skore
U[1:n,(r+1):p1]<-X[,set.nominal]
for (j in (r+1):p1) {U[,j]<-factor(U[,j])}
```

Nyní vypočteme matici nepodobností na základě kritéria (2.11). Jako optimální vyjdou opět pouze 2 shluky, tentokráte s četnostmi $N_1 = 863$, $N_2 = 464$. Přestože kritérium (2.11) doporučilo provést rozklad do dvou shluků, vyzkoušíme pomocí

[1] Tento postup není úplně korektní, ale v praxi se při analýze dat často používá.

procedury `pam` i rozklad do třech shluků. Provedeme obdobnou analýzu jako při shlukování bez úpravy dat – pokusíme se pomocí relativních četností

$$\frac{\text{počet respondentů s daným znakem zařazených do } i\text{-tého shluku}}{\text{počet respondentů s daným znakem}},\ i = 1, 2, 3$$

najít znaky, které určovaly rozřazení do těchto shluků. Do následujícího odstavce shrneme pouze některé významnější výsledky, nebudeme již podrobně uvádět tabulky s relativními četnostmi.

Zjišťujeme, že jeden z vytvořených shluků je velmi obtížně identifikovatelný pomocí proměnných z dotazníku, protože relativní četnosti pro téměř všechny znaky všech proměnných vyšly pro tento shluk menší než 50 %. Pro tento shluk jsou vyšší relativní četnosti ve znacích 7 a 8 proměnné „bydliště“ (jedná se o obyvatele Moravy a Slezska), které se pohybují okolo 40 %. 56 % respondentů, kteří bydlí v podnájmu a 52 % dotázaných, kteří čtou deník Moravskoslezský den, byly zařazeny do tohoto shluku. Zbylé dva shluky se z hlediska vypočtených četností formovaly čitelněji. Významnou roli hrál opět věk. Do jednoho z nich bylo zařazeno 97 % nepracujících důchodců, 85 % pracujících důchodců, 78 % vdovců a vdov. Do třetího z vytvořených shluků se zařadila většina mladých lidí: 74 % občanů do 24 let, 79 % studentů a 76 % těch, kteří svou životní situaci považují za lepší než před rokem.

5.4 Analýza archetypů

Pokusíme se ještě v datech nalézt typické rysy pro českou společnost a pro daná data nalezneme dva a tři archetypy. Protože v archetypální analýze pracujeme s eukleidovskou normou, vyloučíme z dat všechny nominální proměnné s více jak dvěma znaky a binární proměnné budeme opět považovat za spojité proměnné. V takové matici potom budeme hledat archetypy. Při aplikaci archetypálního algoritmu se nyní potýkáme s velkým rozměrem matice dat (1327×59), tudíž je potřeba, aby počet náhodných inicializací byl dostatečně velký (alespoň v řádu stovek). Vzhledem k časové náročnosti výpočtu jsme při hledání 2 archetypů volili pouze 20 náhodných inicializací s výší tolerance na pokles RSS o velikosti 0,01. Pro každou náhodnou inicializaci algoritmus potřeboval zhruba 40 iterací a ve všech případech zkonvergoval k hodnotě $RSS \sim 29\,566$. Výsledné archetypy jsou uvedené v tabulce 5.12. Proměnné o sportech, poslouchanosti rádií a čtenosti deníků nebudeme do tabulky zahrnovat. Důvodem je zejména to, že dosažené výsledky nepoukazují na žádný význačný rys v těchto proměnných.

Na první pohled je z tabulky 5.12 zřejmé, že vytvořené archetypy poukazují na dvě extrémní věkové skupiny. První archetyp reprezentuje mladou spíše dívčí generaci z menších měst a vesnic (nízká hodnota v proměnné `sizec`). Jedná se o lidi jezdící zpravidla jedenkrát nebo dvakrát ročně do zahraničí (viz hodnota 3,5 v proměnné `q3`), což vzhledem k jejich věku může být typická dovolená s rodiči k moři nebo na hory. Jejími dalšími znaky je spíše optimistický postoj k životu (vysoká

Tabulka 5.12: Hodnoty nalezených archetypů pro některé proměnné, zaokrouhleno na desetiny. $RSS = 29\,565{,}9$

prom.	sex	age	agecat	chil18	pers	red	hhincat	sizec
arch1	0,3	16,4	1,0	0,8	0,2	2,1	2,1	2,9
arch2	0,4	68,9	3,0	0,1	0,8	2,5	1,8	3,1

prom.	q3	q8	q12	q13a	q14a	q18a	q32	q38	q44	q46	q47
arch1	3,5	4,9	2,6	3,0	6,3	0,7	2,3	2,6	0,1	0,9	2,9
arch2	4,1	5,5	3,8	4,2	7,0	0,6	2,6	2,5	0,5	0,5	3,3

hodnota 0,9 v proměnné q46) a žádná příslušnost k církvím (viz nízká hodnota 0,1 v proměnné q44). Preferují sport (většina provozuje cyklistiku a plavání), před poslechem rádia a mezi oblíbené celostátní noviny patří bulvární deník Blesk. Poslední 3 jmenované znaky nejsou přímo typické, nicméně zastoupení v těchto proměnných má archetyp výrazně vyšší než ten druhý. Tento první archetyp je zčásti obtížně interpretovatelný, protože z tabulky 5.12 například vidíme, že je to generace mající z velké části dítě mladší 18 let, což v případě šestnáctiletých jedinců není příliš realistický výsledek. Chyba může být způsobena v malém počtu inicializací pro velký datový soubor a díky tomu, že považujeme hodnoty binární proměnné jako spojité. Zkušenosti také ukazují, že hodně dětí záměrně vyplňuje do dotazníků nesmyslné údaje a archetypy jsou na odlehlá pozorování velmi citlivé, neboť leží na hranici konvexní množiny.

