Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Vybrané metody analýzy vícerozměrných dat v MATLABu

Diplomová práce

Autor: Bc. Pavel Vojtas
Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. RNDr., Pavel Pražák, PhD.

Hradec Králové 04.2017

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a uvedl jsem všechny použité prameny a literaturu.

V Hradci Králové dne: 19. dubna 2017 Bc. Pavel Vojtas

Poděkování

Děkuji vedoucímu diplomové práce doc. RNDr., Pavlovi Pražákovi, PhD. za metodické vedení práce a cenné rady.

Anotace

Tato diplomová práce se zabývá vybranými metodami vícerozměrné analýzy dat. Popisuje základní principy vybraných metod a reprezentaci jimiž dosažených výsledků pro různé oblasti využití. Všechny vybrané metody jsou již existující implementace v prostředí MATLAB, ke kterým je vystavěno vlastní grafické rozhraní. To seskupuje funkce do logických celků a usnadňuje práci s nimi. V závěru práce jsou výsledky porovnány se specializovanými statistickými softwary.

Annotation

Title: Selected methods for multivariate data analysis in MATLAB

This diploma thesis follows up selected methods for multivariate data analysis. It describes basic principles of these methods and interpretation of their results for different practical areas. All of the methods have already been implemented in the MATLAB environment. There is built a graphical user interface to these implementations. This GUI gathers functions to logical segments for their easier using. In conclusion, the results, which has come from these methods, are compared with outputs of specialized softwares.

Obsah

Seznam obrázků

Seznam tabulek

1. Úvod

Oblast analýzy vícerozměrných dat je velice široká a hojně využívaná. V praxi nelze totiž, v drtivé většině, popsat reálný objekt jednou nebo dvěma vlastnostmi. Tím nám vzniká vícerozměrný objekt.

Problém potom nastává v grafické reprezentaci, v porovnání dvou a více objektů, atd. Naše myšlení je schopnou lehce zpracovat dvou až tří rozměrnou reprezentaci objektu. Většina objektů reálného světa má ale rozměry větší. S těmito problémy se však můžeme vypořádat pomocí speciálních metod, které mám dané objekty pomohou zjednodušit do dvou až třírozměrné reprezentace.

Vedle zobrazení dat je také důležitá rozlišovací schopnost. Například mějme skupiny určitých chorob, které jsou definovány více vlastnostmi. Bylo by tedy vhodné zařadit nového jedince do skupiny pro patřičnou chorobu na základě jeho příznaků, tedy na základě určitého modelu.

Další oblastí může být rozpoznávání shluků. To se dá využít například v lékařství, kde na základě pořízených obrazových dat potřebujeme zjistit oblasti se zvýšenou koncentrací buněk určité vlastnosti.

2. Cíl práce a metodika zpracování

Cílem práce bude problematika analýzy vícerozměrných dat. Zaměří se na vybrané analytické metody a jejich využití pro různé typy úloh, z různých oblastí. Dalším cílem práce bude vybrané metody seskupit do logických celků pomocí grafického uživatelského rozhraní a tím použití zefektivnit.

Rešerše literatury

Metody pro analýzu dat, lze rozdělit podle různých kritérií. Rozdělení podle rozměru dat na metody jednorozměrné a vícerozměrné. Dalším měřítkem může být účel, např. metody diskriminační a klasifikační (Diskriminační analýza, Logistická regrese, Shluková analýza, Multidimenzionální škálování, ...), regresní, průzkumové (odhady parametrů, testy normality, ...), metody pro odhalení struktury (Metoda hlavních komponent, Faktorová analýza, ...). Těmto a dalším metodám se věnují publikace [14], [5], [16], [6], [18].

Výše popsané metody jsou také základem sofistikovanějších algoritmů pro využití ve speciálních oblastech, kde ostatní metody selhávají. Jedná se například o Fuzzy shlukování, umělé neuronové sítě pro klasifikaci, upravené metody pro rozpoznávání obrazu a další. Více v publikacích [18], [8], [9].

Tyto metody mají důležitou roli v oblasti data miningu (např. určování strategií společnosti, atd.), text miningu (např. fulltextové vyhledávání, atd.), pokročilých metod strojového učení či různé predikce pro průzkumy trhu, a mnohé další. Těmto oblastem se více věnují publikace [11], [22].

Metodika zpracování

První část práce se bude zabývat problematikou vícerozměrné analýzy dat. Bude popisovat různé oblasti využití. Na tomto základě se provede výběr třech metod a jejich popis. Druhá část práce se bude věnovat samotnému zpracování dat. Od přípravy po analýzu a interpretaci výsledků. Součástí analýzy bude implementace vybraných metod. Vše bude probíhat v prostředí MATLAB. Na závěr bude celý proces zhodnocen a porovnám s jinými řešeními.

3. Problematika vícerozměrných dat a oblasti využití

Tato část se zabývá problematikou vícerozměrných dat. Popíší se zde základní oblasti zkoumání a přehled metod, který se v této oblasti využívá.

Oblast explorační

Ve většině případů reálných vícerozměrných dat nejsou splněny předpoklady statistických testů. Buď data nepochází z normálního vícerozměrného rozdělené, nebo jsou hodnoty dat spolu provázané, mohou se v nich vyskytovat odlehlá pozorování aj.

Vzhledem k tomu je vhodné mít nějaký přehled o tom jak vypadá náš datový soubor a na tom základě použít různé korekce, abychom docílili požadovaných vlastností. Více v publikacích [4], [14].

Používané techniky zobrazení lze rozdělit na rozptylové diagramy a symbolové grafy, viz [14]. Rozptylové grafy je vhodné zobrazovat ve dvou nebo tří dimenzionálním grafu. Objekty mají však často více než 3 dimenze. Zde je tedy vhodné strukturu zjednodušit, například použitím metody Hlavních komponent. Touto metodou se budeme zabývat v kapitole 4.

Dále sem patří ověřování vícerozměrné normality. To lze provést grafickým nebo početním způsobem. Mezi grafický způsob patří například Q-Q graf. Mezi početní způsoby například Roystonův H-test, Smallův Q-Test, test založený na vícerozměrné šikmosti a špičatosti. Více o těchto testech v publikaci [14]. V této práci se budeme zabývat Q-Q grafem a testem vícerozměrné šikmosti a špičatosti, viz kapitola 5.

Oblast diskriminační a klasifikační

S oblastí vícerozměrných dat souvisí i diskriminační a klasifikační úlohy. Máme vícerozměrný datový soubor, kde řádky reprezentují objekty nějakého pozorování a sloupce jeho vlastnosti. Jedná se tedy o vícerozměrné objekty. Cílem těchto metod je objekty roztřídit do předem známých nebo neznámých skupin. Tyto metody se označují také jako metody strojového učení. Více v publikaci [14].

Pokud třídíme do předem známých skupin, jedná se o úlohu diskriminace. Označuje se též jako učení s učitelem. Princip spočívá v tom, že datový soubor se rozdělí na trénovací a testovací. Na trénovacím souboru, se vytvoří model, který umožní zařazovat prvky do námi zvolených skupin. Na testovacím souboru se model následně ověřuje. Patří sem metoda Diskriminační analýzy, Logistické regrese a další.

Diskriminační analýza spočívá v hodnocení rozdílů mezi 2 a více skupinami, které jsou charakterizovány více proměnnými. Vysvětlovaná proměnná je spojitá. Tato metoda má více variant. Pokud máme data z normálního rozdělení se shodnými kovariančními maticemi avšak rozdílnými středními hodnotami, je vhodné použít lineární diskriminační analýzu. Pokud máme data z normálního rozdělení, ale liší se střední hodnoty i kovarianční matice, je vhodné použít kvadratickou diskriminační analýzu. Pokud je porušena normalita, použije se nelineární diskriminační analýza. Více v publikacích [14], [16].

Logistická regrese na rozdíl od diskriminační analýzy používá kategoriální vysvětlovanou proměnnou. Je vhodné metodu použít při porušení normality dat. Predikuje pravděpodobnost události a nabývá hodnot 0, 1. Používá tzv. sigmoidní funkci. Více v publikaci [14].

Pokud třídíme do předem neznámých skupin, jedná se o úlohu klasifikace. Označuje se též jako učení bez učitele. Patří sem například metoda Shlukové analýzy. Ta spočívá ve zjištění podobností objektů a zařazení do předem neznámých skupin. Používá se pro popis systematiky, zjednodušení dat, identifikaci vztahů, aj. Více v publikaci [18], [14], [5]. Této metodě se budeme dále věnovat v kapitole 4.

Text mining

Oblast text miningu je důležitá pro zpracování otevřeného textu. Hodí se zejména pro porozumění textovým dokumentům, vyhledávání v dokumentech a pro klasifikaci doku-

mentů. V pokročilejších nástrojích lze také provádět sentimentální analýzu, například z diskuzních komentářů určité služby nebo určitého produktu. Dá se využít pro zpracování otevřených otázek z dotazníkového šetření. Dále lze sledovat veřejné mínění z různých diskuzních skupin, sociálních sítí atd. Určité metody lze využít i pro průzkum trhu, a další. Podrobnější informace jsou v publikaci [11].

V této práci se budeme zabývat pouze oblastí klasifikace dokumentů.

Mějme databázi o stovkách dokumentů. Nalezneme v dokumentech klíčová slova, na jejichž základě provedeme shlukování. Tím získáme sady dokumentů s přibližně podobným informačním obsahem. Výsledkem pro uživatele je tedy určitý stupeň filtrace, kde se identifikují sady obsahující pouze pár jednotek až desítek dokumentů.

Dá se zde využít nehierarchické shlukování, které kategorizuje dokumenty do předem určeného počtu shluků. Dá se využít i hierarchické shlukování, které poskytne celkový pohled a podobnost jednotlivých dokumentů.

Nejvýznamnější částí je zde předúprava dat. Součástí je **extrakce slov**. Metoda top-down rozdělí text pomocí oddělovacích znaků (mezera, tabulátor, interpunkce, ...) do řetězců (tzv. tokenů). Metoda bottom-up naopak prochází celý textový proud postupně znak po znaku a skládá je do řetězce. Jakmile narazí na oddělovací znak, začne vytvářet nový řetězec. Bez správného rozdělení na tyto řetězce mohou být dokumenty špatně klasifikované

Další částí předúpravy je **extrakce kořenů**. Slovo může být použito v mnoha tvarech (Apple, Apples, ...). Všechny tyto tvary mají ale stejný informační obsah. Proto je potřeba tyto slova redukovat a použít jen kořenové.

Po získání datového slovníku je potřeba dokumenty správně indexovat. Vytvořit tabulku, které pojednává o zastoupení jednotlivých slov v dokumentu. K tomu se využívají metody Frequency-Based nebo Inverse Document Frequency.

Frequency-Based metoda je založena na přiřazení jednotlivých řetězcůs k dokumentům. Váhy řetězců jsou počítány z četnosti výskytů slova a normalizovány počtem všech slov v dokumentů. Tato metoda je nedostatečná pro vyhledávání v dokumentech. **Inverse Document Frequency** je rozšíření eliminující nedostatky předešlé metody.

