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

DIPLOMOVÁ PRÁCE

Vybrané moderní metody mnohorozměrné statistické analýzy

Vedoucí diplomové práce:
RNDr. Karel Hron, Ph.D.
Rok odevzdání: 2013

Vypracovala:
Bc. Hana Králová
AME, II. ročník

Prohlášení

Prohlašuji, že jsem diplomovou práci vytvořila samostatně za vedení pana RNDr. Karla Hrona, Ph.D., a že jsem v seznamu použité literatury uvedla všechny zdroje, ze kterých jsem při psaní práce čerpala.

V Července dne 26. března 2013

Poděkování

Ráda bych tímto poděkovala svému vedoucímu diplomové práce, panu RNDr. Karlu Hronovi, Ph.D., za odbornou spolupráci, neustálou ochotu a za čas, který mi věnoval. Rovněž bych ráda poděkovala Mgr. Alžbětě Kalivodové, která mi poskytla data ke zpracování. Poděkování si ale také zaslouží má rodina, která mne ve studiu podporovala.

Obsah

Úvod

Téma mé diplomové práce - Vybrané moderní metody mnohorozměrné statistické analýzy - zní zřejmě nejasně a tajemně. Právě taková vlastnost mne zaujala a byla jedním z důvodů, proč jsem toto téma zvolila. K mnohorozměrné statistice jsem měla vždy velmi blízko, proto jsem uchopila příležitost vytvořit si prostřednictvím diplomové práce ucelený přehled metod mnohorozměrné statistiky. A právě díky takto vhodně zvolenému tématu jsem se po konzultaci s vedoucím mé diplomové práce, panem RNDr. Karlem Hronem, Ph.D., rozhodla, kterými vybranými metodami se budu v této práci nadále zabývat.

Cílem mé diplomové práce je vytvořit přehledný teoretický základ vybraných metod mnohorozměrné statistické analýzy a tuto teorii aplikovat na reálná data pomocí statistického softwaru *R*.

První kapitola této práce, která se zabývá obecně mnohorozměrnou statistikou, nás stručně uvede do studované problematiky. Další kapitola se již věnuje shlukové analýze, kde se čtenář seznámí jak s hierarchickým, tak i s nehierarchickým shlukováním. Navazující podkapitola se zabývá dalším možným přístupem shlukování, a to metodou samoorganizujících map. V další kapitole se seznámíme s regresní metodou dílčích nejmenších čtverců a s jejími možnými algoritmy, přičemž úvod této kapitoly nám nastíní i základy pomocných metod, kterými jsou metoda hlavních komponent a metoda mnohonásobné lineární regrese. Tato teoretická část je proložena několika jednoduchými příklady, na kterých budou ukázány vybrané zmíněné algoritmy, avšak závěrečná část práce čtenáři představí aplikaci uvedené teorie na reálných datech a seznámí jej přitom se základními funkcemi potřebnými pro snadnou práci ve statistickém softwaru *R*.

1 Mnohorozměrná statistická analýza

Mnohorozměrnou statistickou analýzu lze považovat za relativně mladou vědu, která v posledních desetiletích prochází významným rozvojem. Jedním z důvodů rychlého vývoje mnohorozměrné statistiky je reakce na rapidní vývoj výpočetní techniky, a to jak na vznik nových výpočetních softwarů, tak i na možnost ukládání masivního množství dat. Dalším důvodem rostoucí popularity mnohorozměrné statistiky je velký zájem o statistickou analýzu ve všech oborech, jako je například lékařství, psychologie, ekonomika, státní správa, astronomie apod. Mezi nejznámější metody mnohorozměrné statistické analýzy patří metoda hlavních komponent, faktorová analýza, diskriminační analýza, shluková analýza, korelační analýza a regresní analýza, přičemž má práce je zaměřena na shlukovou analýzu, regresní metodu PLS a okrajově i na metodu hlavních komponent a metodu mnohonásobné lineární regrese.

Mnohorozměrná statistická analýza se zabývá zpracováváním mnohorozměrných dat, což jsou opakovaná měření vybraných proměnných (statistických znaků). Tato získaná data ukládáme do databází, které si můžeme obecně představit jako $(n \times p)$-rozměrnou matici, kde n je počet měření a p je počet měřených proměnných.

$$\mathbf{X}_{(n\times p)} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

Tyto databáze jsou řízeny tzv. databázovým systémem, což je speciální software, který uživatelům umožňuje do databází vstupovat, data doplňovat či promazávat a také data statisticky zpracovávat. Tato problematika však není předmětem této práce, proto více informací o databázových systémech lze najít například v [5].

Jak již bylo zmíněno, databáze slouží k ukládání a zpracovávání masivního množství dat, proto je třeba počítat s tím, že často dochází k chybám. Jednou

z nich mohou být zaměněná data, chybějící data nebo odlehlá pozorování. Protože je však tato problematika velmi rozsáhlá, řešení některých zmíněných problémů popíšu jen stručně.

V případě chybějících dat můžeme postupovat různým způsobem. Nejjednodušším, avšak ne nejlepším řešením, je pracovat pouze s těmi pozorováními, u nichž známe hodnotu dle všech proměnných. Vynechání zbylých pozorování však může způsobit zkreslení celé původní množiny dat. Proto se často přistupuje k jinému řešení, a to k imputaci chybějících hodnot. Chybějící hodnoty můžeme nahradit například průměrem známých hodnot dané proměnné (v případě více chybějících hodnot ovšem tento přístup vede k destrukci datové struktury) anebo využít regresi, kdy chybějící pozorování nahradíme jeho predikovanou hodnotou.

Dalším problémem statistické analýzy jsou odlehlá pozorování, která lze vizuálně odhalit pomocí vhodných grafických nástrojů (histogramů, boxplotů apod.). Zda se však jedná o skutečná odlehlá pozorování, můžeme usoudit na základě speciálních testů.

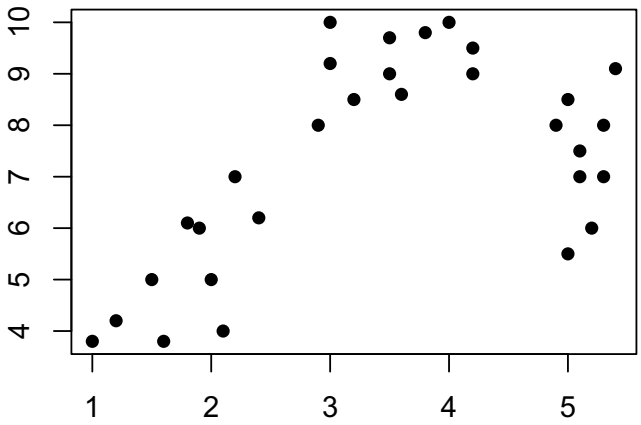
V následujících kapitolách se budu blíže věnovat vybraným moderním metodám mnohorozměrné statistické analýzy, přičemž se budu snažit některé postupy ukázat na praktických příkladech.

Tato kapitola byla sepsána pomocí zdrojů [5] a [6].

2 Shluková analýza

Shluková analýza je v současnosti díky své názornosti velice populární metodou mnohorozměrné statistiky. Metoda shlukové analýzy obecně spočívá v seskupování pozorování z vícerozměrného datového souboru do homogenních skupin neboli shluků. Pro pojem shluk přitom neexistuje žádná obecná definice, proto výsledkem této metody může být vždy různá struktura shluků. Shlukem však můžeme rozumět skupinu pozorování, která jsou v určitém smyslu blízká nějakému reprezentantovi dané skupiny.

Grafické znázornění shluků lze ukázat na následujícím obrázku 1. Příslušný statistický soubor je tvořen 28 dvourozměrnými pozorováními. Na základě tohoto vizuálního nástroje lze usoudit, že data jsou rozdělena do tří shluků.



Obrázek 1: grafické zobrazení shluků

Na shlukovou analýzu se lze dívat také jako na pomocný krok při statistické analýze velkých datových souborů, kdy chceme analyzovat pouze reprezentanty nalezených shluků.

Své využití metoda nachází v různých odvětvích. Například v bankovnictví je metoda využívána k segmentaci klientů na základě různých charakteristik, jako je výše příjmu, průměrné měsíční výdaje, věk apod.

Poznámka 2.1. Vedle třídění objektů do několika stejnorodých skupin lze seskupovat rovněž proměnné, což vede ke snížení rozměru dané úlohy. Speciálním

případem je dvourozměrná shluková analýza, pomocí které lze třídit do skupin jak data, tak i proměnné.

Při zpracování kapitoly o shlukové analýze, hierarchickém i nehierarchickém shlukování byly využity především zdroje [3] a [5].

2.1 Míry vzdálenosti

Často používaným nástrojem metod shlukové analýzy je měření vzdálenosti či podobnosti jednotlivých pozorování. Uvažujeme tedy pozorování $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$ a $\mathbf{x}_j = (x_{j1}, \ldots, x_{jp})^T$, $i, j = 1, \ldots, n$ z množiny $\mathbb{R}^p$. Pro výpočet vzdáleností mezi jednotlivými pozorováními využíváme v praxi nejčastěji následující metriky.
Euklidovská vzdálenost

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2}.$$

Manhattanská vzdálenost

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{p} |x_{ik} - x_{jk}|.$$

Čebyševova vzdálenost

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

Lancey-Williamsova vzdálenost

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{p} \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|}.$$

Minkowského vzdálenost

$$d_m(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{p} |x_{ik} - x_{jk}|^m \right)^{\frac{1}{m}},$$

jejímž speciálním případem je euklidovská, resp. manhattanská vzdálenost pro $m = 2$, resp. pro $m = 1$.

Vzdálenosti mezi těmito pozorováními přitom splňují následující vlastnosti metriky

$$d(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \quad d(\mathbf{x}_i, \mathbf{x}_i) = 0,$$

$$d(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_j, \mathbf{x}_i), \quad d(\mathbf{x}_i, \mathbf{x}_j) \leq d(\mathbf{x}_i, \mathbf{x}_k) + d(\mathbf{x}_k, \mathbf{x}_j),$$

kde $i, j, k = 1, \ldots, n$.

V případě shlukování proměnných jejich podobnost (nejedná se o vzdálenost ve výše uvedeném smyslu) zjišťujeme pomocí výběrového korelačního koeficientu, tedy pro sloupce $\mathbf{x}_{\cdot i}$, $\mathbf{x}_{\cdot j}$, kde $i, j = 1, \ldots, p$, datové matice $\mathbf{X}$ uvažujeme jako míru podobnosti

$$d(\mathbf{x}_{\cdot i}, \mathbf{x}_{\cdot j}) = 1 - \rho_{ij} = 1 - \frac{s_{ij}}{s_i s_j} = 1 - \frac{\sum_{k=1}^{n}(x_{ki} - \overline{x}_i)(x_{kj} - \overline{x}_j)}{\sqrt{\sum_{k=1}^{n}(x_{ki} - \overline{x}_i)^2}\sqrt{\sum_{k=1}^{n}(x_{kj} - \overline{x}_j)^2}},$$

kde $\rho_{ij} \in \langle -1, 1 \rangle$ je korelační koeficient mezi i-tou a j-tou proměnnou a $\overline{x}_i$ a $\overline{x}_j$ jsou výběrové průměry příslušných proměnných. Pro všechny dvojice proměnných bychom dostali výběrovou korelační matici rozměru $(p \times p)$ s jedničkami na diagonále.

Obecně shlukovou analýzu rozlišujeme na hierarchickou a nehierarchickou, přičemž oba typy budou podrobně popsány níže.

2.2 Hierarchická shluková analýza

Hierarchická shluková analýza patří k nejčastěji užívaným postupům shlukové analýzy, přičemž využívá dvou možných algoritmů pro určení shluků, a to aglomerativní shlukování a divizivní shlukování. Hlavní myšlenkou aglomerativního shlukování je předpoklad, že na počátku má každé pozorování vlastní shluk a v průběhu algoritmu se shluky spojují, dokud nevznikne jediný shluk pro všechna pozorování. Naopak u divizivního shlukování předpokládáme, že

v prvním kroku algoritmu jsou všechna pozorování v jednom shluku a postupným rozdělováním získáme tolik shluků, kolik je pozorování. Nyní se zaměřím na jednotlivé algoritmy podrobněji.

2.2.1 Aglomerativní hierarchické shlukování

Jak již bylo v předchozím odstavci zmíněno, hlavní myšlenkou metody aglomerativního hierarchického shlukování je předpoklad, že každé pozorování má na počátku vlastní shluk. Uvažujeme tedy obecně n pozorování $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, kde n je zároveň počátečním počtem shluků. Nyní si algoritmus aglomerativního hierarchického shlukování představíme v jednotlivých krocích.

- Prvním krokem algoritmu aglomerativního hierarchického shlukování bude výpočet párových vzdáleností mezi všemi pozorováními, které následně uspořádáme do symetrické matice $\mathbf{D} = (d_{ij})$, kde $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$.

- Poté najdeme nejmenší vzdálenost $d_{I,J}$ v matici $\mathbf{D}$ (vyjma diagonály) a spojíme shluky I a J, a tak vytvoříme nový shluk IJ.

- Dále vypočítáme vzdálenosti $d_{IJ,K}$ mezi nově vytvořeným shlukem IJ a ostatními shluky $K \neq IJ$ pomocí jednoho z následujících možných vztahů

$$\text{single linkage:} \qquad d_{IJ,K} = min\{d_{I,K}, d_{J,K}\},$$

$$\text{complete linkage:} \qquad d_{IJ,K} = max\{d_{I,K}, d_{J,K}\},$$

$$\text{average linkage:} \qquad d_{IJ,K} = \frac{\sum_{i \in IJ} \sum_{k \in K} d_{ik}}{n_{IJ} n_K},$$

  kde n_{IJ} a n_K jsou počty pozorování ve shlucích IJ a K.

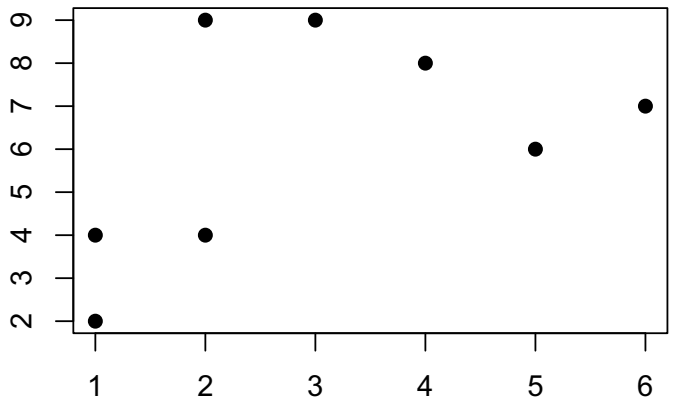
  Následně sestrojíme novou $((n-1) \times (n-1))$-rozměrnou matici $\mathbf{D}_2$, která vznikne z matice $\mathbf{D}$ vynecháním řádků a sloupců I a J a přidáním nového řádku a sloupce IJ se vzdálenostmi určenými pomocí výše uvedených vztahů.

- Předchozí dva kroky opakujeme $(n-1)$-krát, dokud nejsou všechna pozorování seskupena do jednoho shluku.

Takto popsaný algoritmus nyní aplikujeme na následující příklad pro jeho názorné pochopení.

Příklad 2.1. Předpokládejme osm pozorování o dvou proměnných, tedy $\mathbf{x}_1 = (1,2)^T$, $\mathbf{x}_2 = (2,4)^T$, $\mathbf{x}_3 = (1,4)^T$, $\mathbf{x}_4 = (5,6)^T$, $\mathbf{x}_5 = (6,7)^T$, $\mathbf{x}_6 = (4,8)^T$, $\mathbf{x}_7 = (2,9)^T$, $\mathbf{x}_8 = (3,9)^T$. Najděme ve skupině těchto pozorování shluky za využití algoritmu hierarchického aglomerativního shlukování.

Pro ilustraci uvádíme následující obrázek 2, který zobrazuje daná data v rovině.



Obrázek 2: grafické zobrazení dat

Řešení: V prvním kroku algoritmu, za použití vzorce single linkage, sestavíme matici euklidovských vzdáleností mezi jednotlivými pozorováními $\mathbf{D}_1$. Omezíme se přitom na trojúhelníkovou matici, neboť matice vzdáleností je symetrická.