Druhý archetyp představuje českého seniora. Je středoškolsky vzdělaný, žije v obcích do 100 obyvatel, vyznačuje se nechutí cestovat jak do zahraničí (vysoká hodnota 4,1 v proměnné q3), tak i třeba za svými nejbližšími po ČR. Svou životní situaci hodnotí jako rok od roku spíše horší. Z hodnoty 0,5 proměnné q44 vyplývá, že optimistický i pesimistický prvek je rovnoměrně zastoupen.

Vidíme, že oba nalezené archetypy lze označit jako protipóly jen vzhledem k věku a odpovědnosti za nákupy. V ostatních proměnných jsou sice ve vypočtených archetypech rozdíly, ale ne tak výrazné. Tento výsledek koresponduje s charakteristikami shluků v případě shlukování bez předběžných úprav, kde jsme přisoudili mimo jiné důležitost proměnným „věk“ a „odpovědnost za nákupy“.

Pro hledání třech archetypů jsme volili toleranci pro pokles RSS rovněž 0,01. Algoritmus potřeboval ke konvergenci zhruba 160 iterací. Z 20 případů zkonvergoval dvanáctkrát k hodnotě $RSS \sim 24\,311$, osmkrát k hodnotě $RSS \sim 24\,732$. Výsledky jsou zobrazeny v tabulce 5.13. Ze stejného důvodu jako v případě hledání 2 archetypů nebudeme uvádět proměnné o sportech, poslouchanosti rádií a čtenosti deníků.

Hodnoty 1. a 3. archetypu blízké 0,5 u proměnné „pohlaví“ interpretujeme tak, že 2 rysy české společnosti obsahují přibližně stejné množství mužského i ženského prvku.

Tabulka 5.13: Hodnoty nalezených archetypů pro některé proměnné, zaokrouhleno na desetiny. $RSS = 24\,310{,}8$

prom.	sex	age	agecat	chil18	pers	red	hhincat	sizec
arch1	0,5	69,2	3,0	0,0	0,5	1,6	1,6	2,2
arch2	0,3	16,0	1,0	0,8	0,2	2,0	2,1	2,9
arch3	0,5	66,9	3,0	0,1	0,7	3,2	1,7	3,8

prom.	q3	q8	q12	q13a	q14a	q18a	q32	q38	q44	q46	q47
arch1	5,0	5,2	4,6	8,5	8,0	0,2	3,0	2,5	0,7	0,4	3,5
arch2	3,8	4,8	2,6	3,1	6,6	0,7	2,3	2,6	0,1	0,9	3,0
arch3	2,8	5,8	3,1	1,1	5,5	0,6	2,4	2,3	0,5	0,8	3,1

Prvním archetypem může být typický český senior ve věku přibližně 69 let, s převážně středoškolským vzděláním, pocházející z drobné vesnice do 20 obyvatel (viz hodnota 2,2 u proměnné `sizec`), který drtivou většinu svého času tráví v okolí svého příbytku (hodnota 5 u proměnné `q3` vyjadřuje, že soukromě vůbec nejezdí do zahraničí a hodnota 4,6 u proměnné `q12` vyjadřuje, že téměř veškerý volný čas tráví doma). Je spíše spokojený se současným stavem, k životu příliš nepotřebuje telefon (vysoká hodnota 8,5 u proměnné `q13a`) a většinou nevlastní osobní automobil (viz hodnota 0,2 v proměnné `q18a`). Kvalitu oblékání považuje za podřadnou záležitost a vzhledem ke svému věku již nesportuje.

Druhý archetyp se ve svých hodnotách téměř shoduje s vypočteným prvním archetypem z tabulky 5.12, proto ho už znovu nebudeme detailněji popisovat. Pouze dodáme, že v proměnných týkajících se sportů má tento archetyp ze všech ostatních nejvyšší hodnoty, žádný sport však pro něj není typický.

Třetí archetyp představuje starší českou populaci krátce po dosažení důchodového věku se středoškolským až vysokoškolským vzděláním, které ještě nevyprchala životní energie. Pochází z větších vesnic, je obecně spokojená se svým životním stylem, kterému přikládá i patřičnou důležitost (hodnoty v proměnných `q8` a `q32`). Ze všech předchozích archetypů si tito občané nejčastěji dopřejí dovolenou zahraničí – v průměru dvakrát za rok. Tato generace se ze všech tří předchozích rovněž nejvíce zajímá o dění v České republice i ve světě, neboť poslouchá zpravodajsky orientované rozhlasové stanice (Český rozhlas 1 – Radiožurnál) a čte celostátní deníky – zejména Mladou Frontu Dnes. Provozuje fyzicky nenáročné a pro ni dostupné sporty jako plavání (hodnota v archetypu: 0,44) a cyklistiku (hodnota v archetypu: 0,53).

Zhodnocení výzkumu životního stylu provedeme v závěru této diplomové práce. Na tomto místě pouze dodejme, že pro účely interpretace některých závislostí mezi proměnnými a lepšímu pochopení datového souboru lze použít také tzv. asociační pravidla (*association rules*) popsaná v [Tibshirani a kol., 2001], str. 439. Pro prostředí `R` vznikl začátkem dubna 2008 balík `arules` se známým algoritmem `apriori`

Christiana Borgelta, který v datových souborech dokáže mezi znaky proměnných najít logické souvislosti.

Závěr

V této práci jsme se zaměřili na popis některých shlukovacích metod, které se používají k segmentaci datových souborů. Z každé třídy shlukovacích algoritmů jsme vybrali některé zástupce, pojednali jsme o možnostech jejich použití včetně výhod a nevýhod a provedli aplikace na konkrétní data.