4. Vybrané metody analýzy vícerozměrných dat

V této kapitole se teoreticky popíší vybrané metody vícerozměrné analýzy dat. Bude se jednat o metodu hlavních komponent, faktorovou analýzu a shlukovou analýzu. Nejprve je ale třeba vysvětlit pojmy vlastních čísel a SVD rozkladu, které se budou dále používat.

4.1. Vlastní čísla a vlastní vektory

Podle publikace [17] (str. 135) jsou vlastní čísla a vektory definovány následovně.

Nechť matice $A \in C^{n \times n}$, potom platí

$$Ax = \lambda x, \tag{4.1}$$

kde $\lambda \in C$ nazýváme vlastním číslem matice A. Vektor $x \in C^n, x \neq 0$ je vlastní vektor příslušný k vlastnímu číslu. Podmínka nenulovosti zde zajišťuje konečnost řešení.

Vlastní čísla jsou charakterizována charakteristickým polynomem

$$\det(A - \lambda \cdot I) = 0. \tag{4.2}$$

Po rozepsání

$$(-1)^n \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0. \tag{4.3}$$

Jedná se o polynom n-tého stupně. Jeho kořeny jsou vlastní čísla matice A.

Množině všech vlastních čísel se říká spektrum.

4.2. SVD rozklad

Jedná se o rozklad na singulární čísla. Podle autora publikace [17] (str. 108) se jedná o nejdůležitější větou numerické lineární algebry. Pomocí SVD rozkladu, lze efektivně řešit mnoho problémů lineární algebry. Jedná se například o výpočet hodnosti matice, ortonormální báze, ale také o řešení soustav lineárních rovnic a další.

Na rozdíl od Gaussovy eliminační metody je lépe odolná vůči zaokrouhlovacím chybám. Avšak algoritmy pro výpočet jsou pouze iterativní. Více v publikaci [17].

Definice

Mějme matici $A_{m\times n}$. Potom existuje matice $\Sigma_{m\times n}$, která má na diagonále prvky $\sigma_{11} > \sigma_{22}$.... Dále existují ortogonální matice $U_{m\times m}$ a $V_{n\times n}$ takové, že platí

$$A = U\Sigma V^T. \tag{4.4}$$

Výpočet

Výpočet singulárního rozkladu je úzce spjat s rozkladem spektrálním a QR rozkladem.

Spektrální rozklad definujme s využitím spektrální věty pro symetrické matice ([17] str. 154)

$$X = V\Lambda V^T, \Lambda = V^T X V, \tag{4.5}$$

kde $X_{n\times n}$ je čtvercová matice, $V_{n\times n}$ je matice jejíž sloupce obsahují vlastní vektory a Λ je diagonální matice vlastních čísel.

Dále víme, že pro matici $A_{m\times n}$. existuje symetrická matice $A^T A$. Potom lze ukázat, že mezi SVD rozkladem a spektrálním rozkladem je následující vztah

$$A^T A = (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^T U^T U\Sigma V^T = V\Sigma^2 V^T. \tag{4.6}$$

Pro výpočet spektrálního rozkladu $A^T A$ lze použít Jacobiho iterační metodu ([17] str. 157). Po provedení výpočtu získáme matici $V_{n\times n}$. Následuje použití QR rozkladu ([17] str. 105)

$$AV = QR, \tag{4.7}$$

kde R je horní trojúhelníková matice a Q je ortogonální matice. Výpočet může být proveden Householderovým algoritmem, viz publikace [17].

Po úpravě získáme tedy

$$A = QRV^T, \tag{4.8}$$

kde Q je matice levých singulárních vektorů, diagonální prvky matice R jsou singulární čísla a V je matice pravých singulárních vektorů. Po úpravě značení $Q = U, diag(R) = \Sigma$ dostaneme vztah SVD rozkladu $A = U\Sigma V^T$.

4.3. Metoda hlavních komponent

Metoda hlavních komponent (PCA) je založena na ortogonální transformaci původních proměnných, které jsou silně korelované na skupinu nových proměnných, které nejsou korelované. Toho se dá využít například u snížení rozměru úlohy, odstranění multikolinearity (tj. lineární závislosti mezi znaky), aj.. Následující text vychází z publikací [14], [5], [13].

Princip metody

Mějme n-rozměrnou náhodnou veličinu X. Potom n-rozměrný náhodný vektor $x = (x_1, ..., x_n)^T$ má nulový vektor středních hodnot a známou kovarianční matici $\Sigma \in R^{n\times n}$ s vlastními čísly $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ a jim odpovídajícími vlastními vektory v.

Dále můžeme nalézt nové náhodné veličiny, které vzniknou lineární kombinací původní náhodné veličiny

$$\begin{aligned} y_1 &= v_1^T x = v_{11}x_1 + v_{12}x_2 + ...v_{1n}x_n \\ &\vdots \\ y_n &= v_n^T x = v_{n1}x_1 + v_{n2}x_2 + ...v_{nn}x_n \end{aligned} \quad . \tag{4.9}$$

Pro tyto náhodné veličiny $y_1..y_n$ dostaneme

$$D(y_i) = v_i^T \Sigma v_i, \quad i = 1, ...n, \tag{4.10}$$

$$C(y_i, y_k) = v_i^T \Sigma v_k, \quad i, k = 1, ..n. \tag{4.11}$$

Potom hlavní komponenty jsou také náhodné veličiny $y_1, ..., y_n$, které jsou nenulové a jejich rozptyly jsou maximální vzhledem k podmínce jednoznačnosti $v_i^T v_i = 1$, $i = 1, ...n$.

První komponenta odpovídá y_1, pro který platí

$$\max_{v_1} v_1^T \Sigma v_1, \tag{4.12}$$

vzhledem k podmínce $v_1^T v_1 = 1$.

To lze vyřešit metodou Lagrangeových multiplikátorů. Podrobněji o metodě v [24]. Tato metoda se zabývá hledáním extrémů funkce. Vychází z funkce f, podmínek $h_1, h_2, ..., h_r$, které jsou rovny nule. Potom je dán vztah

$$f + \sum_{i=1}^{n} l_i h_i = 0, \tag{4.13}$$

kde l_i jsou Lagrangeovy multiplikátory.

Definujme si tedy funkci

$$L_1 = v_1^T \Sigma v_1 - l_1(v_1^T v_1 - 1) \tag{4.14}$$

a hledejme hodnotu v_1, pro kterou je maximální.

$$\frac{\partial L_1}{\partial v_1} = 2\Sigma v_1 - 2 l_1 v_1 = 0 \quad tj. \tag{4.15}$$

$$\Sigma v_1 - l_1 v_1 = 0. \tag{4.16}$$

To je vlastnost vlastních čísel matice Σ, tedy l_1 je vlastní číslo a v_1 odpovídající vlastní vektor.

Abychom zjistili o které vlastní číslo a vlastní vektor se jedná, vynásobíme rovnici vektorem v_1^T

$$v_1^T \Sigma v_1 = v_1^T l_1 v_1 \quad tj. \quad D(y_1) = l_1. \tag{4.17}$$

Protože $D(y_1)$ vysvětluje největší část rozptylu, potom $l_1 = \lambda_1$, tedy odpovídá největšímu vlastnímu číslu matice Σ.

Tyto vlastní čísla a vektory získáme z rovnice

$$(\Sigma - \lambda_1 I) = 0. \tag{4.18}$$

To je homogenní soustava lineárních rovnic a λ_1 je největším kořenem charakteristického polynomu

$$det(\Sigma - \lambda_1 I) = 0. \tag{4.19}$$

Druhá komponenta odpovídá y_2, který vysvětluje největší rozptyl nevysvětlený první komponentou. Platí pro něj podmínka jednoznačnosti $v_2^T v_2 = 1$ a podmínka ortogonality s první komponentou $v_2^T v_1 = 0$. Tato úloha lze opět řešit metodou Lagrangeových multiplikátorů, pro kterou definujeme funkci

$$L_2 = v_2^T \Sigma v_2 - l_2(v_2^T v_2 - 1) - l_3(v_2^T v_1). \tag{4.20}$$

Potom

$$\frac{\partial L_2}{\partial v_2} = 2\Sigma v_2 - 2l_2 v_2 - l_3 v_1 = 0. \tag{4.21}$$

Vynásobíme rovnici vektorem v_1^T

$$2v_1^T \Sigma v_2 - 2l_2 v_1^T v_2 - l_3 v_1^T v_1. \tag{4.22}$$

Na základě toho jsme zjistili, že $\Sigma v_2 = l_2 v_2$, což je charakteristika matice Σ. Číslo l_2 tedy odpovídá λ_2, tedy druhému největšímu číslu matice Σ.

Pro vyčíslení dalších komponent postupujeme obdobně.

Z výše popsaného postupu vyplývá vztah pro rozložení kovarianční matice na vlastní čísla a vlastní vektory

$$\Sigma = v\Lambda v^T \tag{4.23}$$

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ : & : & \ddots & : \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}. \tag{4.24}$$

Hlavní komponenty ve výběru

Mějme výběrovou matici X obsahující m pozorování. Každé pozorování je n-rozměrný vektor uspořádaný do řádku $x_i = (x_{i1}, ..., x_{in})$, pro $i = 1, ...m$.

Vektor středních hodnot i kovarianční matice jsou neznámé. Pro odhad vektoru středních hodnot použijeme aritmetický průměr. Pro odhad kovarianční matice použijeme výběrovou

kovarianční matici.

$$\widehat{\mu} = \overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{4.25}$$

$$\widehat{\Sigma} = S = \sum_{i=1}^{m}(x_i - \overline{x})(x_i - x)^T. \tag{4.26}$$

Dále vypočítáme vlastní čísla výběrové matice S, které budou sloužit jako odhady vlastních čísel matice Σ, a jim odpovídající vlastní vektory v.

Z předchozí částí víme, že prvních k vlastních čísel vysvětluje největší rozptyl dat ($k \leq n$). Vezměme tedy pouze prvních k vlastních čísel a jim odpovídajících k vlastních vektorů. Potom

$$P = v(:, 1:k), \quad k \leq n, \tag{4.27}$$

kde P označuje **komponentní zátěže**.

Pomocí komponentních zátěží, lze dopočítat **komponentní skóre** T

$$T = (X - \overline{x}) \cdot P. \tag{4.28}$$

Pro stanovení dostatečného počtu hlavních komponent - čísla k - lze využít následujícího vztahu

$$100\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{m}\lambda_i} \geq P, \tag{4.29}$$

kde P se volí mezi 70-90%.

Použití SVD rozkladu

Pro numerický výpočet metody hlavních komponent je vhodnější použít princip SVD rozkladu, než výpočet přes kovarianční matici.

Mějme výběrovou matici X obsahující m pozorování. Každé pozorování je n-rozměrný vektor uspořádaný do řádku $x_i = (x_{i1}, ..., x_{in})$, pro $i = 1, ...m$.