	1	2	3	4	5	6	7	8
1	0	2.236	2.000	5.657	7.071	6.708	7.071	7.280
2		0	**1.000**	3.606	5.000	4.472	5.000	5.099
3			0	4.472	5.831	5.000	5.099	5.385
4				0	1.414	2.236	4.243	3.606
5					0	2.236	4.472	3.606
6						0	2.236	1.414
7							0	**1.000**
8								0

Nejmenší vzdálenost v matici $\mathbf{D}_1$ je $d_{2,3} = d_{7,8} = 1$, přičemž zvolíme jen jednu z nich. Spojíme tedy například shluky 2 a 3, čímž vytvoříme nový shluk 23. Nyní je třeba přepočítat matici vzdáleností. Matice $\mathbf{D}_2$ bude změněna o prvky $d_{23,K} = min\{d_{2,K}, d_{3,K}\}$ pro všechna $K = 1, 4, 5, 6, 7, 8$. Příkladně $d_{23,4} = min\{d_{2,4}, d_{3,4}\} = min\{3.606, 4.472\} = 3.606$ atp.

	1	23	4	5	6	7	8
1	0	2.000	5.657	7.071	6.708	7.071	7.280
23		0	3.606	5.000	4.472	5.000	5.099
4			0	1.414	2.236	4.243	3.606
5				0	2.236	4.472	3.606
6					0	2.236	1.414
7						0	**1.000**
8							0

Nyní opět zvolíme nejmenší prvek matice $\mathbf{D}_2$, kterým je prvek $d_{7,8} = 1$, spojíme tedy shluky 7 a 8 a vytvoříme nový shluk 78. Přepočítáním jednotlivých vzdáleností opět sestavíme novou matici vzdáleností $\mathbf{D}_3$.

	1	23	4	5	6	78
1	0	2.000	5.657	7.071	6.708	7.071
23		0	3.606	5.000	4.472	5.000
4			0	**1.414**	2.236	3.606
5				0	2.236	3.606
6					0	**1.414**
78						0

Analogicky najdeme nejmenší vzdálenost, zde je to $d_{4,5} = d_{6,78} = 1.414$. Spojíme tedy například shluky 6 a 78 a přepočítáme novou matici $\mathbf{D}_4$.

	1	23	4	5	678
1	0	2.000	5.657	7.071	6.708
23		0	3.606	5.000	4.472
4			0	**1.414**	2.236
5				0	2.236
678					0

Minimální vzdálenost matice $\mathbf{D}_4$ je $d_{4,5} = 1.414$ a obdobně vytvoříme nový shluk 45 a sestrojíme matici vzdáleností $\mathbf{D}_5$.

	1	23	45	678
1	0	**2.000**	5.657	6.708
23		0	3.606	4.472
45			0	2.236
678				0

Nejmenší prvek matice $\mathbf{D}_5$ je vzdálenost $d_{1,23} = 2$, proto shluky 1 a 23 spojíme a přepočítáme nové vzdálenosti matice $\mathbf{D}_6$.

	123	45	678
123	0	3.606	4.472
45		0	**2.236**
678			0

Naposledy najdeme minimální vzdálenost matice $\mathbf{D}_6$, což je $d_{45,678} = 2.236$, čímž získáme shluk 45678. Závěrečná matice vzdáleností tohoto algoritmu je následující matice $\mathbf{D}_7$.

	123	45678
123	0	3.606
45678		0

Z této konečné matice vzdáleností vidíme, že jsme získali dva shluky pro daná pozorování. Shluk 1 je tvořen pozorováními $\mathbf{x}_1$, $\mathbf{x}_2$ a $\mathbf{x}_3$ a shluk 2 je tvořen pozorováními $\mathbf{x}_4$, $\mathbf{x}_5$, $\mathbf{x}_6$, $\mathbf{x}_7$ a $\mathbf{x}_8$.

V případě použití vzorce complete linkage bychom postupovali velice podobně, jen bychom nové vzdálenosti mezi shluky přepočítávali jako maximum z původních vzdáleností.

V případě vzorce average linkage si postup výpočtu ukážeme názorně na stejném zadání. I v tomto případě vycházíme z původní matice euklidovských vzdáleností mezi jednotlivými pozorováními $\mathbf{D}_1$.

	1	2	3	4	5	6	7	8
1	0	2.236	2.000	5.657	7.071	6.708	7.071	7.280
2		0	**1.000**	3.606	5.000	4.472	5.000	5.099
3			0	4.472	5.831	5.000	5.099	5.385
4				0	1.414	2.236	4.243	3.606
5					0	2.236	4.472	3.606
6						0	2.236	1.414
7							0	**1.000**
8								0

Jako v předchozích případech nejmenší vzdálenost matice $\mathbf{D}_1$ je $d_{2,3} = d_{7,8} = 1$, tedy spojíme například shluky 2 a 3, čímž vytvoříme nový shluk 23. Dále přepočítáme matici vzdáleností. Matice $\mathbf{D}_2$ bude změněna o prvky $d_{23,K} = \frac{d_{2,K}+d_{3,K}}{2}$ pro všechna $K = 1, 4, 5, 6, 7, 8$.

	1	23	4	5	6	7	8
1	0	2.118	5.657	7.071	6.708	7.071	7.280
23		0	4.039	5.416	4.736	5.050	5.242
4			0	1.414	2.236	4.243	3.606
5				0	2.236	4.472	3.606
6					0	2.236	1.414
7						0	**1.000**
8							0

V matici $\mathbf{D}_2$ opět zvolíme nejmenší prvek matice, kterým je prvek $d_{7,8} = 1$ a vytvoříme nový shluk 78. Přepočítáním příslušných vzdáleností pomocí vztahu $d_{78,K} = \frac{d_{7,K}+d_{8,K}}{2}$ pro všechna $K = 1, 4, 5, 6$ a $d_{23,78} = \frac{d_{2,7}+d_{3,7}+d_{2,8}+d_{3,8}}{4}$ sestavíme novou matici vzdáleností $\mathbf{D}_3$.

	1	23	4	5	6	78
1	0	2.118	5.657	7.071	6.708	7.176
23		0	4.039	5.416	4.736	5.146
4			0	**1.414**	2.236	3.925
5				0	2.236	4.039
6					0	1.825
78						0

Tentokrát je minimální vzdálenost $d_{4,5} = 1.414$. Spojíme proto shluky 4 a 5 a přepočítáme vzdálenosti nové matice $\mathbf{D}_4$ pomocí vztahů $d_{45,1} = \frac{d_{4,1}+d_{5,1}}{2}$, $d_{45,23} =$

$\frac{d_{4,2}+d_{4,3}+d_{5,2}+d_{5,3}}{4}$ atp.

	1	23	45	6	78
1	0	2.118	6.364	6.708	7.176
23		0	4.727	4.736	5.146
45			0	2.236	3.982
6				0	**1.825**
78					0

Minimální vzdálenost matice $\mathbf{D}_4$ je $d_{6,78} = 1.825$, čímž vzniká nový shluk 678. Matici vzdáleností $\mathbf{D}_5$ sestrojíme pomocí vztahů $d_{678,1} = \frac{d_{6,1}+d_{7,1}+d_{8,1}}{2}$, $d_{678,23} = \frac{d_{6,2}+d_{6,3}+d_{7,2}+d_{7,3}+d_{8,2}+d_{8,3}}{6}$ atp.

	1	23	45	678
1	0	**2.118**	6.364	7.020
23		0	4.727	5.009
45			0	3.400
678				0

Nejmenší prvek matice $\mathbf{D}_5$ je vzdálenost $d_{1,23} = 2.118$, proto přepočítáme nové vzdálenosti matice $\mathbf{D}_6$ mezi nově vytvořeným shlukem 123 a ostatními příslušnými shluky.

	123	45	678
123	0	5.273	5.679
45		0	**3.400**
678			0

V posledním kroku algoritmu najdeme minimální vzdálenost matice $\mathbf{D}_6$, což je $d_{45,678} = 3.400$, a tak vytvoříme shluk 45678. Konečná matice vzdáleností je následující matice $\mathbf{D}_7$.

	123	45678
123	0	5.517
45678		0

Závěrem můžeme opět konstatovat, že jsme získali dva shluky pro daná pozorování. Shluk 1 tvoří pozorování $\mathbf{x}_1$, $\mathbf{x}_2$ a $\mathbf{x}_3$ a shluk 2 tvoří pozorování $\mathbf{x}_4$, $\mathbf{x}_5$, $\mathbf{x}_6$, $\mathbf{x}_7$ a $\mathbf{x}_8$.

2.2.2 Divizivní hierarchické shlukování

Jak již bylo zmíněno, hlavní myšlenkou divizivního shlukování je předpoklad, že všechna pozorování jsou na počátku v jednom shluku. Nyní si divizivní hierarchické shlukování popíšeme postupně.

- V prvním kroku algoritmu metody divizivního shlukování rozdělíme počáteční množinu dat na shluk X a shluk Y, kde shluk X bude tvořen těmi pozorováními, která mají největší průměrnou vzdálenost od ostatních pozorování. Zbylá pozorování budou tvořit shluk Y.

- Následně pro každé pozorování ve shluku Y vypočítáme dvě charakteristiky, a to

  ($\star$) průměrnou vzdálenost mezi daným pozorováním a všemi ostatními pozorováními ve shluku Y a

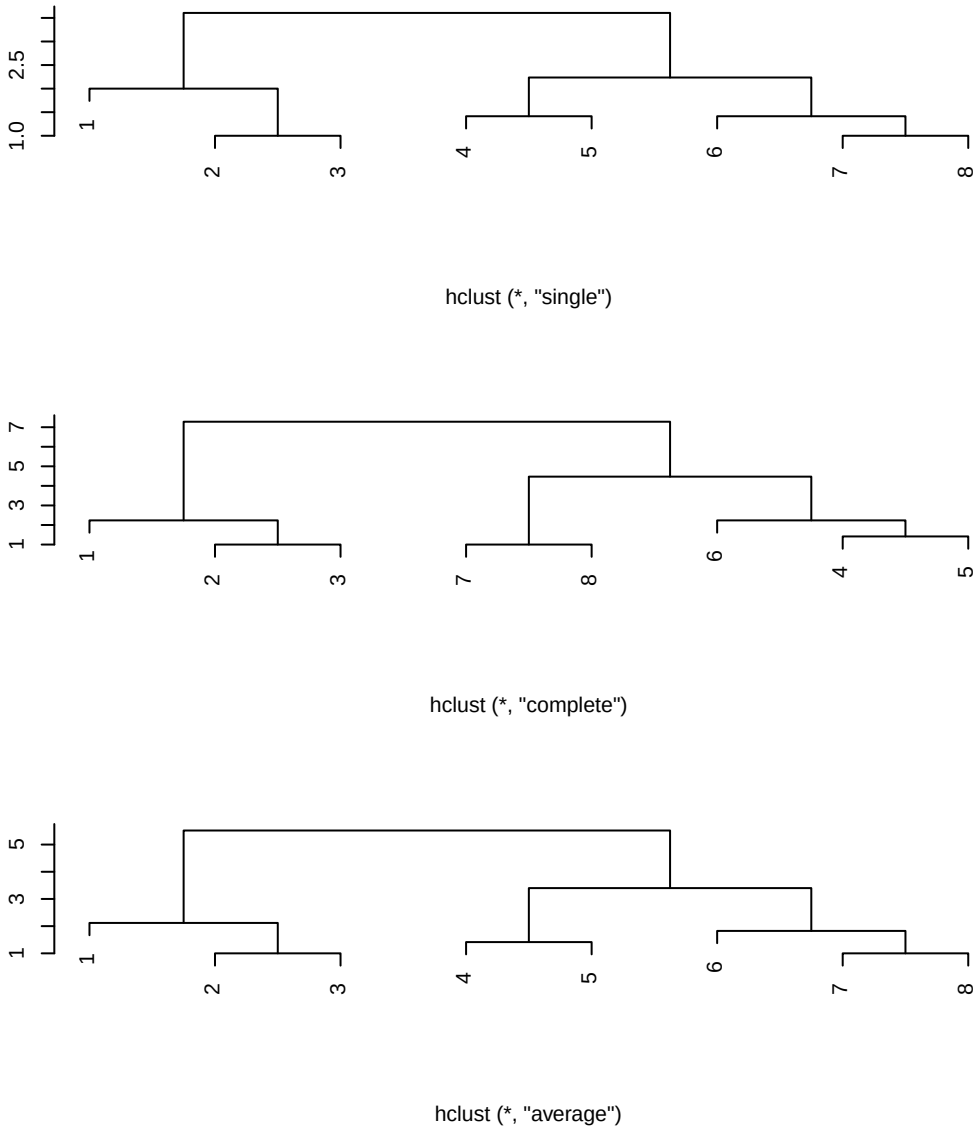
  ($\star\star$) průměrnou vzdálenost mezi daným pozorováním a všemi ostatními pozorováními ve shluku X.

- Poté vypočítáme rozdíl $(\star) - (\star\star)$ pro každé pozorování ve shluku Y. Pokud tento rozdíl nabývá pro všechna pozorování záporné hodnoty, algoritmus ukončíme. V opačném případě pozorování s největší kladnou hodnotou rozdílu přemístíme ze shluku Y do shluku X a postup opakujeme.

- V dalším kroku aplikujeme výše popsaný algoritmus na každý ze shluků X a Y zvlášť, až nakonec každé pozorování tvoří vlastní shluk.

Grafickým výstupem všech algoritmů hierarchického shlukování je dendrogram. Tento vertikální či horizontální stromový graf nás vizuálně informuje o vzniklých shlucích. Dendrogram nám na základě výšky větví, kterou odečteme na y-ové ose, říká, jak jsou si daná pozorování blízká.

Jednotlivé kroky předchozího příkladu lze ilustrovat na obrázku 3.



Obrázek 3: dendrogramy hierarchického shlukování

2.3 Nehierarchická shluková analýza

Základní odlišností nehierarchických metod shlukování od hierarchických je předem daný počet shluků a absence vztahu mezi jednotlivými shluky. Není zde tedy žádný hierarchický vztah mezi k-tým a $(k+1)$-ním shlukem. Mezi nejznámější metody nehierarchické shlukové analýzy patří metoda k průměrů, fuzzy shlukování nebo siluetový graf. Nyní se na jednotlivé metody zaměřím podrobněji.

2.3.1 Metoda k průměrů

Předpokladem algoritmu metody k průměrů jsou daná pozorování $\mathbf{x}_1, \ldots, \mathbf{x}_n$ z množiny $\mathbb{R}^p$ a předem daný počet shluků K.

- Prvotním krokem algoritmu této metody je náhodné přiřazení jednotlivých pozorování do jednoho ze shluků. Následně spočítáme průměry jednotlivých shluků $\overline{\mathbf{x}}_k$ pro všechna $k = 1, \ldots, K$.

- Dále vypočítáme součet čtverců[1] jednotlivých pozorování od příslušných průměrů ESS, tedy

$$ESS = \sum_{k=1}^{K} \sum_{c(i)=k} (\mathbf{x}_i - \overline{\mathbf{x}}_k)^T (\mathbf{x}_i - \overline{\mathbf{x}}_k),$$

  kde $c(i)$ je shluk obsahující pozorování $\mathbf{x}_i$.

- Cílem metody k průměrů je minimalizace výrazu ESS, proto pozorování ve shlucích přeuspořádáme a algoritmus opakujeme tak dlouho, dokud nenajdeme minimum ESS.

Poznámka 2.2. Řešení této metody není jediné. Proto se doporučuje vyjít z různých počátečních rozdělení pozorování do shluků. Výsledkem pak bude uspořádání do shluků s nejmenší hodnotou nalezených minim ESS.

[1] Zkratka součtu čtverců ESS vychází z anglického výrazu Euclidean Sum of Squares.

Příklad 2.2. Uvažujme stejné zadání jako v předchozím příkladu. Předpokládejme osm pozorování o dvou proměnných $\mathbf{x}_1 = (1,2)^T$, $\mathbf{x}_2 = (2,4)^T$, $\mathbf{x}_3 = (1,4)^T$, $\mathbf{x}_4 = (5,6)^T$, $\mathbf{x}_5 = (6,7)^T$, $\mathbf{x}_6 = (4,8)^T$, $\mathbf{x}_7 = (2,9)^T$, $\mathbf{x}_8 = (3,9)^T$ a nechť $K = 2$. Přiřaďme jednotlivá pozorování do jednoho ze dvou shluků za využití metody k průměrů.

Řešení: V prvním kroku jednotlivá pozorování náhodně přiřadíme do jednoho ze shluků. Shluk 1 bude tvořen pozorováními $\mathbf{x}_1$, $\mathbf{x}_3$, $\mathbf{x}_4$ a $\mathbf{x}_8$ a shluk 2 bude tvořen pozorováními $\mathbf{x}_2$, $\mathbf{x}_5$, $\mathbf{x}_6$ a $\mathbf{x}_7$.