Zdaleka jsme přitom nevyčerpali všechny přístupy k segmentaci databází, neboť takových algoritmů existuje velké množství. Vznikají jednak nové postupy a pak také modifikace již existujících algoritmů.

Nové algoritmy většinou vycházejí z požadavků na konkrétní data, jsou postupně zobecňovány do podoby, kdy je lze použít jako univerzální shlukovací nástroje. Jako ilustraci můžeme uvést tzv. model pokrytí (*covering model*), kdy minimalizujeme počet reprezentativních objektů tak, aby každé pozorování mělo od nejbližšího reprezentanta nejvýše předepsanou vzdálenost D. Tyto nové algoritmy se pak vyjádří v řeči minimalizační nebo maximalizační úlohy s okrajovými podmínkami a k jejich řešení existuje vybudovaná teorie optimalizace.

Úpravy již existujících algoritmů jsou motivovány snižováním jejich výpočetní náročnosti a uzpůsobením pro velké datové soubory čítající milióny pozorování. Příkladem takové úpravy algoritmu může být MacQueenova modifikace algoritmu k-means, kde se přepočet centroidů provede po každém individuálním přesunu objektu do jiného shluku.

Mezi tak velkým množstvím algoritmů tedy přirozeně vyvstává potřeba nějakým způsobem změřit kvalitu těchto algoritmů při aplikaci na konkrétní data. Neexistuje pouze jeden správný algoritmus, a proto uživatel zkouší více různých segmentačních nástrojů, z nichž si pak chce vybrat ten nejvhodnější. Cílem je tedy zjistit, jak dobře shlukovací procedura provedla svůj úkol – tedy jak kvalitní je vytvořený rozklad z hlediska možné interpretace a četností pozorování v jednotlivých segmentech v porovnání s jinými metodami. Úspěšnost každého algoritmu totiž často závisí na konkrétní povaze dat.

Pro srovnání algoritmů shlukové analýzy byly navrženy nejrůznější funkcionály kvality rozkladu, z nichž některé jsme v práci vypočetli v závěru kapitoly 3 pro různé algoritmické postupy. Zjistili jsme například, že zvolíme-li počet segmentů, může se ukázat jako nejvhodnější z hlediska hodnoty určitého funkcionálu kvality rozkladu metoda A a rozhodneme-li o změně počtu segmentů, ukáže se jako nejvhodnější při volbě stejného funkcionálu jiná metoda B. Změníme-li funkcionál kvality rozkladu nebo provedeme-li standardizaci dat, můžeme dostat opět jiné výsledky. Zá-

jemce o podrobnější informace o kvalitě shlukování odkazujeme na literaturu [Jain a Dubes, 1988], kde se celá kapitola 4 věnuje právě posuzování kvality vytvořených shluků.

Kapitola 4 v této diplomové práci představuje drobnou odbočku, neboť archetypální analýza se neřadí do klasických segmentačních metod. Nicméně jak bylo ukázáno v úvodní motivaci této kapitoly, archetypy mají vztah k nehierarchickým metodám shlukové analýzy díky přítomnosti reprezentativních objektů. Oba přístupy se liší v interpretaci. Zatímco centroidy a medoidy vytvořené pomocí nehierarchických metod vyjadřují průměrné vlastnosti vytvořeného segmentu, archetypy odráží typické rysy nebo také extrémní vlastnosti v množině dat. Dále jsme ukázali, že k nalezení archetypů lze využít teorii kvadratického programování aplikovanou na problémy nejmenších čtverců.

Poslední kapitola 5 uvádí reálný příklad, který je vhodný pro aplikaci shlukové analýzy. Přistoupili jsme k němu dvěma odlišnými způsoby, když jsme nejprve shlukovali bez jakéhokoliv zásahu do dat a poté jsme s daty provedli některé úpravy. V těchto zásazích jsme vyzkoušeli agregování proměnné do interpretačně vhodnější podoby a poté jsme provedli analýzu chybějících pozorování. V té jsme byli nuceni doplňovat chybějící hodnoty, neboť při jejich vyškrtnutí bychom ztratili nepřípustně mnoho dat. Ukázali jsme, že k doplnění chybějících hodnot lze využít i shlukovou analýzu. Po analýze závislostí proměnných v datech jsme využili další nástroj mnohorozměrné statistiky – analýzu hlavních komponent – ve které jsme úspěšně zredukovali upravený datový soubor do menšího počtu proměnných. Ukázalo se, že tyto úpravy vedly ke stejnému doporučení volit rozklad pouze do dvou shluků. Po provedení rozkladu se ukázalo, že vzniklé segmentace jsou více nesymetrické než při shlukování dat bez předběžných úprav. Při rozkladu do třech shluků jsme se potýkali s obtížnou charakterizací jednoho z vytvořených shluků. Analýza archetypů poskytla interpretačně kvalitnější výsledky. Potvrdil se při ní význam proměnných související s věkem respondentů, odpovědností za svůj vlastní život a také ochotou cestovat.

Příloha A

Posouzení kvality algoritmů

V následující části je uveden zdrojový kód pro vypočtení hodnot uvedených v tabulce 3.2. Jedná se o procedury, které pro danou hodnotu L, která udává požadovaný počet vytvořených shluků, vypočtou číselné hodnoty navržených funkcionálů rozkladu.