Vektor středních hodnot je neznámý a pro jeho použijeme aritmetický průměr.

$$\widehat{\mu} = \overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{4.30}$$

Pro použití SVD rozkladu budeme vycházet z centrovaných dat $X_c = X - \overline{x}$. Po provedení rozkladu

$$X_c = UDV^T, \tag{4.31}$$

získáme matici U, která označuje vlastní vektory $X_c X_c^T$. Dále matice V označuje vlastní vektory $X_c^T X_c$. Matice D je matice singulárních čísel, která obsahuje na diagonále prvky $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. Ty jsou druhými odmocninami vlastních čísel kovarianční matice.

$$D = \begin{pmatrix} \sqrt{\lambda_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda_2} & \cdots & 0 \\ : & : & \ddots & : \\ 0 & 0 & \cdots & \sqrt{\lambda_n} \end{pmatrix}. \tag{4.32}$$

Z principu metody hlavních komponent víme, že prvních k vlastních čísel vysvětluje největší variabilitu dat. Na základě diagonálních prvků matice D a vztahu

$$100 \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{m} \sigma_i^2} \geq P, \tag{4.33}$$

lze určit požadované k.

Po vyčíslení prvních k sloupců matice V získáme matici **komponentních zátěží**

$$P = V(:, 1:k), \quad k \leq n. \tag{4.34}$$

Pomocí komponentních zátěží, lze dopočítat **komponentní skóre** T

$$T = X_c \cdot P. \tag{4.35}$$

Vyšetření reziduí

Vzhledem k tomu, že počítáme s odhadem populačních veličin, vzniká v modelu nepřesnost. Ta se dá vyjádřit pomocí matice reziduí. Přesný model lze poté označit vztahem

$$X = T \cdot P^T + E, \tag{4.36}$$

kde E reprezentuje reziduální složku.

Abychom zjistil jak přesný je náš model, potřebujeme vypočítat matici reziduí a ty vyšetřit. Využijeme k tomu vypočítané matice komponentních zátěží a skóre

$$E = X - TP^T. \tag{4.37}$$

Z této matice lze spočítat **rezidua objektu**. Ty určují jak silně hlavní komponenty popisují daný objekt

$$e_i^2 = \sum_{k=1}^{p} e_{ik}^2. \tag{4.38}$$

4.4. Faktorová analýza

Faktorová analýza vychází ze statistického modelu a formulovaných předpokladů. Cílem je vyšetřit strukturu vnitřních vztahů mezi proměnnými. Dalším cílem je redukce dimenze dat a s tím související vysvětlení korelací mezi původními proměnnými pomocí společných faktorů. Následující test je zpracován na základě literatury [14], [5], [13]

Model faktorové analýzy

Nechť x je n-rozměrný náhodný vektor s vektorem středních hodnot μ. Potom definujeme statistický model faktorové analýzy

$$x = \mu + \Gamma f + \varepsilon, \tag{4.39}$$

kde Γ je matice faktorových zátěží s koeficienty γ_{ij}, f je vektor společných faktorů a ε je vektor specifických faktorů.

Dále vezměme normovaný vektor $z = (z_1, z_2, ..., z_n)^T$, který vznikl pomocí vztahu

$$z = \frac{x - \mu}{s(x)}, \tag{4.40}$$

kde $s(x)$ je směrodatná odchylka vektoru x.

Model faktorové analýzy můžeme považovat za speciální případ lineární regrese. Regresní rovnici rozepíšeme po prvních a dostaneme

$$\begin{aligned} z_1 &= \gamma_{11} f_1 + \gamma_{12} f_2 + ... \gamma_{1p} f_p + \varepsilon_1 \\ &\vdots \\ z_n &= \gamma_{n1} f_1 + \gamma_{n2} f_2 + ... \gamma_{np} f_p + \varepsilon_n \end{aligned}, \tag{4.41}$$

kde $f = (f_1, f_2, ..., f_p)^T$ je vektor společných faktorů. Rovnice přepíšeme v maticové formě

$$\begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1p} \\ \vdots & \ddots & & \vdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & \gamma_{np} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ \vdots \\ f_p \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}. \tag{4.42}$$

Vzhledem k použití normované veličiny **z** platí, že střední hodnoty $E(f) = E(\varepsilon) = 0$. Dále platí, že kovarianční matice $C(f) = I$ a $C(\varepsilon) = \Psi$, kde Ψ je diagonální matice chybových rozptylů. Vzhledem k nezávislosti chybového vektoru a vektoru společných faktorů, můžeme definovat kovarianci $C(f, \varepsilon) = 0$. Při použití těchto podmínek docházíme k rozkladu kovarianční matice Σ

$$\Sigma = \Gamma\Gamma^T + \Psi. \tag{4.43}$$

Odhad parametrů modelu

Mějme datovou matici Z typu $n \times p$ pro p proměnných. Dále mějme matici společných r faktorů. Dále mějme matici neznámých faktorové zátěže typu $p \times r$. Model můžeme napsat tedy ve tvaru

$$Z = F\Gamma^T. \tag{4.44}$$

Pro výpočet faktorových zátěží můžeme vycházet z výběrové kovarianční matice. Pomocí metody maximální věrohodnosti odhadneme parametry Γ, Ψ. Úlohu lze definovat jako maximalizaci věrohodností funkce

$$\ln[L(\Lambda, \Psi)] = -\frac{1}{2}(n-1)[\ln(\det(\Gamma\Gamma^T + \Psi)) + st(\Gamma\Gamma^T + \Psi)\widehat{\Sigma}], \tag{4.45}$$

kde $\det(\cdot)$ značí determinant a $st(\cdot)$ stupeň matice.

Nyní máme známou matici faktorových zátěží Γ a matici chybových rozptylu Ψ. Použitím vhodné regresní metody dopočítáme matici společných faktorů F.

Rotace faktorů

Pomocí rotace faktorových zátěží nalezneme jednodušší struktury pro snadnější reprezentaci. Cílem je tedy získat řešení pro které platí následující podmínky.

Každý řádek Γ obsahuje nulový prvek. Každý sloupec alespoň tolik nulových prvků, kolik je faktorů. Pro každou dvojici sloupců následně platí, že některé znaky by měli být významné v jednom sloupci a nevýznamné v druhém. Každá dvojice sloupců by měla mít co nejmenší počet proměnných s vysokými zátěžemi.

Definujme matici rotace R. Za podmínky ortogonality $R^{-1} = R^T$ a $R^T R = I$. Potom rotace faktorových zátěží označme Γ^* a rotaci faktorového skóre F^*

$$F^* = FR \tag{4.46}$$

$$\Gamma^* = \Gamma R. \tag{4.47}$$

Po úpravě faktorového modelu pro rotované matice je zřejmé, že odpovídají modelu originálnímu. Tím lze ukázat, že rotací faktorů lze zjednodušit reprezentace faktorů bez ovlivnění modelu

$$Z = F^*\Gamma^{*T} = FRR^T\Gamma^T = FI\Gamma^T = F\Gamma^T. \tag{4.48}$$

Mezi nejpoužívanější ortogonální rotace patří metoda Varimax, Quartimax, Orthomax. Dále se využívají šikmé rotace, např. Oblimax, Promax.

4.5. Shluková analýza

Shluková analýza se využívá v úlohách klasifikačních. Používá se pro určení struktury ve znacích, tedy pro rozdělení objektů do kategorií. Hlavním rozdílem oproti úlohám diskriminačním je ta, že se musí předem určit počet shluků, do kterých budou prvky následně klasifikovány. Pro tyto metody se často používá pojem *unsupervised learning methods* - učení bez učitele. Podrobnější informace jsou v publikacích [5], [14], [18], [9].

Předpoklady analýzy shluků

Požadavky normality, homoskedasticity (homogenní rozptyl) nemají na tuto metodu silný vliv. Naopak je nutné počítat s reprezentativností vzorku a vlivem multikolinearity (lineární závislost mezi znaky). U reprezentativnosti je nutné počítat s tím, že analýza shluků odpovídá pouze reprezentativnosti výběru. U výskytu multikolinearity je potřeba zredukovat počet znaků, nebo zvolit vhodnou míru, např. Mahalanobisova.

Předúprava dat

Před vlastní analýzou je potřeba rozhodnout, zda je nutné data standardizovat. Pro standardizaci se běžně používá převod každého znaku na Z-skóre. Existují ale i jiné formy standardizace:

Autoškálování - transformace na Z-skóre

$$z_i = \frac{(x_i - \overline{x_i})}{\sqrt{s^2}}. \tag{4.49}$$

Sloupcové centrování

$$y_{ij} = x_{ij} - \overline{x_j}. \tag{4.50}$$

Sloupcová standardizace

$$y_{ij} = \frac{x_{ij}}{s_j}. \tag{4.51}$$

Škálování sloupcovým rozsahem

$$y_{ij} = \frac{x_{ij} - \min_j x_{ij}}{\max_j x_{ij} - \min_j x_{ij}}. \tag{4.52}$$

Obdobně jako sloupcové centrování a standardizace se provádí také řádkové a celkové. Rozdíl je pouze v použitém průměru a směrodatné odchylky.

Míry vzdálenosti a podobnosti

Důležitým prvkem je určení, na jakém základě budou prvky shlukovány. Jaká metrika bude určovat podobnost objektů. Nejčastěji používané jsou míry vzdálenosti:

Eukleidovská vzdálenost vychází z výpočtu pravoúhlého trojúhelníka

$$D_e(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}. \tag{4.53}$$

Hammingova (manhattanská) vzdálenost

$$D_h(x_i, x_j) = \sum_{k=1}^{m} \mid x_{ik} - x_{jk} \mid . \tag{4.54}$$

Čebyšeova vzdálenost

$$D_c(x_i, x_j) = \max \mid x_{ik} - x_{jk} \mid . \tag{4.55}$$

Kosinova vzdálenost

$$D_{cos}(x_i, x_j) = 1 - \frac{x_i x_j^T}{\sqrt{(x_i x_i^T)(x_j x_j^T)}}. \tag{4.56}$$

Mahalanobisova vzdálenost uvažuje vztahy mezi znaky. Používá kovarianční matici S

$$D_{ma}(x_i, x_j) = \sqrt{(x_i - x_j)^T S^{-1}(x_i - x_j)}. \tag{4.57}$$

Hierarchické shlukování

Základem je uspořádání objektů a jejich shluků. Hierarchické algoritmy produkují stromovité struktury. Patří sem dva základní - aglomerační a divizní. K zobrazení se používá hierarchický dendrogram.

Aglomerační algoritmus vychází z předpokladu, že každý objekt je samostatný jeden shluk. Postup spočívá ve spojení dvou shluků (v 1. kroku objektů) s nejmenší vzdáleností. V každém dalším kroku se přepočítává matice vzdáleností a s tou se postup opakuje. Algoritmus končí v momentě vytvoření jednoho shluku.

Divizní algoritmus vychází z předpokladu, že všechny objekty tvoří jeden shluk. Z toho se postupným dělením a oddělováním nepodobných objektů získávají dílčí shluky. Proces se opakuje dokud není v jednom shluku jeden objekt. Dělení závisí na zvolené metodě. Patří sem např. metoda nejbližšího souseda, nejvzdálenějšího souseda, průměrová, Wardova, metoda těžiště.