Následně spočítáme průměry těchto shluků, tedy $\overline{\mathbf{x}}_1 = (2.5, 5.25)^T$ a $\overline{\mathbf{x}}_2 = (3.5, 7)^T$. Nyní vypočítáme čtvercové vzdálenosti jednotlivých pozorování od příslušných průměrů,

$d^2(\mathbf{x}_1, 1) = (1 - 2.5)^2 + (2 - 5.25)^2 = 12.8125,$

$d^2(\mathbf{x}_1, 2) = (1 - 3.5)^2 + (2 - 7)^2 = 31.25.$

Protože vzdálenost prvního pozorování od průměru prvního shluku je menší než vzdálenost od druhého shluku, pozorování $\mathbf{x}_1$ ponecháme ve shluku 1. Další obdobné výpočty jsou pro přehled uvedeny v následující tabulce.

$d^2(\mathbf{x}_2, 1) = \mathbf{1.8125}$ $d^2(\mathbf{x}_2, 2) = 11.25$	x_2 přemístíme do shluku 1
$d^2(\mathbf{x}_3, 1) = \mathbf{3.8125}$ $d^2(\mathbf{x}_3, 2) = 15.25$	x_3 zůstává ve shluku 1
$d^2(\mathbf{x}_4, 1) = 6.8125$ $d^2(\mathbf{x}_4, 2) = \mathbf{3.25}$	x_4 přemístíme do shluku 2
$d^2(\mathbf{x}_5, 1) = 15.3125$ $d^2(\mathbf{x}_5, 2) = \mathbf{6.25}$	x_5 zůstává ve shluku 2
$d^2(\mathbf{x}_6, 1) = 9.8125$ $d^2(\mathbf{x}_6, 2) = \mathbf{1.25}$	x_6 zůstává ve shluku 2
$d^2(\mathbf{x}_7, 1) = 14.3125$ $d^2(\mathbf{x}_7, 2) = \mathbf{6.25}$	x_7 zůstává ve shluku 2
$d^2(\mathbf{x}_8, 1) = 14.3125$ $d^2(\mathbf{x}_8, 2) = \mathbf{4.25}$	x_8 přemístíme do shluku 2

Pak součet čtverců $ESS = 39.6875$.

Na základě vypočítaných vzdáleností jsme tak přemístili pozorování $\mathbf{x}_2$, $\mathbf{x}_4$ a $\mathbf{x}_8$ do opačných shluků. Proto je třeba opět přepočítat průměry, tedy $\overline{\mathbf{x}}_1 = (1.33, 3.33)^T$ a $\overline{\mathbf{x}}_2 = (4, 9.75)^T$. Obdobným způsobem nyní vypočítáme nové čtvercové vzdálenosti jednotlivých pozorování od jednotlivých průměrů shluků.

$d^2(\mathbf{x}_1, 1) = \mathbf{1.8778}$ $d^2(\mathbf{x}_1, 2) = 69.0625$	x_2 zůstává ve shluku 1
$d^2(\mathbf{x}_2, 1) = \mathbf{0.8978}$ $d^2(\mathbf{x}_2, 2) = 37.0625$	x_2 zůstává ve shluku 1
$d^2(\mathbf{x}_3, 1) = \mathbf{0.5578}$ $d^2(\mathbf{x}_3, 2) = 42$	x_3 zůstává ve shluku 1
$d^2(\mathbf{x}_4, 1) = 20.5978$ $d^2(\mathbf{x}_4, 2) = \mathbf{15.0625}$	x_4 zůstává ve shluku 2
$d^2(\mathbf{x}_5, 1) = 35.2778$ $d^2(\mathbf{x}_5, 2) = \mathbf{11.5626}$	x_5 zůstává ve shluku 2
$d^2(\mathbf{x}_6, 1) = 28.9378$ $d^2(\mathbf{x}_6, 2) = \mathbf{3.0625}$	x_6 zůstává ve shluku 2
$d^2(\mathbf{x}_7, 1) = 35.5978$ $d^2(\mathbf{x}_7, 2) = \mathbf{4.5625}$	x_7 zůstává ve shluku 2
$d^2(\mathbf{x}_8, 1) = 34.9378$ $d^2(\mathbf{x}_8, 2) = \mathbf{1.5625}$	x_8 zůstává ve shluku 2

Tentokrát je součet čtverců $ESS = 39.146$, který je v tomto případě menší než v předchozím uspořádání pozorování.

Na základě vypočítaných vzdáleností není již třeba žádné pozorování přemisťovat, proto lze shrnout, že shluk 1 je tvořen pozorováními $\mathbf{x}_1$, $\mathbf{x}_2$ a $\mathbf{x}_3$ a shluk 2 je tvořen pozorováními $\mathbf{x}_4$, $\mathbf{x}_5$, $\mathbf{x}_6$, $\mathbf{x}_7$ a $\mathbf{x}_8$.

2.3.2 Fuzzy shlukování

Základní myšlenkou metody fuzzy shlukování je předpoklad, že jednotlivým pozorováním přiřazujeme stupně příšlušnosti k jednotlivým shlukům. Stupeň příslušnosti i-tého pozorování ke k-tému shluku značíme u_{ik}, a protože se jedná o normované váhy, budou splňovat vlastnosti $u_{ik} \geq 0$ a $\sum_{k=1}^{K} u_{ik} = 1$ pro každé pozorování $i = 1, \ldots, n$. Tímto se metoda výrazně liší od předchozí metody k průměrů, kde je každé pozorování jednoznačně přiřazeno do jednoho ze shluků.

V algoritmu fuzzy shlukování předpokládáme, že známe matici vzdáleností jednotlivých pozorování $\mathbf{D} = (d_{ij})$ a počet shluků K.

- Neznámé stupně příslušnosti jednotlivých pozorování k jednotlivým shlukům $\{u_{ik}\}$ najdeme minimalizací vztahu

$$\sum_{k=1}^{K} \frac{\sum_i \sum_j u_{ik}^r u_{jk}^r d_{ij}}{2 \sum_l u_{lk}^r},$$

  kde $r > 1$, $i, j, l = 1, \ldots, n$ a $k = 1, \ldots, K$. V praxi se tato minimalizace řeší numericky.

Poznámka 2.3. Většina dostupných literatur uvádí výše zmíněný vzorec pro $r = 2$, avšak v praxi se za r dosazují hodnoty menší než 2, které vedou k větší "fuzzyfikaci" výsledného přiřazení do shluků.

Příklad 2.3. Vycházejme opět ze stejného zadání jako v předchozích příkladech, tedy předpokládejme osm pozorování o dvou proměnných $\mathbf{x}_1 = (1, 2)^T$, $\mathbf{x}_2 = (2, 4)^T$, $\mathbf{x}_3 = (1, 4)^T$, $\mathbf{x}_4 = (5, 6)^T$, $\mathbf{x}_5 = (6, 7)^T$, $\mathbf{x}_6 = (4, 8)^T$, $\mathbf{x}_7 = (2, 9)^T$, $\mathbf{x}_8 = (3, 9)^T$ a uvažujme dva shluky, tj. $K = 2$. Přiřaďme jednotlivá pozorování do jednoho ze dvou shluků za využití metody fuzzy shlukování.

Řešení: Za využití statistického softwaru *R* jsme v knihovně *cluster* za pomocí příkazu *fanny(d,2)*, kde d je matice vzdáleností mezi jednotlivými pozorováními a 2 značí požadovaný počet shluků, získali hledané stupně příslušností jednotlivých pozorování k oběma shlukům $\{u_{ik}\}$, kde $i = 1, \ldots, 8$ a $k = 1, 2$. Výsledku jsme dosáhli po 17 iteracích algoritmu. Výsledné stupně příslušnosti jsou znázorněny v následující tabulce.

	u_{i1}	u_{i2}
$\mathbf{x}_1$	**0.82**	0.18
$\mathbf{x}_2$	**0.89**	0.11
$\mathbf{x}_3$	**0.92**	0.80
$\mathbf{x}_4$	0.27	**0.73**
$\mathbf{x}_5$	0.22	**0.78**
$\mathbf{x}_6$	0.12	**0.88**
$\mathbf{x}_7$	0.18	**0.82**
$\mathbf{x}_8$	0.13	**0.87**

Slovně bychom tyto výsledky mohli interpretovat například tak, že pozorování $\mathbf{x}_1$ patří do shluku 1 ve stupni příslušnosti 0.82 a do shluku 2 patří ve stupni 0.18 atp. Z výsledných stupňů příslušností jednotlivých pozorování k oběma shlukům lze tedy usoudit, že shluk 1 bude tvořen pozorováními $\mathbf{x}_1$, $\mathbf{x}_2$ a $\mathbf{x}_3$ a shluk 2 bude tvořen pozorováními $\mathbf{x}_4$, $\mathbf{x}_5$, $\mathbf{x}_6$, $\mathbf{x}_7$ a $\mathbf{x}_8$. Docílili jsme tedy opět stejného výsledku jako v předchozích metodách.

2.3.3 Siluetový graf

Užitečným výstupem výše zmíněných nehierarchických metod je siluetový graf. Často se rovněž setkáváme s pojmem graf obrysů shluků či s anglickým pojmem silhouette plot. Tento graf nás vizuálně informuje o získaném rozdělení pozorování do jednotlivých shluků.

Předpokládejme, že daná pozorování jsou rozdělena do K shluků a označme $c(i)$ shluk, který obsahuje i-té pozorování, kde $i = 1, \ldots, n$. Nechť dále $a(i)$ je průměrná vzdálenost i-tého pozorování k ostatním pozorováním ve shluku $c(i)$. Dále předpokládejme, že c je jiný shluk a nechť $d(i, c)$ je průměrná vzdálenost i-tého pozorování k pozorováním ve shluku c.

- Prvním krokem postupu vykreslení grafu je výpočet $d(i, c)$ pro všechny shluky c kromě shluku $c(i)$.
- Následně určíme minimální průměrnou vzdálenost $b_i = min_{c \neq c(i)} d(i, c)$. Jestliže $b_i = d(i, C)$, pak shluk C bude nejbližším sousedním shlukem shluku $c(i)$ neboli druhým nejlepším shlukem pro i-té pozorování.

- Pro i-té pozorování vypočítáme tzv. hodnotu obrysu s_i, a to

$$s_i = \frac{b_i - a_i}{max\{a_i, b_i\}},$$

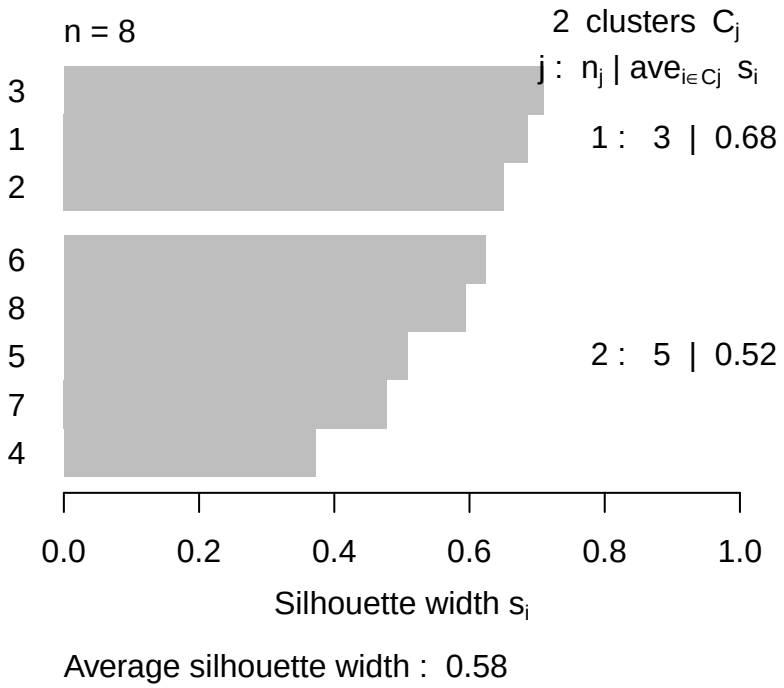
  která nabývá hodnot z $\langle -1, 1 \rangle$. Hodnoty blízké 1 značí, že i-té pozorování leží ve správném shluku, naopak hodnoty blízké -1 nás informují, že bychom měli i-té pozorování přemístit do jiného shluku. Hodnoty blízké 0 značí, že i-té pozorování leží mezi dvěma shluky.

- Graf obrysů shluků je sloupcový graf, kde s_i je délka i-tého sloupce grafu.

Pro ilustraci uvedeme siluetový graf, který přísluší k předchozímu příkladu.

Příklad 2.4. Vycházejme opět ze stejného zadání jako v předchozích příkladech, kde předpokládáme osm pozorování o dvou proměnných $\mathbf{x}_1 = (1, 2)^T$, $\mathbf{x}_2 = (2, 4)^T$, $\mathbf{x}_3 = (1, 4)^T$, $\mathbf{x}_4 = (5, 6)^T$, $\mathbf{x}_5 = (6, 7)^T$, $\mathbf{x}_6 = (4, 8)^T$, $\mathbf{x}_7 = (2, 9)^T$, $\mathbf{x}_8 = (3, 9)^T$ a uvažujeme dva shluky, tj. $K = 2$.

Řešení: Výsledný siluetový graf byl vykreslen opět za využití statistického softwaru R pomocí příkazu $plot(fanny)$, neboť jsme vycházeli z výsledků algoritmu fuzzy shlukování.



Obrázek 4: siluetový graf

Tento výsledný grafický nástroj nám vizuálně opět potvrzuje, že shluk 1 je tvořen pozorováními $\mathbf{x}_1$, $\mathbf{x}_2$ a $\mathbf{x}_3$ a shluk 2 tvoří pozorování $\mathbf{x}_4$, $\mathbf{x}_5$, $\mathbf{x}_6$, $\mathbf{x}_7$ a $\mathbf{x}_8$. Z výsledného siluetového grafu lze dále vyčíst, že průměrná hodnota obrysu pro všechna pozorování $i = 1, \ldots, 8$ je 0.58. Z toho můžeme usoudit, že pozorování jsou umístěna ve správném shluku. Nejvyšší hodnoty obrysu nabývá pozorování $\mathbf{x}_3$, proto si můžeme být u tohoto pozorování téměř jisti, že leží ve správném shluku.

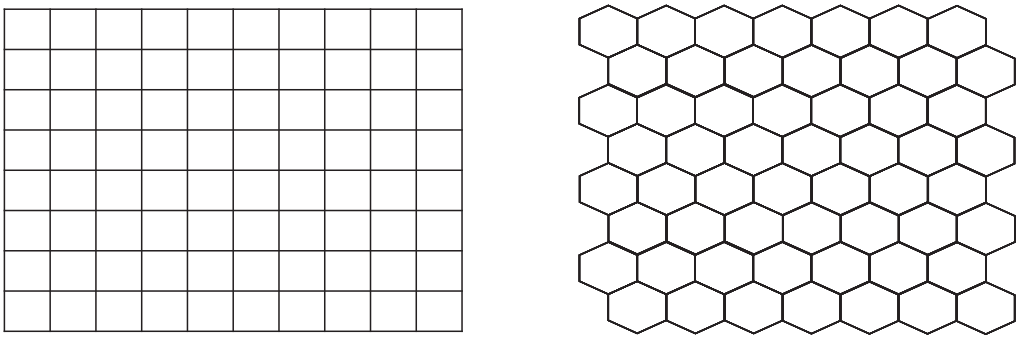
2.4 Samoorganizující mapy

Jedním z dalších přístupů shlukování vícerozměrných dat je využití samoorganizujících map. V literatuře se také často setkáváme s alternativním názvem Kohonenovy mapy, Self-Organizing Maps či metoda SOM. Samoorganizující mapy

vychází z teorie umělých neuronových sítí (touto problematikou se zde však zabývat nebudeme). V současnosti jsou samoorganizující mapy hojně využívány v oblastech medicíny, bioinformatiky, antropologie atp.

Kapitola zabývající se samoorganizujícími mapami byla sepsána za využití literatury [5] a [8].

Primárním cílem této metody je redukce vysoké dimenze vstupních dat na prostor nižší dimenze, nejčastěji na prostor dvou nebo tří dimenzí. Výsledným produktem této metody je mřížka (nebo síť) tvořená velkým počtem uzlů (nebo umělých neuronů). V případě dvou dimenzí mohou být takové uzly znázorněny do čtvercové, obdélníkové či šestiúhelníkové sítě. Na základě těchto uzlů jsme dále schopni identifikovat shluky.