```
## n ..... počet pozorování
## L ..... celkový počet shluků
## dat .... datová matice - řádky tvoří pozorování, sloupce proměnné
## metrika. funkce k výpočtu vzdáleností mezi objekty

##################
### 1. K-MEANS ###
##################
kmeans.stat<-function(L,dat) {
n<-dim(dat)[1]
p<-dim(dat)[2]
x.prum<-1/n*apply(dat,2,sum)
WSS.kmeans<-numeric(1)
SIL.kmeans<-numeric(1)
MSS.kmeans<-numeric(1)

#ROZKLAD#
shl<-kmeans(dat,L,iter.max=100,nstart=100,algorithm="Hartigan-Wong")

#WSS#
WSS.kmeans<-sum(shl$withinss)

#SIL#
a<-vector(mode="numeric",length=n)
b<-vector(mode="numeric",length=n)
s<-vector(mode="numeric",length=n)
d<-matrix(0,nrow=n,ncol=L)
```

```
for (i in 1:n) {
    if (shl$size[shl$cluster[i]]==1) {s[i]<-0}
    else {
        soucet<-0
        for (j in 1:n) {
          if(shl$cluster[j]==shl$cluster[i]) {
               soucet<-soucet+sum((dat[i,]-dat[j,])^2)
                                              }
                      }
        a[i]<-1/(shl$size[shl$cluster[i]]-1)*soucet

        for (k in 1:L) {
            soucet<-0
            for (j in 1:n) {
            if ((shl$cluster[j]==k)&!(shl$cluster[j]==shl$cluster[i])) {
                 soucet<-soucet+sum((dat[i,]-dat[j,])^2)
                                                                      }
                             }
              if (shl$cluster[i]==k) {d[i,k]<-Inf}
              else {
                   d[i,k]<-1/shl$size[k]*soucet
                   }
                     } #for (k in 1:L)
       b[i]<-min(d[i,1:L])
       s[i]<-(b[i]-a[i])/max(a[i],b[i])
     }} #else #for (i in 1:n)
SIL.kmeans<-sum(s)/n

#MSS#
med<-vector(mode="numeric",length=L)
WM<-vector(mode="numeric",length=L)

for (k in 1:L) {
    nk<-shl$size[k]
    if (nk>1) {
        mat.D<-dat[shl$cluster==k,]
        vzd<-vector(mode="numeric",length=nk)
        for (i in 1:nk) {
            for (j in 1:nk) {
                vzd[i]<-vzd[i]+sum((mat.D[i,]-mat.D[j,])^2)
                             }
            vzd[i]<-vzd[i]/(nk-1)
                        }
        med[k]<-which.min(vzd)
        for (i in 1:nk) {
            WM[k]<-WM[k]+sum((mat.D[i,]-mat.D[med[k],])^2)
```

```
                        }
                } #if (nk>1)
                 } #for (k in 1:L)
MSS.kmeans<-sum(WM)

return(c(WSS.kmeans,SIL.kmeans,MSS.kmeans))
}

####################
### 2. K-MEDOIDS ###
####################
kmedoids.stat<-function(L,dat,metrika) {
n<-dim(dat)[1]
p<-dim(dat)[2]
x.prum<-1/n*apply(dat,2,sum)
WSS.kmedoids<-numeric(1)
SIL.kmedoids<-numeric(1)
MSS.kmedoids<-numeric(1)