Nehierarchické shlukování

Principem metody je rozdělení n objektů o m znacích do k shluků, přičemž se meziskupinová suma čtverců minimalizuje. Jedná se o postup iterační. Označují se jako metody nejbližších těžišť (*K-means*).

Pokud jsou těžiště shluků známá, stačí provést hierarchickou analýzu, určit počet a počáteční těžiště shluků.

Pokud nejsou známá je třeba data vyšetřovat opakovaně. Nejprve zvolit prvních k objektů jako dočasná těžiště. Pokud má objekt nejmenší vzdálenost k těžišti větší než mezi dvěma nejbližšími, poté objekt nahradí těžiště, které je blíž k objektu.

5. Metodologie zpracování dat v prostředí MATLAB

Tato kapitola se bude zabývat procesem zpracování dat v prostředí MATLAB. Popíší se zde základní datové struktury, možnosti načítání dat do nich a přístup k nim. Dále se zde popíší metody pro základní popis jednotlivých proměnných (deskriptivní statistiky). Nakonec se zde rozeberou základní grafy pro reprezentaci dat, které se budou v této práci využívat.

5.1. Příprava a standardizace dat

Během procesu přípravy dat se musíme vypořádat s mnoha překážkami. První věcí je identifikovat, jaký typ vstupních dat máme k dispozici. Pokud máme načteny grafická data, nebo video data, tak je zpracujeme jinak, než čistě numerická data. Dalším problémem jsou chybějící hodnoty. Pokud datová množina nějaké obsahuje, může to nepříznivě ovlivnit výsledky analýzy. Je tedy potřeba vhodným způsobem hodnoty odstranit. Další částí je standardizace dat. Některé analytické metody, které porovnávají proměnné, nemohou dojít k reprezentativním výsledkům, pokud jsou data v různém měřítku.

Načítání datového souboru

Do prostředí MATLAB lze importovat různé datové soubory. Může se jednat o data obrazová, numerická, či video data. V této práci se bude pracovat převážně s formáty **CSV, JPEG**. Pro import budeme tedy používat příkazy `csvread` a `imread`.

Přehled vybraných formátů, včetně příkazů pro import, je zobrazen v tabulce 5.1. De-

tailnější informace poskytuje příkaz `help fileformats` a publikace [7].

Typ souboru	Příkaz	Typ proměnné
MAT	`load`	MATLAB proměnné
CSV	`csvread`	pole typu DOUBLE
DAT	`importdata`	pole typu DOUBLE
DLM	`dlmread`	pole typu DOUBLE
TAB	`dlmread`	pole typu DOUBLE
XLS	`xlsread`	pole typu DOUBLE, CELL
CDF	`cdfread`	pole CELL typů CDF
AVI	`aviread`	MATLAB movie
PNG	`imread`	IMAGE (pole INT8, DOUBLE, ...)
JPEG	`imread`	IMAGE (pole INT8, DOUBLE, ...)
BMP	`imread`	IMAGE (pole INT8, DOUBLE, ...)
...	...	...

Tabulka 5.1.: Vybrané formáty vstupních dat

Vzhledem k různým typům vstupních dat existují i různé datové typy. Přehled je zobrazen v tabulce 5.2. Pro převod mezi numerickými typy slouží přetypování. Tyto funkce mají stejný název jako cílový datový typ. Například pro převod do typu double, použijeme funkci `double(variable)`. Pro převod řetězce na numerický typ slouží funkce `str2num`, `str2double`, `str2int`. Z číselné hodnoty na řetězec zase funkce `num2str`, `int2str`, `double2str`.

Kromě primitivních datových typů, existuje ještě typ *Cell*. Jedná se o speciální datovou strukturu, která může obsahovat proměnné různých typů. Pro vytvoření prázdné struktury slouží příkaz `cell`. Pro adresování prvků máme dva možné přístupy. První z nich se nazývá *cell adressing* a využívá hranatých závorek. Tento typ odkazuje na konkrétní buňku, ne však na její obsah. Druhý přístup se nazývá *content adressing* a využívá složených závorek. Tímto přistupujeme přímo ke konkrétním datům buňky.

```
A = cell(1, 2)              %Output: [] [] (1 row, 2 col)
A{1, 1} = [1, 2, 3];
A{1, 1}                     %Output: 1 2 3
A[1, 1]                     %Output: [1x3 double]
```

Datový typ	Popis
`uint8`	bezznaménkový celočíselný datový typ, rozsah 0 až 255
`int8`	znaménkový celočíselný datový typ, rozsah -128 až 127
`uint16`	bezznaménkový celočíselný datový typ, rozsah 0 až 2^{16}
`int16`	znaménkový celočíselný datový typ, rozsah -2^{15} až 2^{15}
`uint32`	bezznaménkový celočíselný datový typ, rozsah 0 až 2^{32}
`int32`	znaménkový celočíselný datový typ, rozsah -2^{31} až $2^{31}-1$
`uint64`	bezznaménkový celočíselný datový typ, rozsah 0 až 2^{64}
`int64`	znaménkový celočíselný datový typ, rozsah -2^{63} až $2^{63}-1$
`double`	datový typ s plovoucí desetinnou čárkou, dvojnásobná přesnost
`single`	datový typ s plovoucí desetinnou čárkou, jednonásobná přesnost
`char`	řetězcový datový typ.

Tabulka 5.2.: Datové typy proměnných

Práce s chybějícími hodnotami

Pokud načtený datový set obsahuje hodnoty, které byly poškozeny, nezadány, či jinak znehodnoceny, říkáme že obsahuje **chybějící hodnoty** (též missing values). V prostředí MATLAB lze tyto hodnoty identifikovat pomocí funkce `isnan()`. Výsledkem je pole logických indikátorů. Tam kde pole indikuje hodnotu 1 se nachází chybějící hodnota. Tímto způsobem lze indikovat pouze hodnoty systémově chybějící. Pro uživatelsky definované, je potřeba datová množina předupravit jiným způsobem. Po identifikaci je plně na uživateli jak se s nimi vypořádá.

První možností vypořádání s chybějícími hodnotami je odstranění celých záznamů. K tomu lze využít funkci `any()`. Ta testuje jestli existuje nenulový prvek v zadané dimenzi (první dimenze značí sloupec, druhá značí řádek, atd.). Pro následující analýzu se bude používat redukovaný datový set obsahující pouze úplné záznamy.

```
A = [ 1 NaN 5; NaN 9 3] % Input data with missing values
idx = isnan(A)  % Logical matrix: 1 - missing, 0 - number
cm = any(idx)   % Logical vector: 1 - missing in first
   dimmension (column)
rm = any(idx,2) % Logical vector: 1 - missing in second
   dimmension (row)
A(:,cm) = []    % Remove columns with missing values
A(rm,:) = []    % Remove rows with missing values
```

Druhou možností je nahrazení chybějící hodnoty. Náhrada pomocí předchozí, nebo následující hodnoty se musí provést ručně. Pro náhradu pomocí interpolace lze použít interpolační funkce. Tomuto způsobu se nebudeme v práci věnovat.

Poznámka: Od vydání MATLABu verze 2016b obsahuje SW pokročilé funkce na práci s chybějícími hodnotami. Jejich výčet je zobrazen v tabulce 5.3. V této práci se s nimi však pracovat nebude.

Funkce	**Popis**
`ismissing()`	- Vrací pole logických indikátorů pro chybějící hodnoty. Pomocí parametru lze nastavit indikaci systémově chybějících nebo uživatelsky definovaných hodnot.
`replaceWithMissing()`	- Vrací původní datový set, ve kterém nově nastaví uživatelsky definované chybějící hodnoty na systémově chybějící.
`rmmissing()`	- Vrací původní datový set, ve kterém odstraní celé záznamy obsahující chybějící hodnoty.
`fillmissing()`	- Umožňuje dopočítat chybějící hodnotu na základě definované metody. Lze použít metoda předchozí, nebo následující hodnoty, interpolace, aj.

Tabulka 5.3.: MATLAB 2016b funkce pro missing hodnoty

Standardizace dat

Pokud je měřítko jednotlivých proměnných diametrálně rozdílné, je potřeba přistoupit k procesu normalizace. Ten transformuje proměnné do stejného měřítka, ve kterém je lze porovnávat. Více v publikaci [15].

Mějme datový set obsahující proměnné znázorňující příjem klienta a věk klient, *income* a *age*. Proměnná označující věk klienta nabývá řádově rozdílných hodnot, než příjem. Pokud budeme chtít obě proměnné porovnat, poté bude příjem klienta přispívat větší váhou. Tomuto lze předejít použitím správné metody normalizace.

První metoda, které se budeme věnovat, je převod na obecný interval hodnot $< a, b >$

Tu označuje vzorec 5.1. Po úpravě na interval $< 0, 1 >$ dostaneme vzorec 5.2

$$.y = \frac{x - x_{min}}{x_{max} - x_{min}}(b - a) + a \tag{5.1}$$

$$.y = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{5.2}$$

Druhou metodou je převod na z-skóre. Ten spočívá v odečtení průměrné hodnoty a vydělením směrodatnou odchylkou. Tato metoda je vhodná pro převod veličiny na normované normální rozdělení N(0,1).

$$z = \frac{x - \overline{x}}{s(x)} \tag{5.3}$$

```
x = [2 5 8 1]   % Input testing values

% First normalize method
y = (x - min(x))/(max(x) - min(x))      % Interval 0, 1
% Output: 0.14286   0.57143   1.00000   0.00000

% Second normalize method
zscore(x)
% Output: -0.63246   0.31623   1.26491   -0.94868
```

5.2. Průzkumová analýza vícerozměrných dat

Po importu dat a vhodné úpravě datového setu, potřebujeme zjistit základní informace o proměnných. K tomu lze využít prvky deskriptivní statistiky. Pro celkový pohled na data, zde využijeme testy vícerozměrné normality.

Průzkum jednotlivých proměnných, deskriptivní statistiky

Pro průzkum jednotlivých proměnných slouží prvky deskriptivní statistiky. Jedná se míry polohy (průměr, medián, kvantily) a variability (variační a mezikvartilové rozpětí, rozptyl, směrodatná odchylka).

Aritmetický průměr je dán vzorcem

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{5.4}$$

Rozptyl a směrodatná odchylka se vypočítá jako

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2, s = \sqrt{s^2} \tag{5.5}$$

```
x = [1 3 5]
mean(x)                    % Mean - Output: 3
median(x)                  % Median - Output: 3
max(x)                     % Maximum - Output: 5
min(x)                     % Minimum - Output: 1
std(x)                     % Standard deviation - Output: 2
var(x)                     % Variance - Output: 4
quantile(x, 0.25)          % 25% quantile - Output: 1.5
quantile(x, 0.75)          % 75% quantile - Output: 3.5
```

Test vícerozměrné normality na základě testů šikmosti a špičatosti

Pro testování vícerozměrné normality dat nejsou va MATLABu žádné funkce. Většina se soustředí na testování jednorozměrné normality dat. V této práci si ukážeme výpočet a jeho implementaci pro test normality založený na vícerozměrných šikmostech a špičatostech. Teorie vychází z publikace [14].