Obrázek 5: čtvercová a šestiúhelníková síť

Předpokládejme například dvoudimenzionální mřížku, kde $\mathcal{K}_1 = \{1, 2, \ldots, K_1\}$ je množina řádků a $\mathcal{K}_2 = \{1, 2, \ldots, K_2\}$ je množina sloupců, kde výška mřížky K_1 a délka mřížky K_2 jsou zvoleny uživatelem. Potom uzel sítě je dvojice $(l_1, l_2) \in (\mathcal{K}_1 \times \mathcal{K}_1)$ a celkový počet uzlů v síti je $K = K_1 K_2$. Tento počáteční velký počet uzlů se během algoritmu zmenšuje. Pro další práci s jednotlivými algoritmy metody SOM je potřeba jednotlivé uzly uspořádat a přeindexovat. Tedy uzel $(l_1, l_2) \in (\mathcal{K}_1 \times \mathcal{K}_1)$ opatříme indexem $k = (l_1 - 1)K_2 + l_2 \in \mathcal{K}$, kde $\mathcal{K} = \{1, 2, \ldots, K\}$.

Nyní si představíme dva algoritmy metody samoorganizujících map, přičemž si všimneme značných podobností těchto algoritmů s metodou k průměrů. Oba po-

stupy totiž na počátku svého algoritmu volí reprezentanty daných skupin a v průběhu algoritmu jednotlivá pozorování umisťují do shluků, které obsahují nejbližšího reprezentanta k danému pozorování. Označme reprezentanta k-tého shluku jako $\mathbf{m}_k \in \mathbb{R}^p$, kde $k \in K$. Obdobně jako v metodě k průměrů jsou reprezentanti jednotlivých shluků na počátku zvoleni náhodně.

2.4.1 On-line algoritmus

Hlavní myšlenkou on-line algoritmu je postupné zařazování jednotlivých pozorování do shluků. Jinak řečeno se v každém kroku zabýváme jediným pozorováním, a to v náhodném pořadí. V následujících řádcích se s algoritmem on-line seznámíme po jednotlivých krocích.

- V prvním kroku on-line algoritmu zvolíme velikost sítě, tedy zvolíme K_1 a K_2 a náhodně zvolíme počáteční množinu reprezentantů $\{\mathbf{m}_k\}$ pro všechna $k \in K$. V dalších krocích budeme předpokládat, že vstupní vektor $\mathbf{x}$ je náhodně vybrán ze vstupní množiny pozorování.

- Následně spočítáme euklidovskou vzdálenost mezi vektorem $\mathbf{x}$ a vektorem reprezentantů $\mathbf{m}_k$ pro všechna $k \in K$, přičemž hledáme minimum z těchto vzdáleností, tj.

  $$k^\star = \ min_k\{\|\mathbf{x} - \mathbf{m}_k\|\}.$$

  Pak $\mathbf{m}_{k^\star}$ je nejbližší reprezentant a $k^\star$ je vítězný shluk vstupního vektoru $\mathbf{x}$.

- Nyní se podíváme, které shluky sousedí s vítězným shlukem. Řekneme, že shluk $k' \in K$ je síťový soused shluku $k^\star \in K$, jestliže euklidovská vzdálenost mezi $\mathbf{m}_{k'}$ a $\mathbf{m}_{k^\star}$ je menší než nějaký předem zvolený práh $c > 0$. Množinu takových blízkých shluků nazýváme sousedstvím vítězného shluku a značíme $N_c(k^\star)$.

- V dalších krocích budeme aktualizovat reprezentanty všech sousedních shluků. Jednotlivé vektory $\mathbf{m}_k$ pro $k \in N_c(k^\star)$ aktualizujeme pomocí následujícího

vztahu

$$\mathbf{m}_k + \alpha(\mathbf{x} - \mathbf{m}_k) \rightarrow \mathbf{m}_k, \quad k \in N_c(k^\star),$$

kde váha $\alpha \in (0, 1)$. Pro $k \notin N_c(k^\star)$ bychom dosadili $\alpha = 0$ a $\mathbf{m}_k$ by tak zůstalo nezměněno. Tento proces opakujeme několikrát, minimálně se však doporučuje 500-krát.

Další možnou alternativou pro upravení reprezentantů $\mathbf{m}_k$ za pomocí vah je následující vztah

$$\mathbf{m}_k + \alpha h_k(\mathbf{x} - \mathbf{m}_k), \quad k \in N_c(k^\star),$$

kde funkce h závisí na tom, jak jsou sousední reprezentanti blízko vítěznému $\mathbf{m}_{k^\star}$. Funkce h nabývá hodnoty 1 v případě nulové vzdálenosti a naopak hodnoty funkce h se přibližují k 0 s rostoucí vzdáleností, tedy pro $k \notin N_c(k^\star)$ položíme $h_k = 0$. Zřejmě nejpopulárnější tvar funkce h je následující

$$h_k = exp\left\{-\frac{\|\mathbf{m}_k - \mathbf{m}_{k^\star}\|}{2\sigma^2}\right\},$$

kde $k \in N_c(k^\star)$ a $\sigma > 0$. Hodnoty c, α i σ volí sám uživatel. Jak již bylo řečeno výše, α nabývá hodnot z $(0, 1)$, přičemž s rostoucím počtem iterací t algoritmu se hodnoty blíží k 0. Lze přitom využít jednu z následujících formulí funkce $\alpha(t)$:

lineární

$$\alpha(t) = \alpha_0\left(1 - \frac{t}{T}\right),$$

mocninná

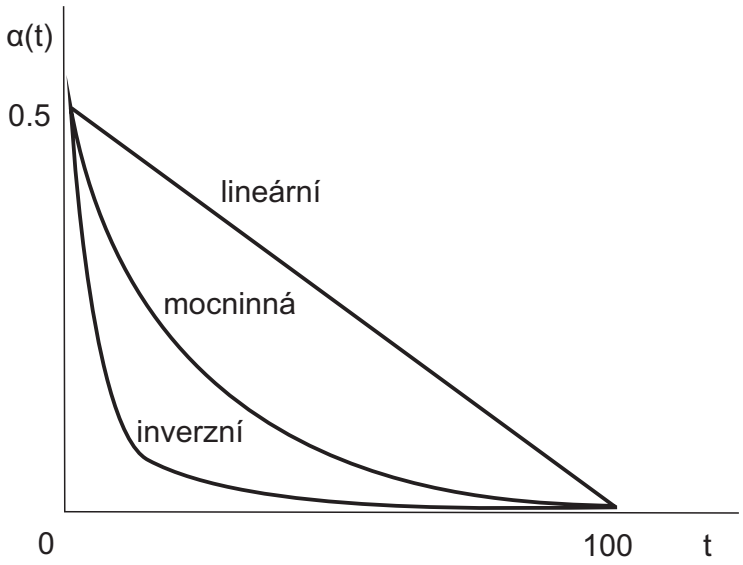
$$\alpha(t) = \alpha_0\left(\frac{0.005}{\alpha_0}\right)^{\frac{t}{T}},$$

inverzní

$$\alpha(t) = \frac{\alpha_0}{(1 + \frac{100t}{T})},$$

kde α_0 je počáteční hodnota a kde T je celkový počet iterací.

Následující obrázek 6 znázorňuje tvar zmíněných funkcí $\alpha(t)$ při počáteční hodnotě $\alpha(0) = 0.5$ a celkovém počtu iterací $T = 100$.



Obrázek 6: grafické zobrazení funkce $\alpha(t)$

2.4.2 Hromadný algoritmus

Hromadný algoritmus (nebo také batch algoritmus) se od výše popsaného postupu liší tím, že pracuje se všemi daty zároveň, proto je tento postup také rychlejší. Nyní si ukážeme jednotlivé kroky tohoto algoritmu.

- Obdobně jako v předchozím postupu je třeba v prvním kroku náhodně zvolit počáteční množinu reprezentantů $\{\mathbf{m}_k\}$ pro všechna $k \in K$.
- Následně ke k-tému shluku, kde $k \in K$, přiřadíme všechna pozorování $\mathbf{x}_i$, jejichž $\mathbf{m}_{k^\star} \in N_c(k)$.
- Na konci algoritmu ještě zaktualizujeme reprezentanty jednotlivých shluků zprůměrováním všech pozorování z předchozího kroku. Přitom lze použít vážený průměr, kde váhy $\{h_{ik^\star}\}$ jsou definovány stejně jako v algoritmu on-line (tj. vztahem pro h_k). Tento proces několikrát zopakujeme.

Grafickým výstupem tohoto algoritmu jsou kruhy znázorňující jednotlivé uzly, v nichž jsou náhodně umístěna příslušná pozorování. Přirozené skupiny těchto pozorování mohou být navíc rozlišeny barevně a poté lze intuitivně najít shluky.

Dalším možným grafickým výstupem metod samoorganizujících map je **U**-matice. Značení vychází z anglického Unified distance. Jednotlivé vstupy matice jsou tvořeny euklidovskými vzdálenostmi mezi sousedícími reprezentanty. Předpokládejme například pět uzlů s množinou reprezentantů $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4, \mathbf{m}_5\}$. Potom **U**-matice bude vektor o rozměru (1×9)

$$\mathbf{U} = (u_1, u_{12}, u_2, u_{23}, u_3, u_{34}, u_4, u_{45}, u_5),$$

kde $u_{ij} = \|\mathbf{m}_i - \mathbf{m}_j\|$, kde $i, j = 1, \ldots, 5$ a $i \neq j$, je euklidovská vzdálenost mezi sousedními i-tým a j-tým reprezentantem a $u_i = \frac{u_{i-1,i} - u_{i,i+1}}{2}$. Malé hodnoty **U**-matice přitom ukazují na blízkost dvou sousedních uzlů.

Řada statistických softwarů umožňuje vykreslení **U**-matice do prostoru o dvou i třech dimenzích. Například v prostoru dvou dimenzí jsou shluky, hranice mezi jednotlivými shluky i pozorování znázorněny pomocí určitých barev. Prostor tří dimenzí hranice mezi jednotlivými shluky znázorňuje zvýšenými hřebeny.

3 Metoda dílčích nejmenších čtverců

Další významnou moderní metodou mnohorozměrné statistiky je metoda PLS[2] neboli metoda dílčích nejmenších čtverců, o níž se poprvé zmínil statistik Herman Wold v roce 1975. Na vývoji této metody se později podílel i jeho syn Svante Wold, který našel uplatnění metody PLS v chemometrii. Právě v tomto oboru je metoda dodnes nejvíce využívána. Specifikum chemometrických dat je přitom zejména v tom, že počet proměnných výrazně přesahuje počet pozorování.

Při zpracování kapitoly o metodě dílčích nejmenších čtverců a jejích algoritmech byly využity především zdroje [9], [11], [12], [13] a dále také [1], [2], [3], [5] a [7].

3.1 Výchozí metody

Metodu PLS lze použít zejména pro řešení regresních problémů. V takovém případě metodu chápeme jako spojení metody hlavních komponent a mnohonásobné, případně mnohorozměrné lineární regrese. Proto se nejprve zaměřím na tyto pomocné metody jednotlivě.

3.1.1 Metoda hlavních komponent

V současnosti velice populární metoda hlavních komponent byla zavedena Karlem Pearsonem již v roce 1901 jako nástroj k redukci mnohorozměrných dat. V dalších letech docházelo k jejímu zobecňování. Nyní je metoda velmi oblíbená a může být považována za startovací krok u mnoha jiných mnohorozměrných metod, neboť redukce mnohorozměrných dat zjednodušuje jejich následnou analýzu.

Jak již bylo naznačeno, cílem metody hlavních komponent je redukce dimenze dat, tedy nahrazení velkého nepřehledného počtu původních proměnných menším počtem proměnných tak, aniž by došlo k velké ztrátě informace. Tyto nově vytvořené proměnné jsou nejčastěji umělé proměnné, tzv. hlavní komponenty. Jedná se o lineární kombinace původních proměnných a hlavní komponenty musí

[2]Název metody PLS vychází z anglického Partial Least Squares.

být vzájemně nekorelované. Algoritmus metody vytváří nové proměnné postupně a s klesající důležitostí, která je ztotožněna s rozptylem příslušné hlavní komponenty. Nejvýznamnější je první hlavní komponenta, která vysvětluje největší část variability původních proměnných. V praxi se nejčastěji setkáváme se situací, kdy jsou původní proměnné nahrazeny dvěma nebo třemi hlavními komponentami, které vysvětlují asi 70 až 80 procent variability původních proměnných (přitom ovšem záleží na počtu původních proměnných).

Při tvorbě kapitoly o metodě hlavních komponent byly využity zejména zdroje [1], [2], [3] a [7].

Předpokládejme náhodný vektor $\mathbf{X} = (X_1, \ldots, X_p)^T$ s varianční maticí $\mathbf{V} = (\sigma_{ij})$, kde $i, j = 1, \ldots, p$, s p kladnými vzájemně různými charakteristickými čísly $\lambda_1 > \lambda_2 > \ldots > \lambda_p > 0$, která odpovídají ortonormálním charakteristickým vektorům $\mathbf{v}_1, \ldots, \mathbf{v}_p$. Našim cílem bude zredukovat $X_1, \ldots, X_p$ na menší počet náhodných veličin, které by byly co nejlepší náhradou celého náhodného vektoru $\mathbf{X}$. Hledáme tedy takovou lineární kombinaci $\mathbf{c}^T\mathbf{X}$, která vyčerpává co největší část variability vektoru $\mathbf{X}$. Jinak řečeno hledáme takový vektor $\mathbf{c} \in \mathbb{R}^p : \mathbf{c}^T\mathbf{c} = 1$, aby náhodná veličina $\mathbf{c}^T\mathbf{X}$ měla co největší rozptyl

$$var(\mathbf{c}^T\mathbf{X}) = \mathbf{c}^T\mathbf{V}\mathbf{c}.$$

Maximalizujeme tedy výraz $\mathbf{c}^T\mathbf{V}\mathbf{c}$ za podmínky $\mathbf{c}^T\mathbf{c} = 1$. Dále si uvědomme následující vlastnost charakteristických čísel symetrické pozitivně definitní matice $\mathbf{A}$.

Věta 3.1. *Nechť je dána symetrická pozitivně definitní matice $\boldsymbol{A}$ o rozměru $(p \times p)$ s charakteristickými čísly $\lambda_1 > \lambda_2 > \ldots > \lambda_p$ příslušící ortonormálním charakteristickým vektorům $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$. Pak pro každý vektor $\boldsymbol{x} \in \mathbb{R}^p$ splňující $\boldsymbol{x}^T\boldsymbol{x} = 1$ platí*

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq \lambda_1.$$

Důkaz této věty lze nalézt např. v [1].

Z této vlastnosti plyne, že maximální hodnota $\mathbf{c}^T\mathbf{V}\mathbf{c}$ je λ_1, která je dosahována

při $\mathbf{c} = \mathbf{v}_1$. Získali jsme tak náhodnou veličinu

$$Z_1 = \mathbf{v}_1{}^T\mathbf{X},$$

kterou nazýváme první hlavní komponenta a kde $var(Z_1) = \lambda_1$.

Nyní je třeba zhodnotit, nakolik první hlavní komponenta vysvětluje chování celého vektoru $\mathbf{X}$. Jestliže podíl rozptylů $\frac{\lambda_1}{\lambda_1+\ldots+\lambda_p}$ je dostatečně blízký jedné, pak lze říci, že Z_1 dostatečně dobře vysvětluje vektor $\mathbf{X}$. Pokud je však zmíněný podíl spíše blízký nule, musíme v algoritmu pokračovat.

Hledáme tedy analogicky další lineární kombinaci $\mathbf{c}^T\mathbf{X}$, která bude nekorelovaná s veličinou Z_1. Proto obecně pro $i \neq j$, kde $i, j = 1, \ldots, p$,

$$cov(Z_i, Z_j) = 0.$$

Přitom se opět snažíme, aby nová i-tá hlavní komponenta Z_i měla co největší rozptyl, uvedeným postupem obdržíme $var(Z_i) = \lambda_i$, kde $i = 1, \ldots, p$.

Při každém kroku tohoto algoritmu je třeba vždy zhodnotit, nakolik všechny vyjádřené hlavní komponenty (prostřednictvím vysvětlené variability) dostatečně vysvětlují chování vektoru $\mathbf{X}$.

Nyní si ukážeme, jak lze vytvořené komponenty v případě výběrové obdoby metody hlavních komponent shrnout do jednoho výrazu. Předpokládejme matici $\mathbf{X}$ o rozměru $(n \times p)$, kde n je počet pozorování a p je počet proměnných. Nechť dále $\overline{\mathbf{x}}$ je výběrový průměr a $\Sigma_{\mathbf{XX}}$ je výběrová varianční matice. Využijme dále singulárního rozkladu matice $\Sigma_{\mathbf{XX}}$, tj.

$$\Sigma_{\mathbf{XX}} = \mathbf{U}\Lambda\mathbf{U}^T,$$

kde sloupce matice $\mathbf{U}$ jsou tvořeny charakteristickými vektory z $\Sigma_{\mathbf{XX}}$ a diagonální matice Λ je tvořena vlastními čísly matice $\Sigma_{\mathbf{XX}}$.