#ROZKLAD#
shl<-pam(dat,L,diss=FALSE,metric=metrika,stand=FALSE)

#WSS#
prum<-vector(mode="numeric",length=p)
WK<-vector(mode="numeric",length=L)

for (k in 1:L) {
    mat.D<-dat[shl$clustering==k,]
    nk<-shl$clusinfo[k,1]
    if (nk>1) {
        prum<-1/nk*apply(mat.D,2,sum)
        for (i in 1:nk) {
        WK[k]<-WK[k]+sum((mat.D[i,]-prum)^2)
                        }
              } #if (nk>1)
               } #for (k in 1:L)
WSS.kmedoids<-sum(WK)

#SIL#
SIL.kmedoids<-shl$silinfo$avg.width

#MSS#
WM<-vector(mode="numeric",length=L)

for (k in 1:L) {
    mat.D<-dat[shl$clustering==k,]
```

```
    nk<-shl$clusinfo[k,1]
    if (nk>1) {
        for (i in 1:nk) {
            WM[k]<-WM[k]+sum((mat.D[i,]-shl$medoids[k,])^2)
                        }
          } #if (nk>1)
           }
MSS.kmedoids<-sum(WM)

return(c(WSS.kmedoids,SIL.kmedoids,MSS.kmedoids))
}

################
### 3. FUZZY ###
################
fuzzy.stat<-function(L,dat,metrika) {
n<-dim(dat)[1]
p<-dim(dat)[2]
x.prum<-1/n*apply(dat,2,sum)
WSS.fuzzy<-numeric(1)
SIL.fuzzy<-numeric(1)
MSS.fuzzy<-numeric(1)

#ROZKLAD#
#(někdy se neprovede rozklad do předem daného počtu shluků L kvůli
#memb.exp, který musíme kvůli tomu snížit)
shl<-fanny(dat,L,diss=FALSE,memb.exp=2,metric=metrika,stand=FALSE)
pokracuj<-TRUE
expon<-2
while (pokracuj) {
        if(expon<=0) {
            print("nelze provest")
            pokracuj<-FALSE
                     }
        if(shl$k.crisp<L) {
            expon<-expon-0.1
            shl<-fanny(dat,L,diss=FALSE,memb.exp=expon,metric=metrika,
                       stand=FALSE)
                          }
        else {pokracuj<-FALSE}
                 } #while (pokracuj)
shl<-fanny(dat,L,diss=FALSE,memb.exp=expon,metric=metrika,stand=FALSE)
print(expon)

#WSS#
prum<-vector(mode="numeric",length=p)
```

```
WK<-vector(mode="numeric",length=L)

for (k in 1:L) {
    mat.D<-dat[shl$clustering==k,]
    nk<-dim(mat.D)[1]
    prum<-1/nk*apply(mat.D,2,sum)
    for (i in 1:nk) {
        WK[k]<-WK[k]+sum((mat.D[i,]-prum)^2)
                    }
                }
WSS.fuzzy<-sum(WK)

#SIL#
SIL.fuzzy<-shl$silinfo$avg.width

#MSS#
med<-vector(mode="numeric",length=L)
WM<-vector(mode="numeric",length=L)

for (k in 1:L) {
    mat.D<-dat[shl$clustering==k,]
    nk<-dim(mat.D)[1]
    vzd<-vector(mode="numeric",length=nk)
    for (i in 1:nk) {
        for (j in 1:nk) {
            vzd[i]<-vzd[i]+sqrt(sum((mat.D[i,]-mat.D[j,])^2))
                        }
        vzd[i]<-vzd[i]/(nk-1)
                    }
    med[k]<-which.min(vzd)
    for (i in 1:nk) {
        WM[k]<-WM[k]+sum((mat.D[i,]-mat.D[med[k],])^2)
                    }
                } #for (k in 1:L)
MSS.fuzzy<-sum(WM)

return(c(WSS.fuzzy,SIL.fuzzy,MSS.fuzzy))
}

###################################
### 4. HIERARCHICAL CLUSTERING ###
###################################
hier.stat<-function(L,dat,metoda,metrika) {
n<-dim(dat)[1]
p<-dim(dat)[2]
x.prum<-1/n*apply(dat,2,sum)
```

```
WSS.hier<-numeric(1)
SIL.hier<-numeric(1)
MSS.hier<-numeric(1)

#ROZKLAD#
shl<-agnes(dat,diss=FALSE,metric=metrika,stand=FALSE,method=metoda)