Pro různé statistické testy je nutné ověřovat zda výběr pochází z normálních rozdělení. Mějme data $X = (x_1, x_2, ..., x_m)$ z vícerozměrného normálního rozdělení daného vektorem středních hodnot a kovarianční maticí $N_m(\mu, \Sigma)$.

Pro odhad vektoru středních hodnot se dá použít vztah

$$\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{5.6}$$

Pro odhad kovarianční matice se používá výběrová kovarianční matice S

$$\widehat{\Sigma} = S = \frac{1}{n-1}(x - \widehat{\mu})(x - \widehat{\mu})^T. \tag{5.7}$$

Pro odhad vícerozměrných šikmostí ($\widehat{g_{1,m}}$) a špičatostí ($\widehat{g_{2,m}}$) je použit čtverec Mahalanobisovy vzdálenosti

$$d^2_{MA} = (x_i - \overline{x})^T S^{-1}(x_j - \overline{x}), \tag{5.8}$$

$$\widehat{g1} = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{[(x_i - \overline{x})^T S^{-1}(x_j - \overline{x})]^3}{n^2}, \tag{5.9}$$

$$\widehat{g2} = \sum_{i=1}^{n} \frac{[(x_i - \overline{x})^T S^{-1}(x_i - \overline{x})]^2}{n}. \tag{5.10}$$

Pokud jsou prvky vybrány na základě náhodného výběru, poté platí veličiny U_1, U_2

$$U_1 = \frac{n\widehat{g_{1,m}}}{6}, \tag{5.11}$$

$$U_2 = \frac{\widehat{g_{2,m}} - m(m-2)}{\sqrt{8m\frac{m+2}{n}}}. \tag{5.12}$$

Veličina U_1 má asymptoticky rozdělení χ^2 s $\frac{m(m+1)(m+2)}{6}$ stupni volnosti. Veličina U_2 má asymptoticky normované normální rozdělení $N(0,1)$.

Nyní následuje vlastní implementace. Nejprve implementujeme výpočet šikmostí U_1 na základě výše popsané teorie. Proměnné `row` a `col` označují počet řádků a sloupců vstupních dat. Proměnná `S` označuje kovarianční matici. Pro její výpočet lze použít příkaz `S = cov(data)`.

```
means = mean(data(:,1:col));
g1 = 0;
for i = 1:row
        r1 = data(i,:) - means;
        for j = 1:row
                r2 = data(j,:) - means;
                w = (r1/S)*r2';
                g1 = g1 + ((w^3) / (row^2));
        end
end
U1 = row*g1/6;
```

Nyní je potřeba spočítat testové kritérium veličiny U_1. Jedná se o kvantil chí-kvadrát rozdělení. Dále se spočítá p-hodnota veličiny U_1.

```
U1stat = chi2inv(0.95,col*(col+1)*(col+2)/6);
U1pvalue = 1-chi2cdf(U1,col*(col+1)*(col+2)/6);
```

Obdobně implementujeme výpočet špičatosti U_2, testového kritéria a p-hodnoty.

```
g2 = 0;
for i = 1:row
        r = data(i,:) - means;
        w = (r/S)*r';
        g2 = g2 + ((w^2) / row);
end
U2 = (g2 - (col*(col+2)))/sqrt(8*col*(col+2)/row);
U2stat = norminv(0.95,0,1);
U2pvalue = 1-normcdf(U2);
```

5.3. Interpretace výsledků

Výsledky samotné analýzy se mohou zobrazovat různými způsoby. Může se jednat o výstup pomocí tabulky hodnot nebo grafu. V této části se budeme zabývat především grafickou reprezentací a jejími možnosti v MATLABu.

Bodový graf a jeho varianty

U bodového grafu se zobrazují hodnoty jako samostatně ležící body, které nejsou nijak spojeny s ostatními. Může se jednat o zobrazení ve 2D nebo 3D prostoru. Více v publikacích [14], [5]. Podle toho jakou vstupní veličinu vizualizujeme, rozdělujeme grafy na různé varianty.

První variantou bodového grafu bude **graf komponentních zátěží**. Tento graf zobrazuje vztahy mezi původními proměnnými a komponenty. Zobrazení probíhá na základě korelace.

Druhou variantou je **graf komponentního skóre**. Ten zobrazuje skóre pro první 2 až 3 hlavní komponenty. Pomocí něj, lze identifikovat odlehlé hodnoty, trendy či shluky.

Kombinací obou grafů dostaneme dvojný graf, tzv **Biplot**. Ten zobrazuje proměnné ve formě úseček a jednotlivé body. Body, které jsou v interakci, poté leží blízko proměnných.

Další variantou je např. **graf reziduí**. Díky němu, lze zobrazit rozptyl reziduí. Ten značí vzdálenost mezi objektem a modelem. Tím lze posoudit vhodnost modelu.

Pro příklad si vizualizujme veličinu $X_{m,n}$. Uvažujme zde obecnou veličinu, tedy obecný bodový graf. K vizualizaci 2D grafu slouží příkaz `plot()`, pro 3D graf `plot3()`. Parametry grafu, jako barva, styl spojení bodů je přidám jako řetězcový vstup. Barvy se označují symboly `b, g, r, c, m, y, k, w`. Spojení pro bodový graf označuje symbol kroužku (malé o). Pro zobrazení dvojného grafu, však slouží přímo příkaz `biplot()`. Ten přijímá jako vstup komponentní zátěže. V případě přidání komponentního skóre, je potřeba tento parametr specifikovat.

```
plot(X(:1), X(:,2) 'o') % Show 2D scatter plot
plot3(X(:,1), X(:,2), X(:,3) 'o')       % Show 3D scatter
   plot
biplot(loadings, 'score', score)        % Show Biplot
```

Graf úpatí vlastních čísel

Tento graf slouží pro identifikaci potřebného počtu hlavních komponent. Zobrazuje relativní velikost vlastních čísel. Tato čísla musí tvořit seřazenou posloupnost. Tento typ grafu není v MATLABu k dispozici, proto je nutné ho implementovat.

Vstupem zde bude vektor vlastních čísel $A = (\lambda_1; \lambda_2; ...\lambda_n)$, kde $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n)$. Bude je jednat o 2D graf, použije se tedy funkce `plot`. Graf zobrazuje závislost vlastních čísel na pořadí prvků. Dále budeme požadovat aby jednotlivé body byly spojeny úsečkou. Vykreslení grafu všech vlastních čísel je poté zobrazeno na následujícím kódu.

```
plot(1:i, A(1:i), 'ko-');
title('Scree plot');
xlabel('Eigen value index');
ylabel('Eigen value');
```

Ikonové grafy

Pro explorační analýzu jsou vhodné, také ikonové grafy. Ty poskytují určitý pohled na celkový datový soubor. Zobrazují jednotlivá pozorování ve formě objektů. Tyto objekty mohou být obličeje - poté se jedná o graf obličejový. Nebo to mohou být hvězdy - poté se jedná o graf hvězdicový. Díky těmto grafům. lze odhadnout shluky proměnných, extrémní pozorování, a další vlastnosti. Více v publikaci [5].

Pro oba typy grafů je v prostředí MATLAB je implementována funkce `glyphplot`. Vstupem metody je celý datový set. Dále se parametrem `'glyph'` dá nastavit typ grafu.

```
glyphplot(data, 'glyph','star') % Show star plot
glyphplot(data, 'glyph','face') % Show face plot
```

Grafické ověření vícerozměrné normality

Pro grafické ověření normality dat se využívají Q-Q grafy nebo p-p grafy. V MATLABu jsou tyto grafy k dispozici pro ověření normality jedné proměnné. Pro vícerozměrná data, však k dispozici nejsou. Pro tento případ je tedy nutné grafy naprogramovat. V této práci se budeme zabývat Q-Q grafem. Více v monografii [20].

Před vlastní implementací, je důležité pochopit, co vlastně budeme programovat. Vycházejme tedy definice vícerozměrného normálního rozdělení. Poté

$$\Sigma_i = [\frac{n}{(n-1)^2}](x_i - \overline{x})^T S^{-1}(x_i - \overline{x}). \tag{5.13}$$

Po úpravě dostaneme čtverec Mahalanobisovy vzdálenosti bodu x_i od průměru $\overline{x}$

$$\frac{\Sigma_i}{\frac{n}{(n-1)^2}} = (x_i - \overline{x})^T S^{-1}(x_i - \overline{x}), \tag{5.14}$$

$$d^2_{MA} = \Sigma_i \frac{(n-1)^2}{n} = (x_i - \overline{x})^T S^{-1}(x_i - \overline{x}). \tag{5.15}$$

Čtverec Mahalanobisovy vzdálenosti má asymptoticky χ^2 rozdělení s m stupni volnosti. Do grafu se následně vynáší d^2_{MA} v závislosti na kvantilech $q_i = \chi^2[(i - 0,5)/n]$, kde přímka značí normalitu dat.

Nyní následuje vlastní implementace. Nejprve začneme výpočtem čtverce Mahalanobisovy vzdálenosti d^2_{MA} podle výše zmíněného vzorce, kde `S` značí kovarianční matici, `y` značí sloupcový vektor mezivýsledků a `means` značí průměrnou hodnotu. Samotný výpočet spočívá v projíti každého řádku vstupních dat, odečtením průměrné hodnoty a vynásobením s kovarianční maticí. Výsledný sloupcový vektor se seřadí podle velikosti pomocí funkce `sort(y)`.

```
means = mean(data(:,1:col));
S = cov(data);
for j=1:row
    r=data(j,:);
    r = r-means;
    y(j) = (r/S)*r';
end
dist = sort(y);
```

Nyní je potřeba spočítat kvantily chí-kvadrát rozdělení pro každý záznam. K tomu slouží funkce `chi2inv()`. Nakonec se pomocí `plot(dist, chi)` vytvoří Q-Q graf normality.

```
for j=1:row
        y(j) = chi2inv((j-0.5)/row,col);
end
chi = y;
```

6. Aplikace vybraných metod analýzy vícerozměrných dat

Po teoretickém úvodu do problematiky analýzy vícerozměrných dat a prostředí MATLAB se přejde k reálnému využití zmiňovaných metod. Budou představeny aplikace na metodu Hlavních komponent a s ní související SVD rozklad, dále na Faktorovou analýzu a Shlukovou analýzu. Vzhledem k širokému spektru využití budou vybrány dvě, zcela odlišné, oblasti pro metody hlavních komponent a shlukovou analýzu. Pro faktorovou analýzu bude použita jedna ukázka.

Metoda hlavních komponent bude použita pro kompresi obrazu a demografická data. Faktorová analýza na sociálních datech. Shluková analýza na demografických datech a text miningu.

6.1. SVD rozklad a metoda Hlavních komponent v prostředí MATLAB

Metoda hlavních komponent má důležité využití při redukci rozměru úlohy, tedy snížení počtu proměnných, které se lépe vizualizují v 2D nebo 3D grafu. Redukce nadbytečných a korelovaných dat se dá také využít při komprimaci obrazových dat.

Obecný postup použití metody

Máme datový soubor obsahující jednotlivá pozorování (objekty), které jsou popsány několika atributy.