Pak lze matici $\mathbf{X}$ nahradit maticí $\mathbf{Z}$ výběrových hlavních komponent ve tvaru

$$\mathbf{Z} = (\mathbf{X} - \mathbf{1}_n\overline{\mathbf{x}}^T)\mathbf{U},$$

jejíž sloupce nazýváme vektory skórů a sloupce matice $\mathbf{U}$ nazýváme zátěže j-té komponenty, které reprezentují vliv původních proměnných na nově vytvořené

komponenty. Matice $\mathbf{Z}$ je stejné dimenze jako původní matice $\mathbf{X}$, tedy $(n \times p)$, ale protože cílem metody hlavních komponent je redukce proměnných, pracujeme nadále jen s několika prvními komponentami, které vysvětlují největší část informace původních proměnných.

3.1.2 Mnohonásobná lineární regrese

První zmínky o regresní analýze pochází z roku 1885, kdy Francis Galton využil předchozích znalostí o metodě nejmenších čtverců a pokusil se o studii závislosti výšky postavy rodičů a jejich dětí. Od této doby regresní analýza prochází velkým rozvojem. Vedle lineární regrese byla vyvinuta i nelineární regrese, kterou se zde však zabývat nebudeme. Nejjednodušší formou regrese je situace, kdy jediná výstupní proměnná závisí na jediné vstupní proměnné. Nyní zde popíšu nejčastější podobu regrese, tzv. mnohonásobnou lineární regresi.

Kapitola o mnohonásobné lineární regresi byla sepsána za pomocí zdrojů [5] a [13].

Předpokládejme, že výstupní neboli závislá náhodná proměnná Y je lineárně závislá na vstupních neboli nezávislých nenáhodných proměnných $x_1, \ldots, x_p$. Pak pro i-té pozorování uvažujeme model

$$Y_i = b_0 + \sum_{j=1}^{p} b_j x_{ij} + e_i, \qquad i = 1, \ldots, n,$$

kde e_i je náhodná chyba se střední hodnotou 0 a rozptylem $\sigma^2 > 0$ a kde $b_0, b_1, \ldots, b_p$ jsou neznámé parametry. Linearita tohoto regresního modelu přitom vychází z linearity těchto neznámých parametrů.

Cílem metody lineární regrese je odhadnout hodnoty parametrů $b_0, b_1, \ldots, b_p$ a σ^2, abychom byli schopni ohodnotit vliv jednotlivých nezávislých proměnných na závisle proměnnou Y, případně počet nezávislých proměnných zredukovat. Odhad těchto neznámých parametrů získáváme metodou nejmenších čtverců. Tato velice známá metoda odhady parametrů určuje minimalizací čtvercových odchy-

lek skutečných hodnot od těch odhadnutých,

$$\sum_{i=1}^{n}(Y_i - b_0 - b_1 x_{i1} - b_2 x_{i2} - \ldots - b_p x_{ip})^2 \to min.$$

Dalším cílem této regresní metody může být predikce hodnoty Y na základě budoucích hodnot $x_1, \ldots, x_p$ a vyhodnocení přesnosti této predikce.

Poznámka 3.1. Vedle mnohonásobné regrese se můžeme setkat i s mnohorozměrnou regresí, která se liší v tom, že předpokládáme kromě jedné či více nezávislých proměnných i více závislých proměnných.

3.2 Regresní metoda PLS

Jak již bylo řečeno, metodu PLS lze využít k řešení regresních problémů neboli k modelování vztahu závislosti mezi dvěma bloky proměnných. Předpokládejme obecně, že $\mathbb{R}^m$ je m-rozměrný prostor nezávisle proměnných a $\mathbb{R}^q$ je q-rozměrný prostor závisle proměnných. Naměřením n pozorování jsme pak schopni pomocí metody PLS identifikovat vztah mezi proměnnými z těchto dvou bloků.

Jednodušší podobou metody PLS je metoda PLS1, kdy uvažujeme pouze jedinou závisle proměnnou. Tedy předpokládejme matici $\mathbf{X}$ nezávisle proměnných rozměru $(n \times m)$ a n-rozměrný vektor $\mathbf{y}$ závisle proměnné. Jinak řečeno, předpokládáme n pozorování, u kterých sledujeme m vlastností. Cílem metody, označované jako PLS1, je pak za využití metody hlavních komponent a mnohonásobné regrese kvantifikovat vztah

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e},$$

kde $\mathbf{b}$ je vektor regresních koeficientů a $\mathbf{e}$ je náhodný vektor chyb, maticově tedy

$$\begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} x_{11} & \ldots x_{1m} \\ \vdots & \ddots \vdots \\ x_{n1} & \ldots x_{nm} \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} + \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix}.$$

Úlohu můžeme zobecnit tím, že budeme předpokládat $(n \times q)$-rozměrnou matici $\mathbf{Y}$ závisle proměnných. Tuto situaci řešíme pomocí metody PLS2, kterou

obdobně chápeme jako propojení metody hlavních komponent a tentokrát mnohorozměrné regresní analýzy. Cílem je zde opět analogicky najít vztah

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E},$$

kde $\mathbf{B}$ je $(m \times q)$-rozměrná matice regresních koeficientů a $\mathbf{E}$ je $(n \times q)$-rozměrná matice chyb, rozepsáno

$$\begin{pmatrix} Y_{11} & \dots Y_{1q} \\ \vdots & \ddots \vdots \\ Y_{n1} & \dots Y_{nq} \end{pmatrix} = \begin{pmatrix} x_{11} & \dots x_{1m} \\ \vdots & \ddots \vdots \\ x_{n1} & \dots x_{nm} \end{pmatrix} \begin{pmatrix} b_{11} & \dots b_{1q} \\ \vdots & \ddots \vdots \\ b_{m1} & \dots b_{mq} \end{pmatrix} + \begin{pmatrix} e_{11} & \dots e_{1q} \\ \vdots & \ddots \vdots \\ e_{n1} & \dots e_{nq} \end{pmatrix}.$$

Ještě před vyřešením tohoto vztahu (tj. před odhadem regresních koeficientů) je třeba si uvědomit, že matice $\mathbf{X}$ i $\mathbf{Y}$, jejichž sloupce jsou z důvodu snadnějšího značení dále centrovány, jsou modelovány umělými komponentami opět s využitím regresního modelu, pracujeme tedy s následujícími dekompozicemi matic $\mathbf{X}$ a $\mathbf{Y}$

$$\mathbf{X} = \mathbf{TP}^T + \mathbf{E_X},$$

$$\mathbf{Y} = \mathbf{UQ}^T + \mathbf{E_Y},$$

kde $\mathbf{E_X}$ o rozměru $(n \times m)$ a $\mathbf{E_Y}$ o rozměru $(n \times q)$ jsou matice chyb a kde $(n \times a)$-rozměrné matice skórů $\mathbf{T}$, resp. $\mathbf{U}$ jsou tvořené lineárními kombinacemi původních nezávislých, resp. závislých proměnných a matice $\mathbf{P}$ o rozměru $(m \times a)$ a $\mathbf{Q}$ o rozměru $(q \times a)$ jsou matice zátěží. Všechny tyto matice $\mathbf{T}$, $\mathbf{U}$, $\mathbf{P}$ i $\mathbf{Q}$ mají a sloupců, kde a je daný počet komponent a platí $a \leq min\{m, n, q\}$.

Pro nalezení regresního vztahu budeme maximalizovat kovarianci mezi složkami, reprezentovanými sloupci matic $\mathbf{X}$ a $\mathbf{Y}$, neboť právě kovariance kombinuje vysoký rozptyl jednotlivých složek v matici $\mathbf{X}$ a vysokou korelaci mezi složkami z $\mathbf{X}$ a $\mathbf{Y}$. Kovarianci mezi vektory $\mathbf{t}$ a $\mathbf{u}$, kde $\mathbf{t} = \mathbf{Xw}$ a $\mathbf{u} = \mathbf{Yc}$, můžeme odhadnout například pomocí vztahu $\frac{\mathbf{t}^T\mathbf{u}}{n-1}$ nebo $\frac{\mathbf{t}^T\mathbf{u}}{n}$ za podmínek $\|\mathbf{t}\| = \|\mathbf{u}\| = 1$, které nám zajistí jednoznačnost řešení:

$$cov(\mathbf{Xw}, \mathbf{Yc}) \rightarrow max \qquad \|\mathbf{t}\| = \|\mathbf{Xw}\| = 1, \quad \|\mathbf{u}\| = \|\mathbf{Yc}\| = 1.$$

Maximalizací uvedené kovariance za daných podmínek pro vektory $\mathbf{t}$ a $\mathbf{u}$ získáme vektory $\mathbf{t}_1$ a $\mathbf{u}_1$.

Analogickou maximalizací kovariance za stejných podmínek bychom získali další vektory, které však musí být navíc ortonormální k předchozím získaným vektorům, tedy $\mathbf{t}_j^T\mathbf{t}_l = 0$ a $\mathbf{u}_j^T\mathbf{u}_l = 0$ pro $1 \leq j \leq l \leq a$.

Dalším krokem u téměř všech algoritmů metody dílčích nejmenších čtverců je odhadnutí vektorů zátěží ze vztahů pro dekompozice matic $\mathbf{X}$ a $\mathbf{Y}$, a to

$$\mathbf{p}_1 \quad \text{a} \quad \mathbf{q}_1.$$

Přesné vztahy pro $\mathbf{p}_1$ a $\mathbf{q}_1$ budou ukázány v jednotlivých algoritmech.

Tyto kroky analogicky opakujeme pro všechna $j = 2, \ldots, a$. Přitom většina algoritmů vyžaduje na konci tohoto kroku očištění matice $\mathbf{X}$. Přesný vztah pro očištění matice $\mathbf{X}$ bude následně uveden v jednotlivých algoritmech.

Vraťme se nyní zpět k hlavnímu cíli regresní metody PLS. Hlavním úkolem této metody je vysvětlení vztahu mezi proměnnými z matic $\mathbf{X}$ a $\mathbf{Y}$ a predikování hodnot z $\mathbf{Y}$. Využijme proto předchozích vztahů a následujícího rozkladu matice $\mathbf{U}$ do lineárního vztahu

$$\mathbf{U} = \mathbf{TD} + \mathbf{H},$$

kde $\mathbf{D}$ je čtvercová diagonální matice s prvky $d_1, \ldots, d_a$ a $\mathbf{H}$ je $(n \times a)$-rozměrná matice chyb. Nyní jsme schopni predikovat $\mathbf{Y}$, tedy

$$\mathbf{Y} = \mathbf{UQ}^T + \mathbf{E_Y} = (\mathbf{TD} + \mathbf{H})\mathbf{Q}^T + \mathbf{E_Y} = \mathbf{TC}^T + \mathbf{E}^\star,$$

kde $\mathbf{C}^T = \mathbf{DQ}^T$ je $(a \times q)$-rozměrná matice regresních koeficientů a $\mathbf{E}^\star = \mathbf{E_Y} + \mathbf{HQ}^T$ je matice reziduí.

Poznámka 3.2. V následujících kapitolách budou popsány jednotlivé algoritmy pro řešení regresní metody PLS. Ještě předtím je však třeba uvést poznámku o značení. Obdobně jako v dostupné literatuře věnující se metodě PLS zde nebudeme odlišovat odhady a teoretické hodnoty regresních parametrů. Odhady parametrů označíme stříškou jedině tam, kde by mohlo dojít k nedorozumění.

3.2.1 Algoritmus NIPALS

Historicky prvním zveřejněným algoritmem, který řeší problém regrese metodou dílčích nejmenších čtverců, je algoritmus NIPALS. Název algoritmu NIPALS vychází z anglického Nonlinear Iterative Partial Least Squares. Nyní si tento algoritmus představíme postupně.

Předpokládejme, že $\mathbf{u}_1$ je inicializován například prvním sloupcem matice $\mathbf{Y}$.

- V prvním kroku algoritmu NIPALS určíme vektor $\mathbf{w}_1$, kde

$$\mathbf{w}_1 = \mathbf{X}^T\mathbf{u}_1(\mathbf{u}_1^T\mathbf{u}_1)^{-1} \quad \text{splňuje podmínku} \quad \mathbf{w}_1 = \frac{\mathbf{w}_1}{\|\mathbf{w}_1\|}.$$

  Dále určíme příslušný vektor skóre $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1$.

- Obdobně určíme vektor $\mathbf{c}_1$, kde

$$\mathbf{c}_1 = \mathbf{Y}^T\mathbf{t}_1(\mathbf{t}_1^T\mathbf{t}_1)^{-1} \quad \text{splňuje podmínku} \quad \mathbf{c}_1 = \frac{\mathbf{c}_1}{\|\mathbf{c}_1\|}.$$

  Rovněž zde určíme příslušnou lineární kombinaci $\mathbf{u}_1^\star = \mathbf{Y}\mathbf{c}_1$, navíc

$$\mathbf{u}_\triangle = \mathbf{u}_1^\star - \mathbf{u}_1,$$

$$\triangle\mathbf{u} = \mathbf{u}_\triangle^T\mathbf{u}_\triangle.$$

  Pokud $\triangle\mathbf{u} < \varepsilon$, kde ε je dostatečně malé, například 10^{-6}, budeme pokračovat v dalších krocích algoritmu. V opačném případě položíme $\mathbf{u}_1 = \mathbf{u}_1^\star$ a celý algoritmus zopakujeme od prvního kroku.

- Následně vypočítáme vektory zátěží $\mathbf{p}_1$ a $\mathbf{q}_1$,

$$\mathbf{p}_1 = \mathbf{X}^T\mathbf{t}_1(\mathbf{t}_1^T\mathbf{t}_1)^{-1},$$

$$\mathbf{q}_1 = \mathbf{Y}^T\mathbf{u}_1(\mathbf{u}_1^T\mathbf{u}_1)^{-1}.$$

- Poté algoritmus NIPALS vyžaduje očištění jak matice $\mathbf{X}$, tedy

$$\mathbf{X}_1 = \mathbf{X} - \mathbf{t}_1\mathbf{p}_1^T.$$

- Všechny tyto kroky algoritmu opakujeme pro všechna $j = 2, \ldots, a$, přičemž vždy pracujeme s již očištěnou maticíi $\mathbf{X}_j$.

- Na závěr vypočítáme odhady regresních parametrů z původního regresního vztahu $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, tedy

$$\widehat{\mathbf{B}} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{C}^T,$$

  kde matice $\mathbf{W}$, $\mathbf{P}$ a $\mathbf{C}$ jsou matice tvořené sloupcovými vektory $\mathbf{w}_j$, $\mathbf{p}_j$ a $\mathbf{c}_j$, kde $j = 1, \ldots, a$.

Takto popsaný algoritmus nyní aplikujeme na jednoduchý příklad.

Příklad 3.1. Nechť jsou dána tři pozorování, u kterých sledujeme čtyři proměnné. Tedy předpokládejme matici nezávislých proměnných $\mathbf{X}$ o rozměru (3×4) a 3-rozměrný vektor $\mathbf{y}$ závisle proměnné

$$\mathbf{X} = \begin{pmatrix} 3 & 1 & 2 & 2 \\ 5 & 2 & 3 & 4 \\ 7 & 0 & 1 & 4 \end{pmatrix},$$

$$\mathbf{y} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}.$$

Zjistěme vztah mezi proměnnými z matice $\mathbf{X}$ a vektoru $\mathbf{y}$ pomocí algoritmu NIPALS.

Řešení: Celý algoritmus byl vyřešen pomocí statistického softwaru R v knihovně *chemometrics*. Pomocí příkazu

$$pls1\_nipals(X, y, a = 2),$$

kde $a = 2$ vyjadřuje požadovaný počet komponent, jsme získali výslednou matici zátěží $\mathbf{P}$, matici skórů $\mathbf{T}$ a dále matice $\mathbf{W}$ a $\mathbf{C}$:

$$\mathbf{P} = \begin{pmatrix} 0.8466 & -0.1703 \\ -0.2540 & -0.5959 \\ -0.2540 & -0.5959 \\ 0.3951 & -0.5108 \end{pmatrix},$$

$$\mathbf{T} = \begin{pmatrix} -2.1758 & 0.9274 \\ -0.3108 & -1.5457 \\ 2.4867 & 0.6183 \end{pmatrix},$$

$$\mathbf{W} = \begin{pmatrix} 0.8393 & -0.1703 \\ -0.2798 & -0.5959 \\ -0.2798 & -0.5959 \\ 0.3730 & -0.5108 \end{pmatrix},$$

$$\mathbf{C} = (2.1541 \quad 0.1622).$$

Vedle toho jsme dále pomocí uvedeného příkazu získali vektor regresních koeficientů **b**, který nás informuje právě o vztahu mezi nezávisle proměnnými a závisle proměnnou:

$$\mathbf{b} = \begin{pmatrix} 1.7861 \\ -0.7013 \\ -0.7013 \\ 0.7233 \end{pmatrix}.$$

Z tohoto výsledného vektoru regresních koeficientů lze usoudit, že největší vliv na hodnoty závisle proměnné má proměnná x_1. Menší vliv na hodnoty závisle proměnné má také proměnná x_4 a vliv proměnných x_2 a x_3 je ještě méně významný.