#následující procedura uřízne dendrogram na požadované hladině shluků
#a vypíše do souboru čísla pozorování pro jednotlivé shluky

i.n<-vector(mode="logical",length=n-1) #indikátor zařazení do shluku
kkk<-TRUE
while (kkk) {
    indikator<-FALSE
    i<-0
    k<-0
while ((i<(n-L+1))&(!indikator)) {
        i<-i+1
        if ((!i.n[i])&((shl$merge[i,1]<0)|(shl$merge[i,2]<0))) {
            indikator<-TRUE
            k<-i
            }}
    if (!indikator) {kkk<-FALSE}
    else {
        i.p<-k #indikátor průchodu
        shluk<-c()
        if (shl$merge[i.p,1]<0){shluk<-c(shluk,-shl$merge[i.p,1])}
        if (shl$merge[i.p,2]<0){shluk<-c(shluk,-shl$merge[i.p,2])}
        i.n[i.p]<-TRUE
        vratit<-c()
        for (i in i.p:(n-L)){
            for (j in 1:2){
                if (shl$merge[i,j]==i.p) {
                    i.p<-i
                    if ((j==1)&(shl$merge[i,2]<0)) {
                        shluk<-c(shluk,-shl$merge[i,2])
                        i.n[i]<-TRUE
                                                }
                    if ((j==2)&(shl$merge[i,1]<0)) {
                        shluk<-c(shluk,-shl$merge[i,1])
                        i.n[i]<-TRUE
                                                }
                    if ((j==1)&(shl$merge[i,2]>0)) {
                        vratit<-c(vratit,shl$merge[i,2])
                        i.n[i]<-TRUE
                                                }
```

```
                    if ((j==2)&(shl$merge[i,1]>0)) {
                        vratit<-c(vratit,shl$merge[i,1])
                        i.n[i]<-TRUE
                                                }
                                            } #if (shl$merge[i,j]==i.p)
                                }} #for
#zpětný průchod
        while (sum(vratit)>0) {
            j<-1
            for (k in 1:length(vratit)) {
                if (vratit[k]>=n-L+1) {vratit[k]<-0}
                                            }
            while (j<=length(vratit)){
                if (vratit[j]==0) {j<-j+1}
                else {
                    i<-vratit[j]
                    j<-length(vratit)+1
                     }
                                        } #while (j<=length(vratit))
            if ((shl$merge[i,1]<0)|(shl$merge[i,1]<n-L+1)) {
                if (shl$merge[i,1]<0) {
                    if (shl$merge[i,2]<0) {i.n[i]<-TRUE}
                    shluk<-c(shluk,-shl$merge[i,1])
                    vratit[which(vratit==i)]<-0
                                            }
                if (shl$merge[i,1]>0) {
                    i.n[i]<-TRUE
                    vratit<-c(vratit,shl$merge[i,1])
                    vratit[which(vratit==i)]<-0
                                            }
} #if ((shl$merge[i,1]<0)|(shl$merge[i,1]<n-L+1))

            if ((shl$merge[i,2]<0)|(shl$merge[i,2]<n-L+1)) {
                if (shl$merge[i,2]<0) {
                    if (shl$merge[i,1]<0) {i.n[i]<-TRUE}
                    shluk<-c(shluk,-shl$merge[i,2])
                    vratit[which(vratit==i)]<-0
                                            }
                if (shl$merge[i,2]>0) {
                    i.n[i]<-TRUE
                    vratit<-c(vratit,shl$merge[i,2])
                    vratit[which(vratit==i)]<-0
                                            }
        } #if ((shl$merge[i,2]<0)|(shl$merge[i,2]<n-L+1))
    } #while (sum(vratit)>0)
    write(shluk,"shluky.dat",ncolumns=n,append=TRUE)
```

```
} #else
        } #while (kkk)

#nyní dojde k uložení pozorování, které na dané hladině L tvoří
#samostatný shluk
for (i in (n-L+1):(n-1)){
for (j in 1:2){
   if (shl$merge[i,j]<0) {
        write(-shl$merge[i,j],"shluky.dat",ncolumns=n,append=TRUE)
                              }
                 }}

#z datového souboru se načtou informace o počtu pozorování v jednotlivých
#shlucích
datice<-read.table("shluky.dat",header=FALSE,fill=TRUE)
size<-vector(mode="numeric",length=L) #počet pozorování v každém shluku
for (k in 1:L) {
     size[k]<-0
     for (j in 1: length(datice[k,])){
          if (!(is.na(datice[k,j]))) {
                size[k]<-size[k]+1
                                    }
                                    }
               }

#WSS#
prum<-vector(mode="numeric",length=p)
WK<-vector(mode="numeric",length=L)

for (k in 1:L) {
     mat.D<-matrix(0,nrow=size[k],ncol=p)
     index<-1
     for (i in datice[k,]) {
          if (!is.na(i)) {
              for (j in 1:p) {
                   mat.D[index,j]<-dat[i,j]
                               }
              index<-index+1
                         }
                           }
    nk<-dim(mat.D)[1]
    prum<-1/nk*apply(mat.D,2,sum)
    for (i in 1:nk) {
         WK[k]<-WK[k]+sum((mat.D[i,]-prum)^2)
                    }
               } #for (k in 1:L)
```

```
WSS.hier<-sum(WK)