Pokud používáme grafická data, tak řádky datového souboru reprezentují řádky pixelů obrázku. Sloupce datového souboru reprezentují sloupce pixelů. Ve výchozím stavu se tedy provádí řádková komprese.

Pro samotný výpočet využijeme metodu SVD rozkladu popsanou v kapitole 4.3.

Vstupní datový soubor načteme jako příslušnou matici X. Odečtením průměru zajistíme centrování matice X_c. Tu následně rozložíme na matici vlastních vektorů V a vlastních čísel D^2

$$X_c = UDV.$$

Na základě grafu úpatí vlastních čísel se vybere prvních k komponent a vytvoří se matice komponentních zátěží P. Dále se dopočte komponentní skóre T, které se dá zobrazit v grafu. Pro zobrazení ve 2D, nebo 3D grafu se volí 2 - 3 hlavní komponenty

$$P = V(:, 1:2),$$

$$T = X_c P.$$

Pokud chceme získat matici původní velikosti, ale s komprimovanými hodnotami (např. grafická data v komprimované podobě), použijeme vztah

$$X_c = TP^T.$$

Samotné analýze může předcházet klasická průzkumová analýza. Tedy výpočet deskriptivních statistik, průzkum normality, aj. Podrobněji v kapitole 5.

Příklad 6.1.1: Demografická data

Pro analýzu je vybrán datový soubor popisující vliv sociálně ekonomických ukazatelů pro porovnání jednotlivých okresů v České republice. Data čerpána z [3].

Vstupní matice má *76* objektů. Každý objekt reprezentuje jeden okres ČR a je definován *11* proměnnými. Jedná se o:

- Počet cizinců
- Počet trestných činů
- Počet obyvatel - muži
- Počet obyvatel - ženy
- Průměrný věk - muži
- Průměrný věk - ženy
- Sňatky
- Průměrná výše důchodu - muži
- Průměrná výše důchodu - ženy
- Plocha území
- Koeficient ekologické stability

Rozměr datového souboru je tedy *11*, avšak z prvního pohledu na data vyvstává otázka, jestli kategorie muži a ženy nejdou nahradit jednou a tím rozměr úlohy snížit. Nebo jestli nelze "sloučit" více proměnných, tedy zjistit které proměnné jsou mezi sebou silně korelované a nahradit je novými. Tím se zjednoduší popis objektů a mohou se lépe vizualizovat. To je také cílem tohoto příkladu.

Nejprve je potřeba si připravit datovou matici okresů. Tato matice musí obsahovat pouze numerické hodnoty oddělené čárkou. Názvy proměnných, nebo objektů můžeme připravit do externích souborů.

V první fázi se načteme datovou matici. Tím, že se jedná o čistě numerická data použijeme funkci `csvread()`. Volitelně můžeme načíst názvy objektů. Na rozdíl od datové matice, zde se jedná o řetězcové hodnoty. Musíme použít funkci `textscan()`. Ten, ale přijímá jako vstupní parametr deskriptor souboru. Ten získáme pomocí funkce `fopen()`.

```
data = csvread(dataFileName);
objId = fopen(labelsFileName);
labels = textscan(objId, '%s', 'Delimiter', ',');
fclose(objFileId);
```

Nyní se budeme zabývat základním průzkumem dat. Vypočítáme deskriptivní statistiky, vyšetříme normalitu dat a pro určitý pohled na variabilitu dat si vykreslíme ikonové grafy.

Pro základní přehled o datech si vypočítáme deskriptivní statistiky. Jedná se především o aritmetický průměr, 1 až 3. kvartil, minimum, maximum, rozptyl, popř. směrodatná odchylka. Použijeme na to funkce z kapitoly 5.2. Tím můžeme získat určitý pohled na variabilitu proměnných. Výsledek v tabulce 6.1.

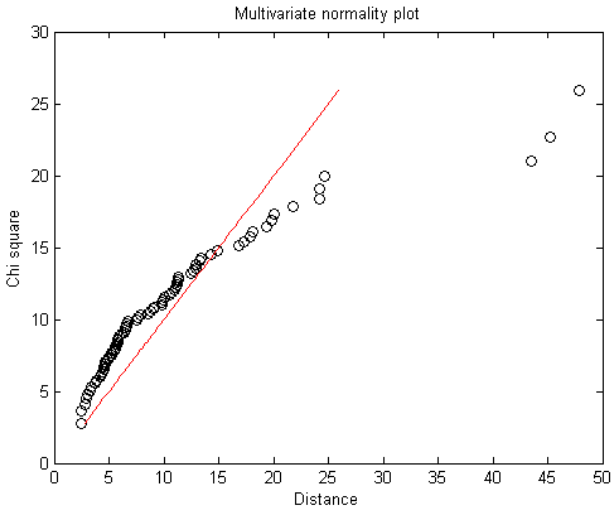
Průměr	**Medián**	**Max**	**Min**	**Rozptyl**	**1. kvartil**	**3. kvartil**
$3,85 \cdot 10^3$	$2,53 \cdot 10^3$	$2,49 \cdot 10^4$	811	$1,34 \cdot 10^7$	$1,73 \cdot 10^3$	$4,78 \cdot 10^3$
$2,41 \cdot 10^3$	$1,84 \cdot 10^3$	$1,29 \cdot 10^4$	638	$4,37 \cdot 10^6$	$1,26 \cdot 10^3$	$2,88 \cdot 10^3$
$6,02 \cdot 10^4$	$5,49 \cdot 10^4$	$1,82 \cdot 10^5$	$1,95 \cdot 10^4$	$7,93 \cdot 10^8$	$4,42 \cdot 10^4$	$6,92 \cdot 10^4$
$6,20 \cdot 10^4$	$5,64 \cdot 10^4$	$1,95E \cdot 10^5$	$1,97 \cdot 10^4$	$9,04 \cdot 10^8$	$4,54 \cdot 10^4$	$7,15 \cdot 10^4$
40,5	40,5	41,6	37,5	0,5	40,1	40,9
43,2	43,3	44,4	39,4	0,87	42,8	43,9
36,7	32,0	140	10	472	24	42
$1,16 \cdot 10^4$	$1,15 \cdot 10^4$	$1,3 \cdot 10^4$	$1,09 \cdot 10^4$	$1,24 \cdot 10^5$	$1,14 \cdot 10^4$	$1,17 \cdot 10^4$
$1,03 \cdot 10^4$	$1,03 \cdot 10^4$	$1,09 \cdot 10^4$	$9,89 \cdot 10^3$	$4,16 \cdot 10^4$	$1,02 \cdot 10^4$	$1,05 \cdot 10^4$
$1,03 \cdot 10^5$	$1,01 \cdot 10^5$	$1,95 \cdot 10^5$	$2,30 \cdot 10^4$	$1,46 \cdot 10^9$	$7,83 \cdot 10^5$	$1,29 \cdot 10^5$
1,28	0,9	4,5	0,3	0,85	0,65	1,75

Tabulka 6.1.: Popisné statistiky proměnných matice okresů
(Proměnné jsou po řádcích: Počet cizinců, Počet trestných činů, Počet obyvatel - muži, Počet obyvatel - ženy, Průměrný věk - muži, Průměrný věk - ženy, Sňatky, Průměrná výše důchodu - muži, Průměrná výše důchodu - ženy, Plocha území, Koeficient ekologické stability)

Pro testování vícerozměrné šikmosti a špičatosti jsme využili funkci `NormalityTests()`, kterou jsme si připravili na základě kapitoly 5.2. Po zhodnocení výsledků (Tabulka 6.2) můžeme rozhodnout, že vícerozměrná šikmost, ani špičatost neodpovídá normálnímu rozdělení. Normalita dat, tedy není splněna. Pro lepší vizualizaci použijeme q-q graf využitím funkce `MultivariateQQPlot()`, kterou jsme vytvořili na základě kapitoly 5.3. Výsledek na obrázku 6.1.

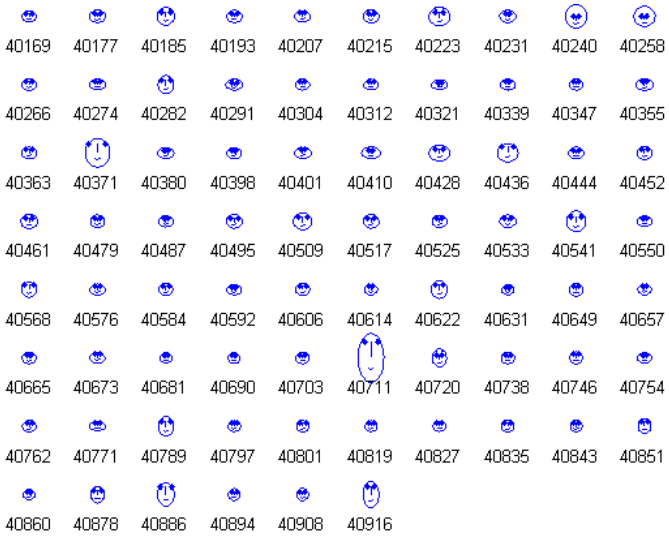
Veličina	**Statistika**	**Testové kritérium**	**p hodnota**	**Přesnost**
Šikmost	1 033,90	326,44	0	0,05
Špičatost	14,26	1,65	0	0,05

Tabulka 6.2.: Test vícerozměrné šikmosti a špičatosti matice okresů



Obrázek 6.1.: Q-Q graf vícerozměrné normality matice okresů

Pro zobrazení variability objektů použijeme ikonové grafy, jedná se o obličejový nebo hvězdicový graf. Zde si ukážeme pouze obličejový (Obrázek 6.2). Z těchto grafů můžeme získat pohled na podobnost objektů, nalézt určité shluky, aj. Ve druhé fázi provedeme



Obrázek 6.2.: Ikonový graf datové matice okresů

samotnou metodu PCA. Nejprve vypočítáme průměry datové matice a pomocí nich se provedeme centrování. Centrovaná data upravíme podle vzorce

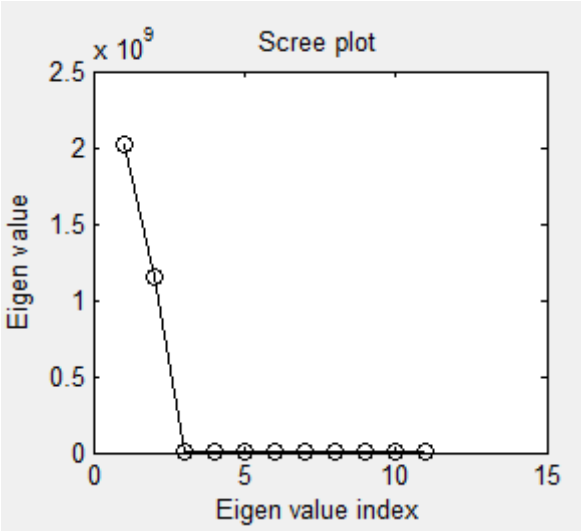
$$\frac{X_C}{\sqrt{n-1}}$$

a použijeme metodu pro SVD rozklad.

```
[n, ~] = size(X);
Xc = X - repmat(mean(X),n,1);
[~,D,V] = svd(Xc/sqrt(n-1));
```

Poznámka: Pokud nechceme uložit výstupní parametr funkce do proměnné použijeme zástupný symbol ∼.