Poznámka 3.3. Možnou modifikací algoritmu NIPALS je algoritmus O-PLS, jehož název vychází z anglického Orthogonal Projections to Latent Structures. Tento algoritmus patří mezi nejmodernější postupy metody PLS (algoritmus byl představen Tryggem a Woldem v roce 2002).

Hlavní myšlenkou postupu O-PLS je rozdělení matice vstupních proměnných **X** na dvě části. První část je tvořena těmi proměnnými, které nejsou ortogonální k výstupním proměnným z **Y**, druhá část je potom tvořena proměnnými z **X**, které jsou ortogonální k proměnným z **Y**, tj. jsou nekorelované. Matici **X** proto můžeme vyjádřit pomocí následující dekompozice

$$\mathbf{X} = \mathbf{T}_p\mathbf{P}_p^T + \mathbf{T}_o\mathbf{P}_o^T + \mathbf{E},$$

kde $\mathbf{T}_0$, resp. $\mathbf{P}_0$ reprezentují skóry, resp. zátěže ortogonální části $\mathbf{X}$ a pro predikci $\mathbf{Y}$ vycházíme pouze z $\mathbf{T}_p$ a $\mathbf{P}_p$. Algoritmus tedy pracuje pouze s očištěnou maticí $\mathbf{X} - \mathbf{T}_0\mathbf{P}_0^T$.

3.2.2 Jádrový algoritmus

Jedním z dalších možných přístupů, jak vyřešit regresní úlohu metodou dílčích nejmenších čtverců, je jádrový algoritmus (v cizojazyčné literatuře se setkáváme s názvem Kernel algorithm). Tento postup byl zveřejněn v roce 1993 Fredrikem Lindgrenem a kol. Základy tohoto algoritmu stojí na dekompozici výrazu $\mathbf{X}^T\mathbf{Y}$ na vlastní vektory. V následujících bodech si představíme jednotlivé kroky tohoto algoritmu.

- V prvním kroku jádrového algoritmu určíme vlastní vektory

  $$\mathbf{w}_1 - \text{přísluší největšímu vlastnímu číslu výrazu} \quad \mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X},$$

  $$\mathbf{c}_1 - \text{přísluší největšímu vlastnímu číslu výrazu} \quad \mathbf{Y}^T\mathbf{X}\mathbf{X}^T\mathbf{Y}.$$

  Přitom oba tyto vektory splňují podmínku $\|\mathbf{X}\mathbf{w}_1\| = \|\mathbf{Y}\mathbf{c}_1\| = 1$. Tímto jsme tedy našli lineární kombinace $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1$ a $\mathbf{u}_1 = \mathbf{Y}\mathbf{c}_1$.

- Následně jsme schopni vypočítat odhad vektorů zátěží z výrazu $\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}_\mathbf{X}$, a to pomocí vztahu

  $$\mathbf{p}_1^T = (\mathbf{t}_1^T\mathbf{t}_1)^{-1}\mathbf{t}_1^T\mathbf{X}.$$

- Pro opakování předchozích kroků algoritmu a vyhledání dalších komponent je třeba matici $\mathbf{X}$ očistit, tedy

  $$\mathbf{X}_1 = \mathbf{X} - \mathbf{t}_1\mathbf{p}_1^T.$$

- Další komponenty $\mathbf{t}_j$ a $\mathbf{p}_j$, kde $j = 2, \ldots, a$, najdeme stejným postupem jako $\mathbf{t}_1$ a $\mathbf{p}_1$, přičemž vždy pracujeme s již očištěnou maticí $\mathbf{X}$. Analogickým

způsobem bychom mohli najít i $\mathbf{u}_j$ a $\mathbf{q}_j$ pro všechna $j = 1, \ldots, a$ pomocí vztahů

$$\mathbf{u}_j = \mathbf{Y}\mathbf{c}_j,$$

$$\mathbf{q}_j^T = (\mathbf{u}_j^T \mathbf{u}_j)^{-1} \mathbf{u}_j^T \mathbf{Y}.$$

Výpočet vektorů $\mathbf{u}_j$ a $\mathbf{q}_j$ pro všechna $j = 1, \ldots, a$ však není nutný pro odhad regresních koeficientů z matice $\mathbf{B}$.

- Závěrečným krokem jádrového algoritmu je výpočet odhadu regresních koeficientů v matici $\mathbf{B}$ z původní úlohy. Tento odhad určíme pomocí stejného vztahu jako v algoritmu NIPALS, tj.

$$\widehat{\mathbf{B}} = \mathbf{W}(\mathbf{P}^T \mathbf{W})^{-1} \mathbf{C}^T,$$

kde matice $\mathbf{W}$, $\mathbf{P}$ a $\mathbf{C}$ jsou matice tvořené sloupcovými vektory $\mathbf{w}_j$, $\mathbf{p}_j$ a $\mathbf{c}_j$, kde $j = 1, \ldots, a$.

Takto popsaný algoritmus je vhodný zejména pro případy, kdy uvažujeme velký počet pozorování n. V situaci, kdy uvažujeme spíše velký počet proměnných p, bychom postupovali následovně.

- V prvním kroku bychom opět určili vlastní vektory

$$\mathbf{w}_1 - \text{přísluší největšímu vlastnímu číslu výrazu} \quad \mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T,$$

$$\mathbf{c}_1 \ - \text{přísluší největšímu vlastnímu číslu výrazu} \quad \mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{X}^T.$$

- V dalších krocích bychom postupovali stejně jako v předchozím algoritmu.

3.2.3 Algoritmus SIMPLS

Jedním z dalších možných postupů regresní metody PLS je algoritmus SIMPLS, jehož název je zkratkou anglického Statistically Inspired Modification of the PLS. Tento algoritmus byl navržen Sijmenem de Jongem v roce 1993.

Zásadní odlišností algoritmu SIMPLS od jádrového algoritmu a algoritmu NIPALS je absence očištění matic $\mathbf{X}$ a $\mathbf{Y}$. Algoritmus SIMPLS však očištění

provádí u matice $\mathbf{S} = \mathbf{X}^T\mathbf{Y}$. Proto se výsledky těchto algoritmů shodují jen po první zjištěnou komponentu.

Nyní si přesný postup algoritmu SIMPLS popíšeme v jednotlivých krocích.

- Nejprve položme
$$\mathbf{S}_0 = \mathbf{X}^T\mathbf{Y}.$$
- Pro $j = 1$ položíme $\mathbf{S}_1 = \mathbf{S}_0$ a spočítáme $\mathbf{w}_1$ jako první vlastní vektor matice $\mathbf{S}_1$ splňující podmínku $\mathbf{w}_1 = \frac{\mathbf{w}_1}{\|\mathbf{w}_1\|}$. Poté jsme schopni vypočítat první komponentu $\mathbf{t}_1$ pomocí známého vztahu
$$\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1$$
za podmínky $\mathbf{t}_1 = \frac{\mathbf{t}_1}{\|\mathbf{t}_1\|}$.
- Dále vypočítáme odhad zátěžového vektoru $\mathbf{p}_1$, tedy
$$\mathbf{p}_1 = \mathbf{X}_1^T\mathbf{t}_1.$$
- Tyto kroky opakujeme pro všechna $j = 2, \ldots, a$, přičemž vždy pracujeme s již očištěnou maticí $\mathbf{S}_j$
$$\mathbf{S}_j = \mathbf{S}_{j-1} - \mathbf{P}_{j-1}(\mathbf{P}_{j-1}^T\mathbf{P}_{j-1})^{-1}\mathbf{P}_{j-1}^T\mathbf{S}_{j-1},$$
kde matice $\mathbf{P}_{j-1} = [\mathbf{p}_1, \ldots, \mathbf{p}_{j-2}]$.
- Nakonec jsme schopni vypočítat odhady parametrů z původního regresního vztahu $\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}$, tedy
$$\widehat{\mathbf{B}} = \mathbf{W}\mathbf{T}^T\mathbf{Y},$$
kde matice $\mathbf{W}$, $\mathbf{T}$ jsou tvořeny sloupcovými vektory $\mathbf{w}_j$, $\mathbf{t}_j$, $j = 1, \ldots, a$.

Příklad 3.2. Vycházejme ze stejného zadání jako v předchozím příkladu. Předpokládejme matici nezávislých proměnných $\mathbf{X}$ o rozměru (3×4) a 3-rozměrný vektor $\mathbf{y}$
$$\mathbf{X} = \begin{pmatrix} 3 & 1 & 2 & 2 \\ 5 & 2 & 3 & 4 \\ 7 & 0 & 1 & 4 \end{pmatrix},$$

$$\mathbf{y} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}.$$

Pomocí algoritmu SIMPLS zjistěme vztah mezi proměnnými z matice $\mathbf{X}$ a vektoru $\mathbf{y}$.

Řešení: Kompletní algoritmus byl vyřešen opět za pomoci statistického softwaru *R*, tentokrát však v knihovně *pls*. Využili jsme přitom následujícího příkazu

$$simpls = mvr(Y \sim X, ncomp = 2, method = "simpls"),$$

kde $ncomp = 2$ je počet požadovaných komponent. Dále jsme použili příkaz

$$b = as.vector(coef(simpls)),$$

který nám již vygeneroval výsledný vektor regresních koeficientů

$$\mathbf{b} = \begin{pmatrix} 0.5333 \\ -0.2333 \\ -0.2333 \\ 0.2000 \end{pmatrix}.$$

Na základě tohoto výsledku bude závěr příkladu obdobný jako v přechozím příkladu. Lze totiž usoudit, že největší vliv na hodnoty závisle proměnné má proměnná x_1. Určitý vliv na hodnoty závisle proměnné má také proměnná x_4 (v kladném smyslu) a proměnné x_2 a x_3 (v záporném smyslu).

Pro zajímavost jsme zkusili celý algoritmus zopakovat s rozdílem, že tentokrát uvažujeme pouze jednu komponentu, tedy opět pomocí příkazů

$$simpls = mvr(Y \sim X, ncomp = 1, method = "simpls"),$$

$$b = as.vector(coef(simpls))$$

jsme získali vektor regresních koeficientů

$$\mathbf{b} = \begin{pmatrix} 0.5447 \\ -0.1816 \\ -0.1816 \\ 0.2421 \end{pmatrix}.$$

Došli jsme k obdobnému výsledku, předchozí závěr lze tedy jen potvrdit.

Poznámka 3.4. V předchozích odstavcích byly uvedeny ty nejznámnější algoritmy pro řešení regresní metody PLS. Kromě nich se však můžeme setkat i s dalšími postupy. Jedním z nich je algoritmus vlastních vektorů, který je jakousi jednodušší formou jádrového algoritmu. Na rozdíl od jádrového algoritmu, kde hledáme vlastní vektory příslušející největším vlastním číslům matic $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$ a $\mathbf{Y}^T\mathbf{X}\mathbf{X}^T\mathbf{Y}$, budeme v algoritmu vlastních vektorů hledat vlastní vektory $\mathbf{p}_1, \ldots, \mathbf{p}_a$, resp. $\mathbf{q}_1, \ldots, \mathbf{q}_a$ příslušející a největším vlastním číslům matic $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$, resp. $\mathbf{Y}^T\mathbf{X}\mathbf{X}^T\mathbf{Y}$. Takto najdeme lineární kombinace $\mathbf{t}_j = \mathbf{X}\mathbf{p}_j$ a $\mathbf{u}_j = \mathbf{X}\mathbf{q}_j$ pro všechna $j = 1, \ldots, a$. Přitom ani nebylo zapotřebí matici $\mathbf{X}$ očistit.

Více informací o těchto a dalších algoritmech lze nalézt například v [13].

Na konec této kapitoly poznamenejme, že dalším krokem po určení odhadů regresních koeficientů je testování, které z nich jsou statisticky významné. Toto v případě metody dílčích nejmenších čtverců provádíme křížovou validací pomocí metody jackknife. Její použití na konkrétním příkladu si ukážeme v následující kapitole.

4 Praktický příklad

V následující kapitole budou předchozí teoretické poznatky aplikovány na numerický příklad. Reálná data pochází z prostředí Fakultní nemocnice v Olomouci a z důvodu citlivosti těchto dat je třeba dodržovat určitou anonymitu. Proto výchozí data nebudou k této práci přiložena.

Při zpracování praktické části této práce jsem vycházela z nápovědy statistického softwaru *R*, kterou lze nalézt v [10].

4.1 Shluková analýza

Máme k dispozici 67 pozorování, na kterých byly sledovány hladiny určitých látek - 14 typů aminokyselin. Mezi těmito pozorováními jsou přitom pozorování získaná z kontrol zdravých osob a dále pozorování získaná od pacientů, kteří mají určitou poruchu metabolismu spojenou s aminokyselinami. Přesné rozdělení těchto vzorků je blíže popsáno v následující tabulce.

kontr./nemoc	pozorování
kontrola	1, . . . , 50
nemoc 1	51, 52, 53, 54, 58
nemoc 2	55, 59, 60, 61, 62, 63
nemoc 3	56, 57
nemoc 4	64
nemoc 5	65, 66
nemoc 6	67

Pomocí následujících metod se pokusíme najít shluky mezi pozorováními a rozlišit vzorky získané z kontrol a vzorky pacientů s různými chorobami neboli poruchami metabolismu.

Všechny metody shlukové analýzy byly vypočteny pomocí statistického softwaru *R* a jeho knihovny *cluster*. Data z excelového souboru byla načtena pomocí příkazu $x = read.csv2(\text{"}metabol.csv\text{"})$. Z důvodu jejich lepší interpretace jsme na tato data aplikovali centrovanou logratio (clr) transformaci - tentokrát s užitím knihovny *robCompositions* pomocí příkazu $X = cenLR(x)\$x.clr$. Obecně

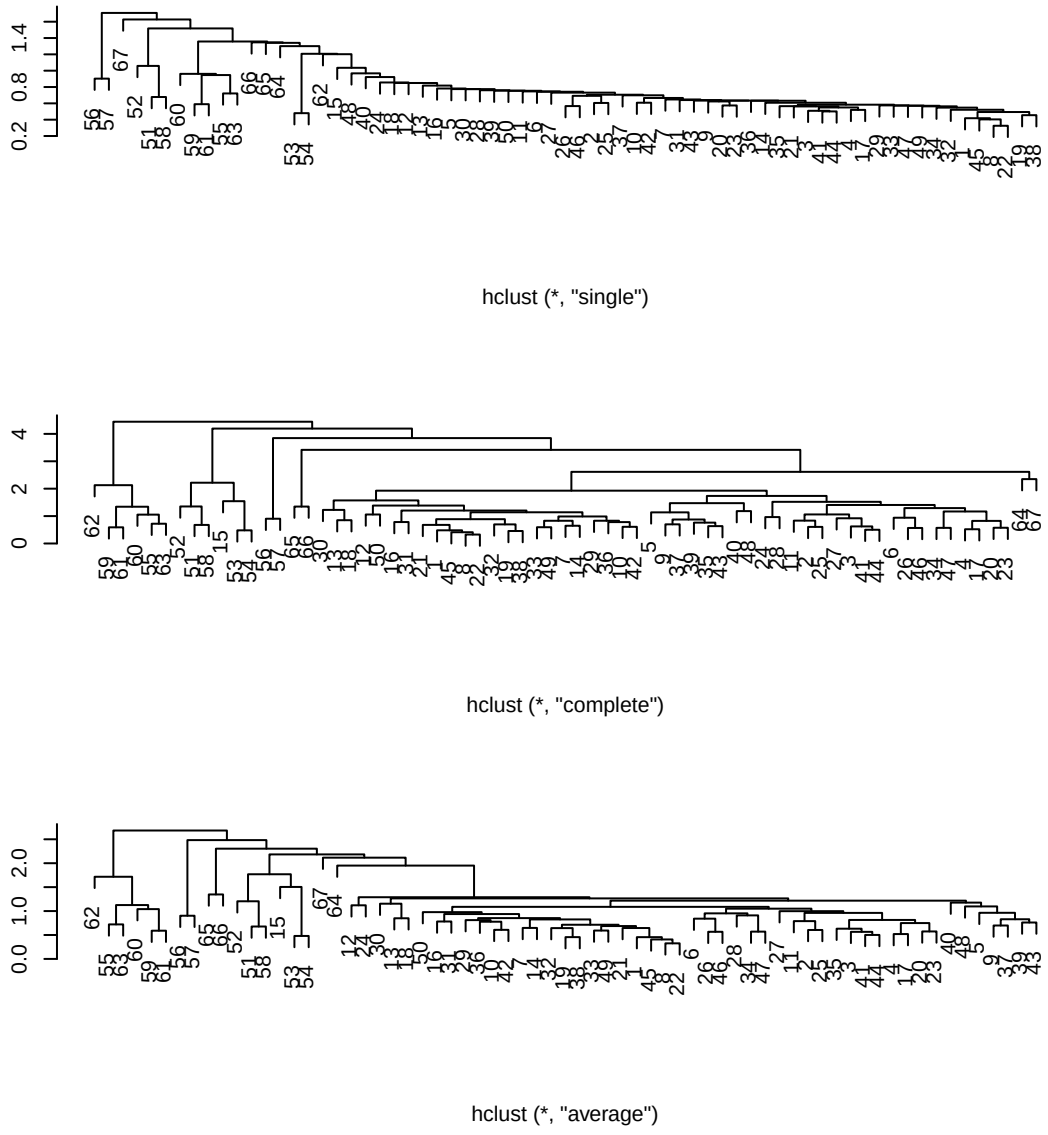
clr transformaci pozorování $\mathbf{x}$ o p proměnných definujeme pomocí následujícího vztahu

$$clr(\mathbf{x}) = \left(\ln \frac{x_1}{\sqrt[p]{\prod_{i=1}^{p} x_i}}, \ldots, \ln \frac{x_p}{\sqrt[p]{\prod_{i=1}^{p} x_i}} \right)^T .$$

Více informací o clr transformaci lze nalézt v [4].