#SIL#
cl<-vector(mode="numeric",length=n)
for (i in 1:n) {
    for (k in 1:L) {
        for (j in 1:length(datice[k,])) {
            if ((!is.na(datice[k,j]))&(i==datice[k,j])) {cl[i]<-k}
}}}

a<-vector(mode="numeric",length=n)
b<-vector(mode="numeric",length=n)
s<-vector(mode="numeric",length=n)
d<-matrix(0,nrow=n,ncol=L)

for (i in 1:n) {
    if (size[cl[i]]==1) {s[i]<-0}
    else {
        soucet<-0
        for (j in 1:n) {
            if(cl[j]==cl[i]) {
                soucet<-soucet+sqrt(sum((dat[i,]-dat[j,])^2))
                            }
                        }
        a[i]<-1/(size[cl[i]]-1)*soucet

        for (k in 1:L) {
            soucet<-0
            for (j in 1:n) {
                if ((cl[j]==k)&!(cl[j]==cl[i])) {
                    soucet<-soucet+sqrt(sum((dat[i,]-dat[j,])^2))
                                                }
                            }
            if (cl[i]==k) {d[i,k]<-Inf}
            else {
                d[i,k]<-1/size[k]*soucet
                 }
                        } #for (k in 1:L)
        b[i]<-min(d[i,1:L])

        s[i]<-(b[i]-a[i])/max(a[i],b[i])
  }}
SIL.hier<-sum(s)/n

#MSS#
med<-vector(mode="numeric",length=L)
```

```
MS<-vector(mode="numeric",length=L)
for (k in 1:L) {
    mat.D<-matrix(0,nrow=size[k],ncol=p)
    index<-1
    for (i in datice[k,]) {
        if (!is.na(i)) {
            for (j in 1:p) {
                mat.D[index,j]<-dat[i,j]
   }
            index<-index+1
                       }
                          }
    nk<-dim(mat.D)[1]
    vzd<-vector(mode="numeric",length=nk)
    for (i in 1:nk) {
        for (j in 1:nk) {
            vzd[i]<-vzd[i]+sqrt(sum((mat.D[i,]-mat.D[j,])^2))
                         }
            vzd[i]<-vzd[i]/(nk-1)
                       }
    if (nk>1) {med[k]<-which.min(vzd)}
    if (nk==1) {med[k]<-i}

    for (i in 1:nk) {
        MS[k]<-MS[k]+sum((mat.D[i,]-mat.D[med[k],])^2)
                       }
  } #for (k in 1:L)
MSS.hier<-sum(MS)
return(c(WSS.hier,SIL.hier,MSS.hier))
}
```

Příloha B

Algoritmus k nalezení archetypů

Následující procedura nalezne v dané matici pozorování požadovaný počet archetypů podle algoritmu uvedeného v části 4.2. Na výstupu vrací reziduální součet čtverců, vypočtené archetypy i pomocné matice **A** a **B**.

```
najdi.archetypy <- function(X,arch,tol,maxiter,init) {

 #X ......... datová matice
 #arch ...... počet archetypů
 #tol ....... tolerance na změnu RSS
 #maxiter ... maximalní počet iterací
 #init ...... počet náhodných inicializací

n<-dim(X)[1]
p<-dim(X)[2]

nnsto2<-function(A,b) {
         require(nnls)
         m<-dim(A)[1]
         n<-dim(A)[2]
         x<-matrix(0,nrow=n,ncol=1)
         M<-10^(12)
         AM<-rbind(A,M)
         bM<-rbind(b,M)
         najdi<-nnls(AM,bM)$x
         for (i in 1:n) {x[i]<-najdi[i]}
         return(x)
                     }

### inicializace ###
RSS<-numeric(1)
RSS.opt<-Inf
RSS.optimalni<-Inf
A.opt<-matrix(0,nrow=n,ncol=arch)
```

```
B.opt<-matrix(0,nrow=arch,ncol=n)
Z.opt<-matrix(0,nrow=arch,ncol=p)

for (vv in 1:init) {
ALPHA<-matrix(0,nrow=n,ncol=arch)
BETA<-matrix(0,nrow=arch,ncol=n)
Z<-matrix(0,nrow=arch,ncol=p)

### náhodný výběr pozorování do matice Z ###
init.koef<-TRUE
while (init.koef) {
       init.koef<-FALSE
       mnozina<-rbinom(arch,n,runif(1,0,1))
       for (i in 1:arch) {
            cislo<-mnozina[i]
            if (length(which(mnozina==cislo))>1) {init.koef<-TRUE}
                        }
                  }

Z<-rbind(X[mnozina[1],])
if (arch>1) {
    for (s in mnozina[2:arch]) {Z<-rbind(Z,X[s,])}
            }

pokracuj<-TRUE
iter<-0

while ((pokracuj)&(iter<=maxiter)) {
iter<-iter+1

### výpočet hodnot pro matici ALPHA ###
for (i in 1:n) {
     ALPHA[i,]<-t(nnsto2(t(Z),cbind(X[i,])))
               }

for (l in 1:arch) {
if (round(sum(ALPHA[,l]^2),10)>0) {

v.pruh<-numeric(arch)
V<-matrix(0,nrow=n,ncol=p)
pom<-vector(mode="numeric",p)
pom2<-numeric(1)

for (i in 1:n) {
     if (round(ALPHA[i,l],10)>0) {
         az<-vector(mode="numeric",p)
```