Pomocí matice D se vytvoří graf úpatí vlastních čísel (Obrázek 6.3). Z toho je vidět, že pro získání největšího rozptylu postačí už první 3 komponenty. To je vhodné pro zobrazení zdrojových dat do 3D grafu.



Obrázek 6.3.: Graf úpatí vlastních čísel datové matice okresů

Z matice V se vyčíslí pouze 2 až 3 proměnné a pomocí centrované vstupní matice se dopočte komponentní skóre. Dále se mohou spočítat objektová rezidua a zjistit, jak přesný model je.

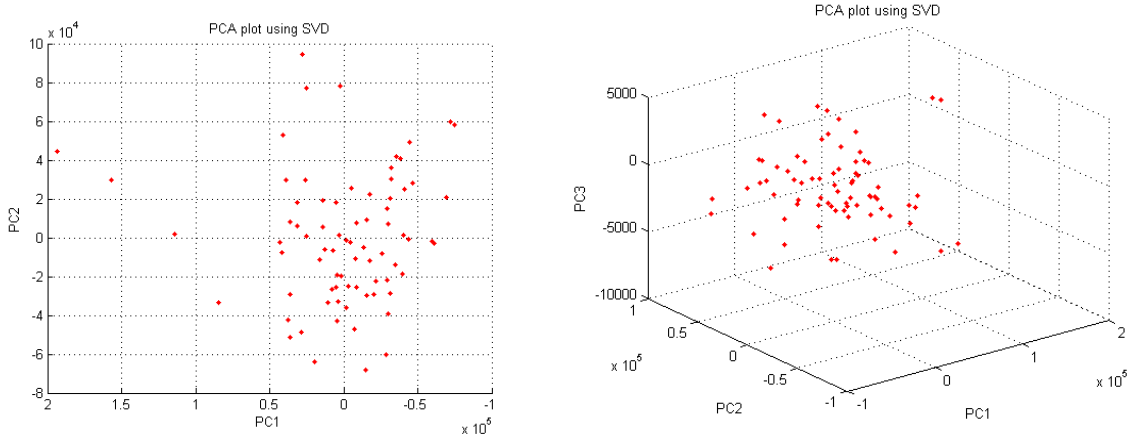
```
coef = V(:,1:pcs);
score = Xc*coef;
residue = Xc - score*coef';
```

Z matice hlavních komponent (Tabulka 6.3), lze zjistit, jak silně která proměnná přispívá do dané komponenty. Tato matice se může dále vizualizovat pomocí grafu komponentních zátěží.

Proměnná/komponenta	PC1	PC2	PC3
Cizinci	0,060	0,012	-0.982
Trestné činy	0.040	0.018	-0.166
Počet můžu	0.544	0.413	0.068
Počet žen	0.583	0.436	0.021
Věk muži	$-3.165 \cdot 10^6$	$9.4717 \cdot 10^7$	$4.6232 \cdot 10^5$
Věk ženy	$2.3498 \cdot 10^7$	$1.6197 \cdot 10^6$	$1,152 \cdot 10^4$
Sňatky	$4,069 \cdot 10^4$	$2,674 \cdot 10^4$	$-5,854 \cdot 10^4$
Důchod ženy	$4,770 \cdot 10^3$	$-3,973 \cdot 10^4$	-0.017
Důchod muži	$1,529 \cdot 10^3$	$-4,378 \cdot 10^4$	-0,037
Plocha území	-0.6	0.8	-0.028
Ekonomická stabilita	$-4.539 \cdot 10^6$	$-1.223 \cdot 10^6$	$-1.758 \cdot 10^5$

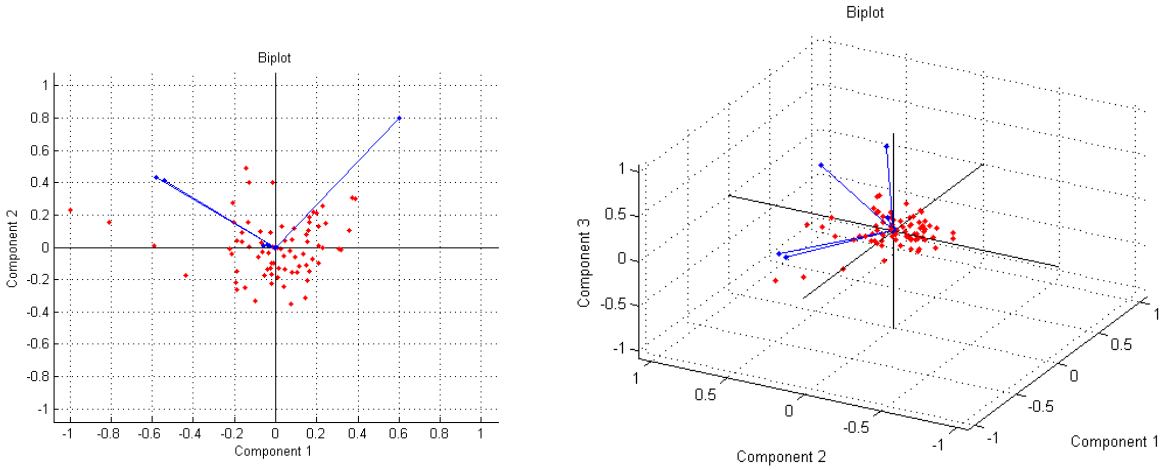
Tabulka 6.3.: Komponentní zátěže

Cílem metody bylo také snížení dimenze vstupních dat. Vykreslením grafu komponentního skóre pro 2 nebo 3 proměnné, můžeme snadno vizualizovat objekty původního souboru (Obrázek 6.4). Lze zde určit odlehlé body. Tyto extrémy pak udávají směr dané komponenty.



Obrázek 6.4.: Rozptylový graf matice okresů

Spojením grafu komponentních zátěží a komponentního skóre do dvojného grafu, můžeme zjistit, který objekt je silně svázán s jakou hlavní komponentou (Obrázek 6.5).



Obrázek 6.5.: Vizualizace dvojného grafu

Tato metoda je implementována v aplikaci *Data Analysis Application* (příloha A). Ta celý proces zastřešuje a poskytuje grafické uživatelské rozhraní pro snadnější použití.

Příklad 6.1.2: Komprese dat

Každá úloha začíná přípravou vlastních dat. Nejprve se importují grafická data a převedou na dvourozměrná pole. Pokud jsou vstupní data v odstínech šedé, provede se import do jednoho dvourozměrného pole, pokud jsou barevná importují se do třech dvourozměrných polí. Každé pole odpovídá jednomu barevnému kanálu.

```
image = imread(filename);
R = image(:,:,1); G = image(:,:,2); B = image(:,:,3);
```

Hodnoty v těchto polích reprezentují intenzitu dané barevné složky. Tato hodnota může být kódována dvěma způsoby. První způsob je kódování celočíselným datovým typem s rozsahem od 0 do 255, kde 0 znamená žádnou intenzitu a 255 maximální intenzitu. Druhý způsob kódování je pomocí datového typu s plovoucí řádovou čárkou. V takovém případě je rozsah hodnot od 0 do 1, kde 0 znamená žádnou intenzitu a 1 maximální intenzitu. V této práci se bude používat druhý typ. Zdrojová pole je potřeba přetypovat a upravit rozsah hodnot. Přetypování se provádí explicitním napsáním nového datového typu a upravení rozsahu hodnot je pouze aritmetická operace.

```
R = double(R); R = R./255;
G = double(G); G = G./255;
B = double(B); B = B./255;
```

V této fázi se provede samotná komprese pro každý barevný kanál zvlášť. Použití metody Hlavních komponent či SVD rozkladu je však spojeno s další předúpravou a tou je centrování. Výsledná datová matice se upraví podle vzorce

$$\frac{X_C}{\sqrt{n-1}}$$

a použije v metodě pro SVD rozklad.

```
[n,~] = size(X);          % X represents color channel
Xmean = repmat(mean(X),n,1);
Xc = X - Xmean;
Xsvd = Xc/sqrt(n-1);
```

Poznámka: Pokud nechceme uložit výstupní parametr funkce do proměnné použijeme zástupný symbol ∼.

Z diagonální matice vlastních čísel se vykreslí graf úpatí a podle něj se určí počet hlavních komponent. Z matice vlastních vektorů se odebere pouze prvních **k** komponent, odečtených z grafu a vytvoří se redukovaná matice prvních k hlavních komponent. Menší **k** značí vyšší kompresi. Z redukované matice komponent a zdrojové centrované se dopočítá komponentní skóre. Z komponentního skóre a hlavních komponent se dopočte komprimované centrované dvourozměrné pole. K tomuto poli se přičte celkový průměr a vznikne výsledné komprimované pole pro daný barevný kanál.

```
[~,~,V] = svd(Xsvd);
comp = V(:,1:k);
score = Xc*comp;
Xc_compressed = score*comp';
Xcompress = Xc_compressed + Xmean;
```

Po kompresi všech barevných kanálů se vytvoří jedno komprimované trojrozměrné pole, do kterého se složí barevné kanály. Toto pole se uloží zpět do grafického souboru.

```
[n, m, ~] = size(image);
compressedImage = zeros(n, m, 3);
compressedImage(:,:,1) = compress(R, k);
compressedImage(:,:,2) = compress(G, k);
compressedImage(:,:,3) = compress(B, k);
imwrite(compressedImage, filename);
```

Detailnější popis v publikacích [19], [8].

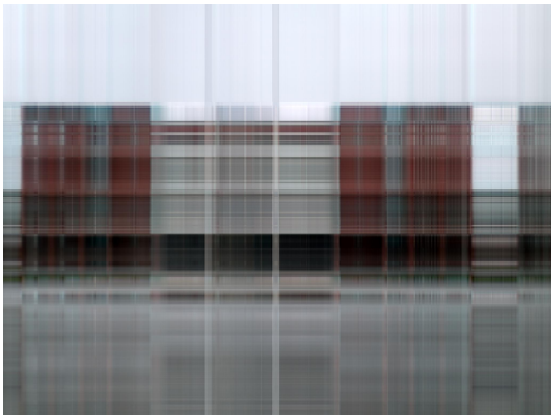
Pro kompresi je použita fotografie Fakulty informatiky a managementu Univerzity Hradec Králové. Tato fotografie je následně porovnána se zkomprimovanými variantami (Tabulka 6.4). Komprese probíhá v barevném formátu.

Obrázek 6.6.: FIM UHK před kompresí

Původní nekomprimovaný obrázek měl velikost 719,264 kB. Po provedení komprese na 200 hlavních komponent, si můžeme povšimnout, že velikost se nám výrazně snížila (na 272,418 kB). Zároveň kvalita obrázku je od originálu pro běžného člověka nerozeznatelná. Jak je vidět z porovnání komprimovaných výstupů, nejlepší poměr kvalita/velikost dává 100 hlavních komponent.

Ve výsledku se tato metoda dá využít tam, kde je potřeba uchovávat velké množství obrázků a kde je potřeba zachovat určitou kvalitu. Použití například metodě rozpoznávání obličejů, analýzy lékařských dat, aj.