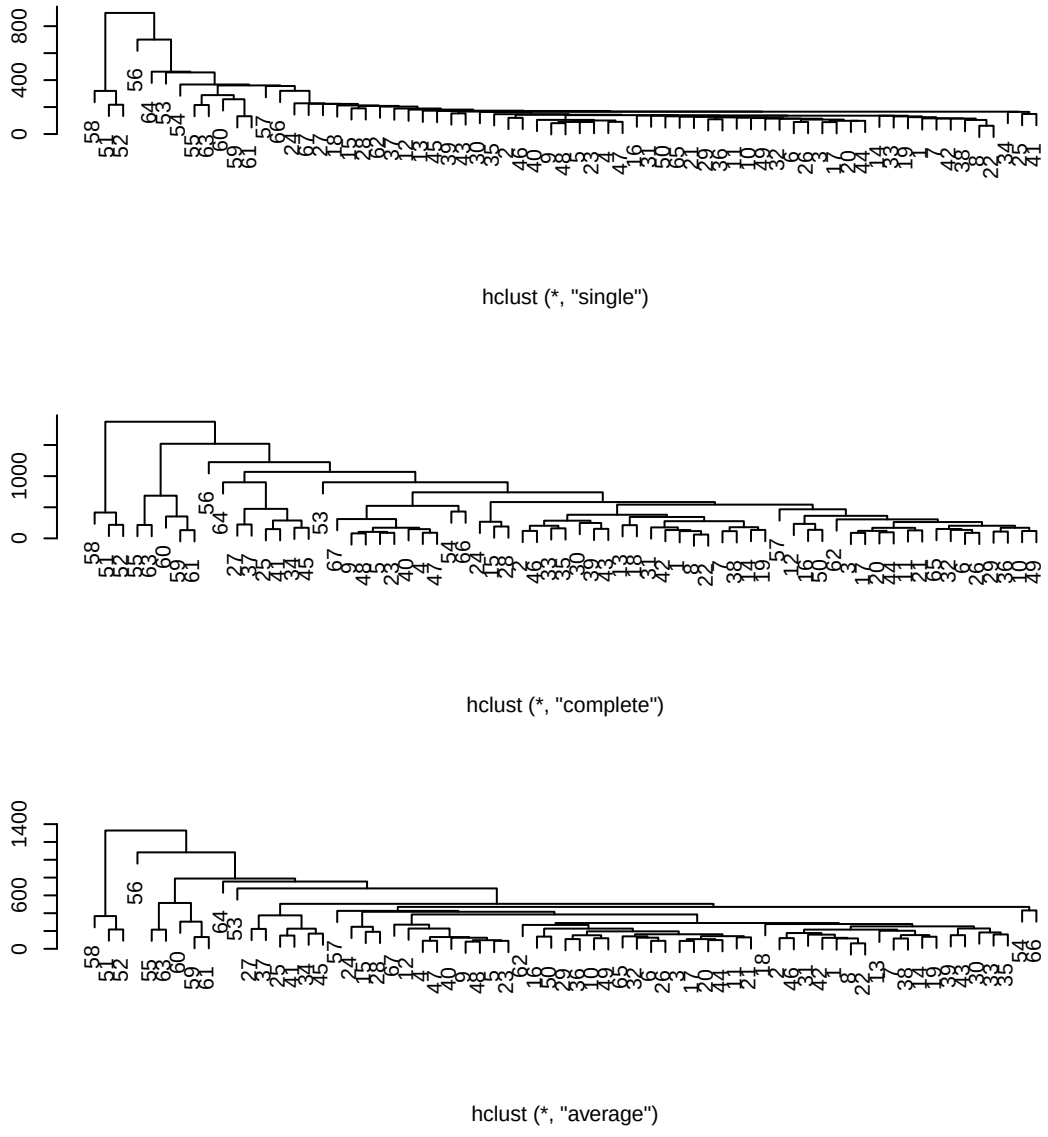
Aglomerativní shlukování

Společným počátečním krokem algoritmů hierarchického shlukování je výpočet matice vzdáleností, kterou jsme získali pomocí příkazu $d = dist(X)$. V tomto případě jsme obdrželi matici vzdáleností o rozměru (67×67). Dále byl použit příkaz $hclust = hclust(d, method = "complete")$, kde jsme parametr $method$ v závislosti na použitém vztahu pro výpočet vzdáleností mezi pozorováními (shluky) změnili na "$average$" nebo "$single$". Následně jsme pro vykreslení jejich příslušných dendrogramů užili funkci $plot(hclust)$.



Obrázek 7: dendrogramy transformovaných dat

Pro srovnání uvedeme i dendrogramy původních netransformovaných dat. Přitom si lze všimnout (viz. podrobná analýza výsledků dále), že struktura dendrogramů původních dat je méně přehledná a vytvořené shluky neodpovídají skutečnému rozdělení pozorování do skupin tak dobře, jako dendrogramy transformovaných dat. Proto budeme v této kapitole nadále vycházet z dendrogramů clr transformovaných dat.



Obrázek 8: dendrogramy původních dat

Určení shluků pozorování z výsledných dendrogramů je velice subjektivní. V závislosti na výšce řezu větví v grafu dostaneme vždy trochu jiný počet shluků. Obecně lze však říci, že nejjednodušší strukturu rozdělení pozorování nám dává v obrázku 7 první dendrogram, který vychází z metody "single", tedy single linkage. Tato jednoduchá struktura grafu však skutečné rozdělení dat vystihuje hůře než zbylé dva dendrogramy.

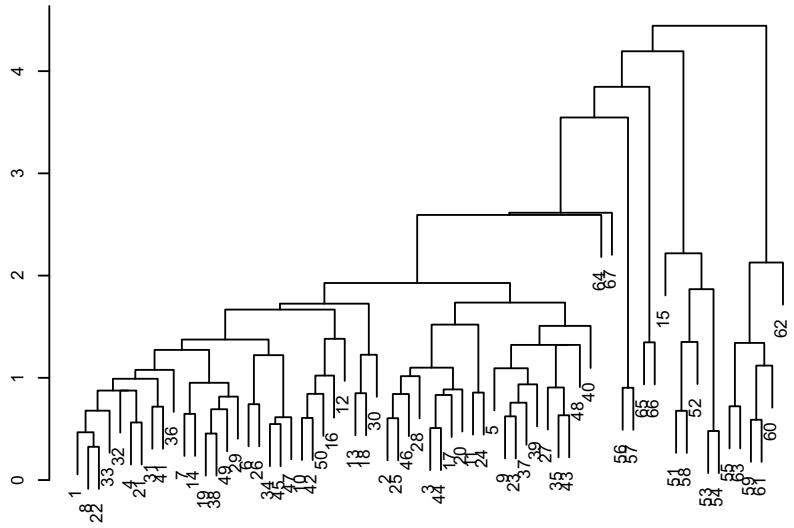
Ve všech třech dendrogramech si můžeme všimnout odlehlého pozorování 62. Také pozorování 56, 57 a 65, 66 by se mohla zdát jako odlehlá, ale jedná se o dva malé shluky dvou chorob po dvou pozorováních. Obdobně je tomu u pozorování 64 a 67, která odpovídají dvěma chorobám, které mají ve vstupním datovém souboru jen po jednom vzorku.

V dendrogramech vycházejících ze vzorců "complete" a "average" je velice čitelně rozpoznatelný shluk o pozorováních 15, 51, 52, 53, 54 a 58, která odpovídají jedné chorobě (pozorování 15 je však kontrolní vzorek, u kterého jsme tak identifikovali příznaky této choroby). Obdobně je tomu u další choroby, pro kterou lze v těchto dendrogamech najít shluk o pozorováních 55, 59, 60, 61, 62 a 63. V těchto dendrogramech lze vidět dále dva větší shluky, přičemž oba odpovídají vzorkům z kontrol. Jeden z těchto dvou shluků je tvořen pozorováními 2, 3, 4, 5, 6, 9, 11, 17, 20, 23, 24, 25, 26, 27, 28, 34, 35, 37, 39, 40, 41, 43, 44, 46, 47 a 48. Druhý shluk je tvořen pozorováními 1, 7, 8, 10, 12, 13, 14, 16, 18, 19, 21, 22, 29, 30, 31, 32, 33, 36, 38, 42, 45, 49 a 50.

Výsledky lze tedy interpretovat tak, že dendrogramy nám rozlišily pozorování získaná z kontrol od pozorování odpovídající chorobám a zároveň dokázaly rozlišit jednotlivé choroby.

Divizivní shlukování

Shluky ve skupině zadaných pozorování jsme v metodě divizivního shlukování našli pomocí příkazu $diana = diana(d)$, kde parametr d je matice vzdáleností pozorování, která byla použita i v předchozí metodě. Interpretovatelné výsledky zde lze získat opět z dendrogramu, který vytvoříme příkazem $plot(diana)$.



Obrázek 9: dendrogram divizivního shlukování

Rozdělení jednotlivých pozorování do shluků je zde téměř stejné jako v předchozím aglomerativním shlukování. Dendrogram tedy i tomto případě dokázal rozlišit pozorování získaná z kontrol a pozorování odpovídající jednotlivým chorobám.

Metoda k průměrů

Výsledky získané metodou k průměrů jsou pro stejná data pokaždé trochu jiné v důsledku náhodného počátečního rozdělení pozorování do shluků. Využili jsme zde příkazu $kmeans(X, 2)$, kde parametr 2 značí požadovaný počet shluků. Našim cílem totiž je, aby byl algoritmus schopen rozlišit vzorky získané z kontrol a vzorky pacientů, kteří mají některou chorobu.

Tímto algoritmem jsme získali následující rozdělení pozorování. Jeden ze shluků je tvořen 12-ti pozorováními 15, 51, 52, 53, 54, 55, 58, 59, 60, 61, 62 a 63. Zbylá pozorování pak tvoří druhý shluk. V porovnání s původními daty lze usoudit, že data jsou víceméně rozdělena na skupinu vzorků získaných z kontrol a skupinu vzorků nemocných pacientů.

Pomocí uvedeného příkazu jsme rovněž získali průměry obou shluků dle všech proměnných, které zde pro přehlednost uvedeme v následující tabulce.

látka	$\overline{\mathbf{x}}_1$	$\overline{\mathbf{x}}_2$
Arg	-0.15	0.01
Gln	1.40	1.53
Gly	0.95	0.95
His	-0.57	-0.32
Met	-1.70	-1.42
Orn	-0.88	-0.69
Phe	0.45	-0.61
Pro	-0.11	0.16
Ser	0.12	0.27
Thr	-0.06	-0.01
Trp	-0.62	-0.42
Tyr	-0.84	-0.39
Val	0.93	0.42
xLeu	1.08	0.52

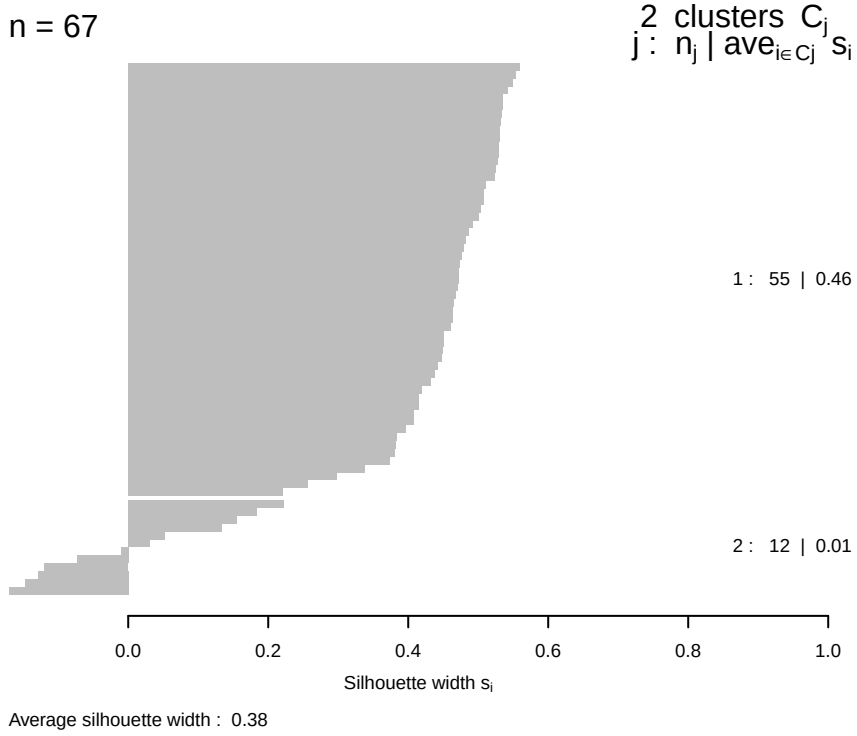
Pomocí funkce $kmeans(X, 2)\$totss$ jsme navíc dostali informaci o celkovém součtu čtverců jednotlivých pozorování od příslušných průměrů, tedy $ESS = 115.04$.

Fuzzy shlukování

V případě fuzzy shlukování je opět naší volbou, do kolika shluků chceme pozorování rozdělit. Stejně jako v předchozí metodě se pokusíme data rozdělit do dvou shluků, využijeme tedy příkazu $fanny = fanny(X, 2, memb.exp = 1.1)$.

Výsledek jsme získali po 17 iteracích. Rozdělení pozorování do dvou shluků zde vyšlo úplně identicky jako v metodě k průměrů. Našli jsme dva shluky, kde jeden z nich je víceméně tvořen vzorky pacientů s chorobami a druhý shluk tvoří zbylá pozorování.

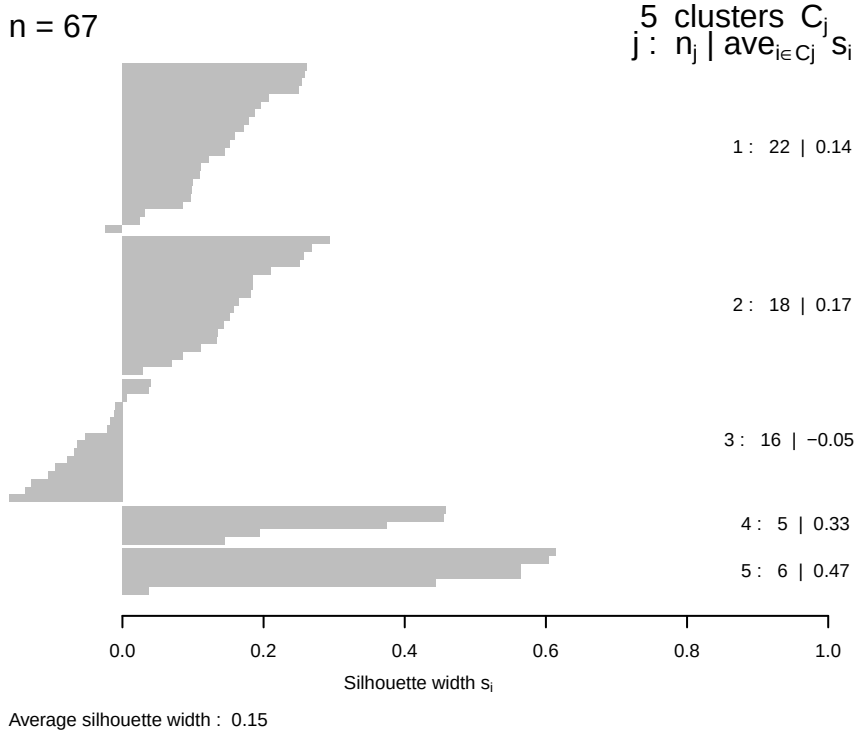
Příslušný siluetový graf zde získáme zavoláním funcke $plot(fanny)$, který nám příslušné výsledky interpretuje graficky.