```
         for (j in 1:p) {
              for (k in 1:arch){
                   az[j]<-az[j]+ALPHA[i,k]*Z[k,j]
                              }
                         }
         for (j in 1:p) {
              V[i,j]<-(X[i,j]-az[j]+ALPHA[i,l]*Z[l,j])/ALPHA[i,l]
                         }
                                  } #if(ALPHA[i,l]>0)
     else {
           V[i,]<-rep(0,p)
          }
               }  #for(i in 1:n)

for (j in 1:p) {
     for (ii in 1:n) {
          pom[j] <- pom[j]+ALPHA[ii,l]^2*V[ii,j]
                     }
               }
for (ii in 1:n) {
     pom2 <- pom2+ALPHA[ii,l]^2
                 }
v.pruh<-(pom2)^(-1)*pom

### výpočet eltého řádku matice BETA ###
BETA[l,]<-t(nnsto2(t(X),cbind(v.pruh)))

### výpočet archetypů v případě, že suma alfa il > 0 ###
for (j in 1:p) {
     Z[l,j]<-BETA[l,]%*%X[,j]
                }
} #if sum(ALHPA[,l]^2)>0

### výpočet archetypů v případě, že suma alfa il = 0 ###
else {
      U1<-matrix(0,nrow=n,ncol=p)
      for (ii in 1:n) {
           for (j in 1:p) {
                for (k in 1:arch) {
                     U1[ii,j]<-U1[ii,j]+ALPHA[ii,k]*Z[k,j]
                                 }
                            }
                       }
      RSS.akt<-0
      RSSp.opt<--Inf
      for (i in 1:n) {
```

```
           RSS.akt<-crossprod(X[i,]-U1[i,])
           if (RSS.akt>RSSp.opt) {
                RSSp.opt<-RSS.akt
                Z[l,]<-X[i,]
                                }
                     }

     } #else
               } # for(l in 1:arch)

### výpočet RSS ###
RSS<-0
U<-matrix(0,nrow=n,ncol=p)
for (ii in 1:n) {
     for (j in 1:p) {
          for (k in 1:arch) {
               U[ii,j]<-U[ii,j]+ALPHA[ii,k]*Z[k,j]
                           }
                     }
                }

for (i in 1:n) {
     RSS<-RSS + crossprod(X[i,]-U[i,])
               }
if (RSS.opt-RSS>tol) {
    RSS.opt<-RSS
                     }
else {
      pokracuj<-FALSE
     }
} #while(pokracuj)

### výběr nejlepšího RSS v závislosti na inicializacích ###
if (RSS.opt<RSS.optimalni) {
    RSS.optimalni<-RSS.opt
    Z.opt<-Z
    A.opt<-ALPHA
    B.opt<-BETA
    print(vv)
                          }
}
return(vystup<-list(z=Z.opt,a=A.opt,b=B.opt,res=RSS.optimalni))
}
```

Literatura

J. Anděl. *Základy matematické statistiky*. Matfyzpress, Praha, 2005.

A. Cutler. A Branch and Bound Algorithm for Constrained Least Squares. *Communications in Statistics: Simulation and Computation*, 22(2):305–321, 1993.

A. Cutler a L. Breiman. Archetypal Analysis. *Technometrics*, 36(4):338–347, 1994.

B. Duran a P. Odell. *Cluster Analysis*. Springer-Verlag, New York, 1974.

L. Finesso a P. Spreij. Approximate Nonnegative Matrix Factorization via Alternating Minimization. 2004.
`www.mtns2004.be/database/papersubmission/upload/184.pdf`.

H. P. Friedman a J. Rubin. On Some Invariant Criteria for Grouping Data. *Journal of the American Statistical Association*, 62(320):1159–1178, 1967.

P. Hebák a kol. *Vícerozměrné statistické metody s aplikacemi (3)*. Informatorium, Praha, 2005.

A. K. Jain a R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, New Jersey, 1988.

L. Kaufman a P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, Inc., New York, 1990.

C. Lawson a R. Hanson. *Solving Least Squares Problems*. Prentice-Hall, New Jersey, 1974.

A. McLennan. Lecture notes - Introduction to Mathematical Economics. 1999.
`www.econ.umn.edu/ mclennan/Classes/Ec5113/ec5113-lec14-3.9.99.pdf`.

Q. H. Nguyen a V. J. Rayward-Smith. Internal Quality Measures for Clustering in Metric Spaces. Prezentace na konferenci EURO, 2007.

J. Norstad. Portfolio Optimization - Part 2: Constrained Portfolios. 2005.
`http://homepage.mac.com/j.norstad/finance/portopt2.pdf`.

M. Paldam. Data. 2000.
www.econ.au.dk/vip_htm/MPaldam/papers/corruption&religion/correl-data.pdf.

R. Tibshirani, T. Hastie, a J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

A. Weingessel a E. Dimitriadou. An Examination of Indexes for Determining The Number of Clusters in Binary Data Sets. *Psychometrika*, 67(1):137–160, 2002.