Obrázek 10: siluetový graf pro dva shluky

Průměrná hodnota obrysu je 0.38, z čehož můžeme usoudit, že rozdělení pozorování do těchto shluků není úplně jednoznačné. Specielně hodnota obrysu shluku výše vyjmenovaných 12-ti pozorování je 0.01, což nám říká, že taková pozorování leží mezi dvěma shluky. To je zřejmě způsobeno tím, že je tento shluk tvořen pozorováními, která odpovídají dvěma menším shlukům, které v předchozích metodách odpovídaly dvěma typům chorob. Hodnota shluku zbylých 55 pozorování je 0.46.

Pokud bychom chtěli ověřit podrobnější rozdělení pozorování do menších shluků, jak již bylo ukázáno v předchozích (hierarchických) metodách, upravili bychom v příkazu požadovaný počet komponent, tedy $fanny(X, 5, memb.exp = 1.1)$. Interpretaci výsledného rozdělení pozorování do pěti shluků popíšeme pomocí následujícího siluetového grafu.



Obrázek 11: siluetový graf pro pět shluků

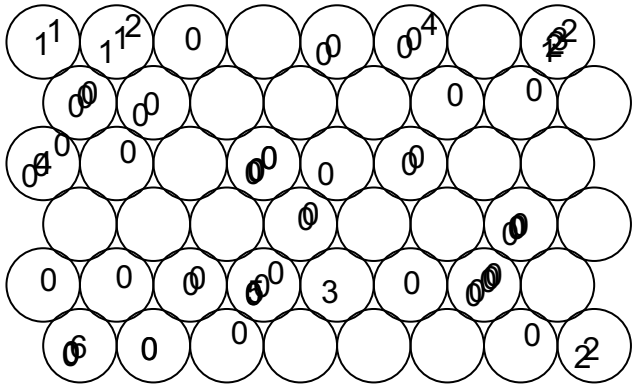
Průměrná hodnota obrysu je tentokrát 0.15, což nás upozorňuje na nezcela dobré rozložení pozorování do shluků. Ale podíváme-li se na tyto shluky jednotlivě, lze říci, že takto nízká průměrná hodnota obrysu je způsobena nespolehlivým rozdělením prvních tří shluků, avšak čtvrtý, resp. pátý shluk, jejichž hodnoty obrysu jsou 0.33, resp. 0.47, jsou mnohem zřetelnější. Tyto dva shluky přitom odpovídají zcela přesně pozorováním odpovídajícím dvěma chorobám.

Samoorganizující mapy - hromadný algoritmus

Hromadný algoritmus metody samoorganizujících map lze na daná data aplikovat rovněž ve statistickém softwaru *R*, tentokrát však v knihovně *class*. Pomocí následujícího souboru příkazů vykreslíme uzlový graf a pokusíme se v něm najít sedm shluků pozorování, které odpovídají vzorkům z kontrol a dále vzorkům pacientů s šesti různými chorobami.

V prvním kroku algoritmu je třeba zavést označení jednotlivých dat, kdy pomocí čísel 0 až 6 rozlišíme vzorky získané z kontrol a vzorky pacientů se šesti různými nemocemi, tedy $oznac = factor(c(rep("0", 50), rep("1", 5), rep("2", 6), rep("3", 2), rep("4", 2), "5", "6"))$. Dále pomocí příkazu $sit = somgrid(xdim = 8, ydim = 6, topo = "hexagonal")$ navrhneme šestiúhelníkovou síť o 48 uzlech. Počáteční množinu reprezentantů necháme sestavit náhodně a poté lze na data aplikovat samotný hromadný algoritmus pomocí $batch = batchSOM(X, sit, c(1))$, kde parametr c určuje práh vzdálenosti sousedních shluků, které identifikujeme pomocí příkazu $soused = as.numeric(knn1(batch\$code, X, 0 : 47))$.

Nyní lze získané výsledky interpretovat graficky pomocí příkazu $plot(batch\$grid, type = "n")$, přičemž specielně pro hromadný algoritmus jsou uzly znázorněny jako kruhy, tedy $symbols(batch\$grid\$pts[, 1], batch\$grid\$pts[, 2], circles = rep(0.5, 48), inches = FALSE, add = TRUE)$. Do takto vytvořených uzlů je třeba umístit příslušná pozorování ve značení, jaké jsme zavedli na začátku algoritmu, přičemž tato pozorování jsou v příslušném uzlu umístěna náhodně, proto využijeme tohoto posledního příkazu algoritmu $text(batch\$grid\$pts[soused,] + rnorm(67, 0, 0.1), as.character(oznac))$.



Obrázek 12: uzlový graf

Z výsledného grafu lze zřejmě usoudit, že pozorování odpovídající kontrolním vzorkům, která jsou zde značena číslem 0, tvoří jeden shluk. Také pozorování značená číslem 1, která zastupují jednu z chorob, zřejmě tvoří shluk. O dalším

rozdělení dat do shluků lze špatně usoudit. V tomto případě není uzlový graf v porovnání s předchozími metodami nejvhodnější metodou pro interpretaci výsledků shlukování.

4.2 Metoda dílčích nejmenších čtverců

Protože jsou algoritmy metody dílčích nejmenších čtverců vhodné také pro situace, kdy počet proměnných převyšuje počet pozorování, využijeme datový soubor s takovou vlastností. Tentokrát tedy budeme vycházet z 11 pozorování o 23 proměnných, kde prvních 14 proměnných jsou jednotlivé typy aminokyselin (Arg, Gln, Gly, His, Met, Orn, Phe, Pro, Ser, Thr, Trp, Tyr, Val, xLeu) a dalších 9 proměnných jsou jednotlivé druhy acylkarnitinů (C0, C10, C10.1, C10.2, C12, C12.DC, C12.1, C14, C14.1). Datový soubor je tvořen 5-ti pozorováními, které odpovídají kontrolním vzorkům, zbylých 6 pozorování jsou vzorky těch pacientů, kteří trpí stejnou chorobou. Právě toto rozdělení vzorků budeme nadále považovat za závisle proměnnou, kde kontrolní vzorky označíme číslem -1 a vzorky odpovídající chorobě označíme číslem 1. Pomocí následujících algoritmů se pokusíme zjistit, která látka (který typ aminokyseliny či acylkarnitinu) má největší vliv na danou nemoc.

Algoritmus NIPALS

Jak již bylo uvedeno v teoretické části práce, algoritmus NIPALS řešíme v softwaru R v knihovně *chemometrics*. Výpočet tohoto algoritmu zadáme pomocí příkazu $pls1\_nipals(X, y, a = 2)$, přičemž požadujeme, aby všechny původní proměnné byly nahrazeny dvěma novými proměnnými.

Našim cílem je získat vektor regresních koeficientů, který je pro přehlednost uveden v následující tabulce.

látka	odhad	látka	odhad
Arg	-0.07332	Val	0.48864
Gln	0.24681	xLeu	-0.04287
Gly	0.22233	C0	0.11423
His	-0.03819	C10	-0.00064
Met	0.07354	C10.1	0.00154
Orn	0.02354	C10.2	0.00030
Phe	**2.66587**	C12	0.00028
Pro	0.86567	C12.DC	0.00041
Ser	-0.51338	C12.1	0.00446
Thr	0.10479	C14	-0.00004
Trp	-0.00391	C14.1	-0.00002
Tyr	-0.08810		

Z těchto výsledných odhadů regresních parametrů lze usoudit, že na danou nemoc mají obecně větší vliv hodnoty aminokyselin. Vliv hodnot acylkarnitinů je v porovnání s vlivem aminokyselin téměř zanedbatelný. Výrazně největší vliv na tuto chorobu mají však hodnoty aminokyseliny značené Phe.

Algoritmus SIMPLS

V tomto algoritmu budeme vycházet ze stejného zadání jako v algoritmu NIPALS a pokusíme se potvrdit jeho závěr. Navíc zde provedeme testování významnosti odhadů parametrů prostřednictvím křížové validace.

Jedná se o proces, kdy náhodně rozdělíme vstupní data na V zhruba stejně velkých disjunktních množin. Jednu z těchto množin odebereme a označíme T. Ze zbylých $V - T$ množin (označ. L) vytvoříme model, jehož výsledky otestujeme na množině T. Tento proces opakujeme V-krát a dosažené výsledky využijeme k vyhodnocení příslušného modelu. V případě použití křížové validace typu "Leave One Out" je množina T tvořena vždy jedním pozorováním.

Více informací o křížové validaci lze nalézt například v [5].

Ve statistickém softwaru R v knihovně *pls* aplikujeme na načtená data algoritmus SIMPLS pomocí funkce $simpls = mvr(y \sim X, method = "simpls", validation = "LOO", jackknife = TRUE)$.

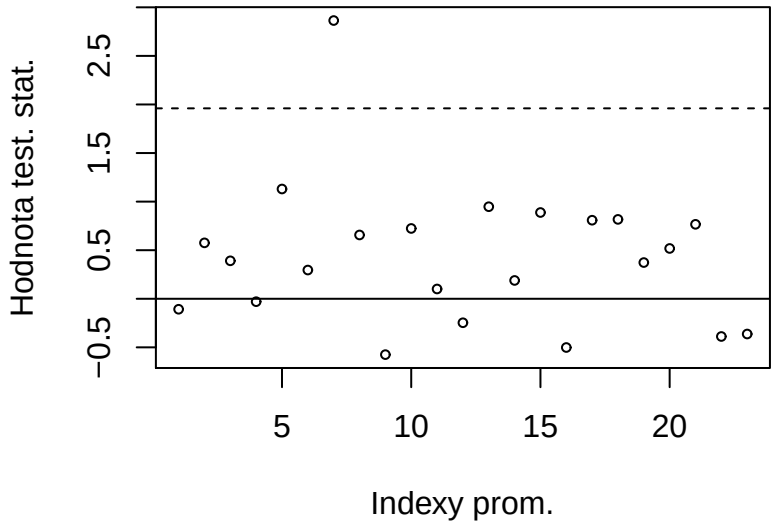
Pro otestování statistické významnosti odhadů regresních parametrů využijeme dále funkci $t\_test = jack.test(simpls, ncomp = 2)$, kde požadujeme, aby původní proměnné byly nahrazeny dvěma novými proměnnými. V následující tabulce jsou uvedeny odhady všech regresních parametrů.

látka	odhad	látka	odhad
Arg	-0.0001006	Val	0.0015780
Gln	0.0012405	xLeu	0.0002367
Gly	0.0006365	C0	0.0003533
His	-0.0000177	C10	-0.0000072
Met	0.0002411	C10.1	0.0000060
Orn	0.0001490	C10.2	0.0000011
Phe	**0.0021149**	C12	0.0000009
Pro	**0.0028245**	C12.DC	0.0000014
Ser	-0.0013424	C12.1	0.0000172
Thr	0.0004676	C14	-0.0000002
Trp	0.0000473	C14.1	-0.0000003
Tyr	-0.0001356		

Na základě těchto výsledků lze opět obecně usoudit, že hodnoty aminokyselin mají zřejmě větší vliv na danou nemoc než hodnoty acylkarnitinů. Navíc nejvýznamnější látkou je zde aminokyselina značená Pro a dále také aminokyselina Phe.

Testování významnosti odhadů regresních koeficientů zde vysvětlíme na grafu. Statistická významnost jednotlivých odhadů byla testována pomocí křížové validace, kdy byly empiricky určeny směrodatné odchylky jednotlivých odhadů parametrů a následně pro každý regresní parametr vypočtena hodnota testovací statistiky, tvořené podílem odhadu a příslušné směrodatné odchylky. Absolutní hodnoty testovací statistiky byly porovnány s $(\frac{\alpha}{2})$-kvantilem normálního normovaného rozdělení při $\alpha = 0.05$. Pomocí následující funkce jsme v softwaru R vykreslili graf, kde na ose x jsou jednotlivé indexy proměnných a na ose y jsou hodnoty testovací statistiky, tedy $plot(1 : ncol(X), drop(t\_test\$tvalues), cex = 0.6, xlab = "Indexy\ prom.", ylab = "Hodnota\ test.\ stat.")$. Přitom jsme pomocí funkce $abline(h = 1.96, lty = "dashed")$ zvýraznili jednu krajní hodnotu kritic-

kého oboru $u_{0.975} = 1.96$. Druhou krajní hodnotu $u_{0.025} = -1.96$ nebylo v tomto případě potřeba aplikovat.



Obrázek 13: grafické znázornění statistické významnosti parametrů

Z grafu lze určit, že jediný statisticky významný odhad parametrů je odhad proměnné Phe. Ačkoliv se tedy z vektoru regresních parametrů zdálo, že nejvýznamnější proměnnou bude aminokyselina Pro a druhá nejvýznamnější bude aminokyselina Phe, lze na základě výsledku testování významnosti těchto parametrů potvrdit výsledek předchozího algoritmu, a sice, že na danou nemoc má podle obou algoritmů vliv jediná proměnná Phe.

Závěr

Při průzkumu velkého množství metod mnohorozměrné statistické analýzy, ze kterých jsem vybírala ty, jimiž se zabývám v této práci, jsem byla ohromena jejich názorností a využitelností v praxi. Takto získaný přehled je pro mne velkým přínosem ve studiu mnohorozměrné statistiky.

Za velkou zkušenost rovněž považuji práci se statistickým softwarem *R*, ve kterém jsem zpracovávala reálná data na základě získaných teoretických poznatků. Díky tomu, že jsem se v práci nezabývala jedinou oblastí mnohorozměrné statistiky, měla jsem možnost pracovat s několika různými knihovnami softwaru *R* a věřím, že mnoho jiných zájemců o tuto problematiku ocení uvedené a popsané funkce, které byly v algoritmech použity. Zajímavá pro mne byla i často opomíjená situace, kdy jsem statisticky zpracovala data, kde počet proměnných převyšoval počet pozorování. Přitom tato situace je v praxi, a to zejména v oblastech medicíny a chemie, velice častou. Jediným problémem, se kterým jsem se v diplomové práci potýkala, byl nedostatek vhodné základní literatury při studiu metody samoorganizujících map, který byl zřejmě dán určitou specifičností užití tohoto algoritmu.

Podle mého názoru byl tedy cíl práce, vytvořit teoretický přehled vybraných metod a teorii aplikovat na reálná data pomocí softwaru *R*, splněn.

Literatura

[1] Anděl, J., *Matematická statistika*, 1. vydání. Praha: SNTL - Nakladatelství technické literatury, 1978.

[2] Filzmoser, P., Hron, K., Reimann, C., *Principal component analysis for compositional data with outliers*, Environmetrics **20**, 621 - 632 (2009).

[3] Hebák, P. a kol., *Vícerozměrné statistické metody (3)*, 1. vydání. Praha: Informatorium, 2005.

[4] Hron, K., *Elementy statistické analýzy kompozičních dat*, Informační bulletin České statistické společnosti **21**, 41 - 48 (2010).

[5] Izenman, A., J., *Modern multivariate statistical techniques: regression, classification, and manifold learning*, 1. vydání. New York: Springer, 2008.

[6] Jajuga, K., Sokołowski, A., Bock H. - H., *Classification, clustering, and data analysis: recent advances and applications*. Berlin: Springer, 2002.

[7] Kalivodová, A., *Diplomová práce: Kompoziční biplot*, Olomouc: UPOL, 2012.

[8] *Kohonenova samoorganizující mapa a její aplikace v marketingu* [online], dostupné z: http://www.lemonway.com/research/nan-clanek-final.pdf, [citováno 7. 11. 2012].

[9] Rosipal, R., Krämer, N., *Overview and recent advances in partial least squares*, C. Saunders a kol: SLSFS, 34–51 (2006).

[10] *The R Project for Statistical Computing* [online], dostupné z: http://www.r-project.org/, [citováno 5. 3. 2013].

[11] Trygg, J. a kol., *Chemometrics in metabonomics*, Journal of Proteome Research **6**, 469-479 (2007).

[12] Trygg, J., Wold, S., *Orthogonal projections to latent structures (O-PLS)*, Journal of Chemometrics **16**, 119-128 (2002).

[13] Varmuza, K., Filzmoser, P., *Introduction to multivariate statistical analysis in chemometrics*, 1. vydání. Boca Raton: CRC Press, 2009.