Tato metoda je implementována v aplikaci *Image Compression Application* (příloha B.). Ta celý proces zastřešuje a poskytuje grafické uživatelské rozhraní pro snadnější použití.



3 komponenty - 171,285 kB

10 komponent - 212,540 kB

25 komponent - 234,818 kB

50 komponent - 249,328 kB

100 komponent - 263,978 kB

200 komponent - 272,418 kB

Tabulka 6.4.: Porovnání komprese metodou PCA

6.2. Faktorová analýza v prostředí MATLAB

Metoda faktorové analýzy se dá použít pro nalezení vnitřních vztahů mezi proměnnými i jako prostředek pro redukci dimenze dat.

Obecný postup použití metody

Máme datový soubor obsahující jednotlivá pozorování (objekty), které jsou popsány několika atributy.

Naším cílem je nalézt v pozadí stojící společné faktory. Na základě kapitoly 4.4 provedeme nejprve normování zdrojových dat

$$Z = \frac{X - \overline{x}}{s(x)}.$$

Potom provedeme rozklad kovarianční matice zdrojového souboru

$$\Sigma = \Gamma\Gamma^T + \Psi.$$

Tím získáme matici faktorových zátěží Γ. Pomocí regrese následně odhadneme faktorové skóre F. To reprezentuje v pozadí stojící společné faktory

$$Z = \Gamma \cdot F.$$

Pokud máme 2 - 3 společné faktory, můžeme pomocí nich zobrazit původní objekty ve 2D, nebo 3D grafu.

Samotné analýze může předcházet klasická průzkumová analýza. Tedy výpočet deskriptivních statistik, průzkum normality, aj. Podrobněji v kapitole 5.

Příklad 6.2.1: Sociální data

Pro analýzu je vybrán datový soubor popisující úspěšnost dvojčat ve vědomostních testech a závislost na sociálních faktorech. Data čerpána z [12].

Vstupní matice má *1678* objektů. Každý objekt reprezentuje jednu sledovanou osobu. Ta je charakterizována *10* proměnnými. Jedná se o:

- Pohlaví (1 - muži, 2 - ženy)
- Podobnost (1 - identické, 2 - bratrská dvojčata)
- Stupeň vzdělání matky
- Stupeň vzdělání otce
- Příjem rodiny
- Výsledky testu - Angličtina
- Výsledky testu - Matematika
- Výsledky testu - Sociální vědy
- Výsledky testu - Přírodní vědy
- Výsledky testu - Slovní zásoba

Nejprve je potřeba si připravit datovou matici dvojčat. Tato matice musí obsahovat pouze numerické hodnoty oddělené čárkou, viz Příklad 6.1.1.

V první fázi se načteme datovou matici. Zobrazením matice, zjistíme, že obsahuje chybějící hodnoty. Ty mohou nepříznivě ovlivnit výsledky. Je tedy nutné neúplné záznamy úplně vynechat.

```
remove = any(isnan(data),2);
data(remove, :) = [];
```

Po odstranění chybějících hodnot se už můžeme zabývat základním průzkumem dat. Vypočítáme deskriptivní statistiky a vyšetříme normalitu dat. Vzhledem k velkému počtu objektů, zde nemají smysl ikonové grafy.

Pro základní přehled o datech si vypočítáme deskriptivní statistiky. Jedná se především o aritmetický průměr, 1 až 3. kvartil, minimum, maximum, rozptyl, popř. směrodatná odchylka. Použijeme na to funkce z kapitoly 5.2. Tím můžeme získat určitý pohled na variabilitu proměnných. Výsledek v tabulce 6.5.

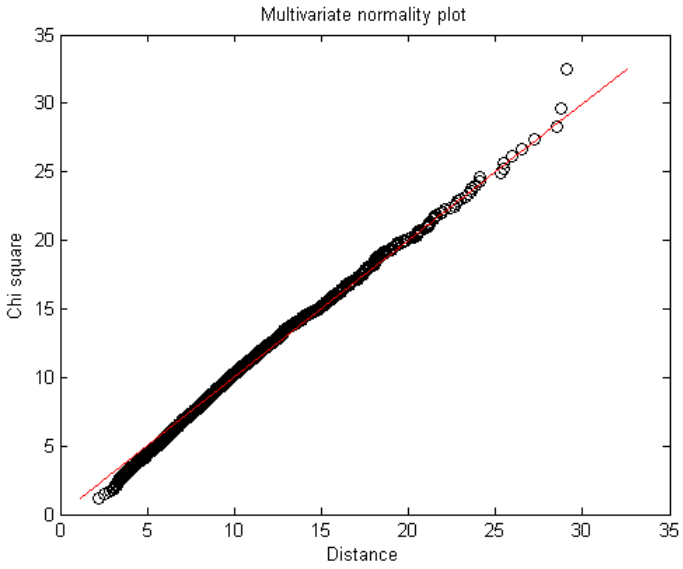
Průměr	Medián	Max	Min	Rozptyl	1. kvartil	3. kvartil
1,57	2	2	1	0,24	1	2
1,39	1	2	1	0,24	1	2
3,42	3	6	1	1,45	3	4
3,59	4	6	1	2,30	3	5
3,24	3	7	1	2,38	2	4
19,76	20	31	3	21,75	17	23
21,25	21	35	3	39,72	17	26
20,69	21	32	3	23,78	17	24
20,09	21	33	3	32,11	16	24
21,06	21	32	4	22,49	18	24

Tabulka 6.5.: Popisné statistiky proměnných matice dvojčat
(Proměnné jsou po řádcích: Pohlaví, Podobnost, Stupeň vzdělání matky, Stupeň vzdělání otce, Příjem rodiny, Výsledky testu - Angličtina, Výsledky testu - Matematika, Výsledky testu - Sociální vědy, Výsledky testu - Přírodní vědy, Výsledky testu - Slovní zásoba)

Pro testování vícerozměrné šikmosti a špičatosti jsme postupovali podobně jako v Příklad 6.1.1. Využili jsme funkce `NormalityTests()` a `MultivariateQQPlot()`. Výsledek testu šikmosti i špičatosti můžeme zamítnout. Tedy data nepochází z vícerozměrného normálního rozdělení. Avšak pokud si prohlédneme graf (6.1) zjistíme, že odchylky od normality nejsou tak významné. Po odstranění extrémních hodnot, by se dalo rozdělení považovat za normální.

Veličina	Statistika	Testové kritérium	p hodnota	Přesnost
Šikmost	789.357	255.602	0	0,05
Špičatost	-3.257	-1,65	0,00056	0,05

Tabulka 6.6.: Test vícerozměrné šikmosti a špičatosti matice dvojčat



Obrázek 6.7.: Q-Q graf vícerozměrné normality matice dvojčat

Ve druhé fázi provedeme samotnou metodu faktorové analýzy. Na rozdíl od metody hlavních komponent, je nutné specifikovat počet proměnných hned na začátku analýzy.

V MATLABu pro výpočet slouží funkce `factoran()`. Vstupem je počet cílových proměnných a datový soubor. Dále lze specifikovat metodu pro odhad faktorového skóre a rotaci. Výstupem funkce je matice faktorových zátěží, reziduální složka, rotační matice, statistický test a faktorové skóre. Statistickým testem se zde zjišťuje, zda zvolený počet faktorů je dostatečný pro reprezentaci hodnot.

V tomto ukázkovém příkladě, je nastaven počet 3 faktorů, rotace žádná a pro odhad faktorového skóre použita regrese (kromě regresní metody, lze nastavit metoda vážených nejmenších čtverců). Výstupem funkce je matice faktorových zátěží Γ a matice faktorového skóre F.

```
[gamma,~,~,~, F] = factoran(data, 3, 'xType', 'data', '
   rotate', 'none', 'scores', 'regression');
```

Poznámka: Pokud nechceme uložit výstupní parametr funkce do proměnné použijeme zástupný symbol $\sim$.

Z matice faktorových zátěží (Tabulka 6.7), lze zjistit, jak silně která proměnná přispívá do daného faktoru.

Proměnná/faktor	Faktor 1	Faktor 2	Faktor 3
Pohlaví	-0,118	0,160	0,719
Stupeň podobnosti	0,029	0,056	-0,001
Vzdělání matky	0,324	0,567	-0,076
Vzdělání otce	0,434	0,724	-0,072
Příjem rodiny	0,347	0,515	-0,005
Test angličtina	0,747	-0,106	0,258
Test matematika	0,735	-0,096	-0,250
Test sociální vědy	0,852	-0,176	0,034
Test přírodní vědy	0,764	-0,201	-0,155
Test slovní zásoba	0,841	-0,074	0,159

Tabulka 6.7.: Nerotované faktorové zátěže matice dvojčat

Dále lze určit, jestli nalezené řešení je smysluplné. Zde vidíme, že jsou některé koeficienty sporné. Použijeme tedy Varimax rotaci pro zjednodušení struktury.

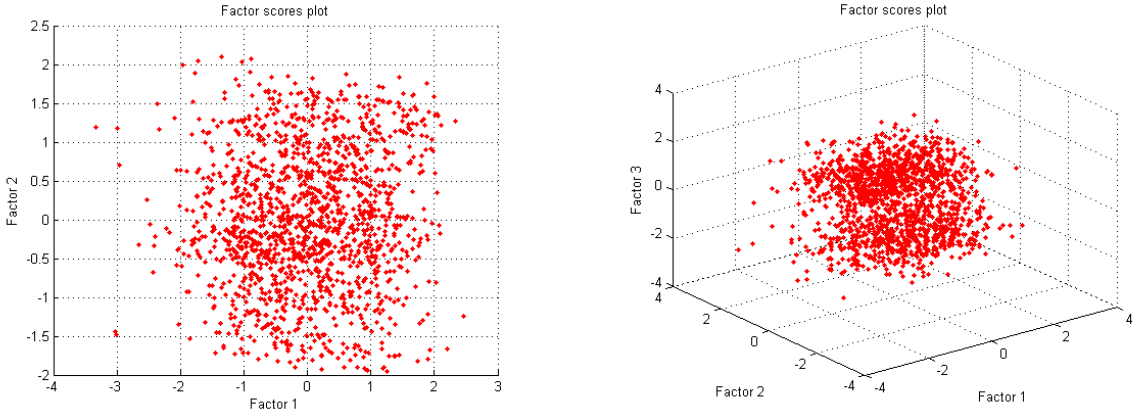
Proměnná/faktor	Faktor 1	Faktor 2	Faktor 3
Pohlaví	-0,070	0,041	0,742
Stupeň podobnosti	0,008	0,062	0,004
Vzdělání matky	0,103	0,648	-0,030
Vzdělání otce	0,154	0,833	-0,015
Příjem rodiny	0,151	0,602	0,032
Test angličtina	0,767	0,137	0,173
Test matematika	0,686	0,189	-0,327
Test sociální vědy	0,858	0,129	-0,067
Test přírodní vědy	0,760	0,092	-0,249
Test slovní zásoba	0.830	0,208	0,071

Tabulka 6.8.: Faktorové zátěže matice dvojčat s rotací Varimax

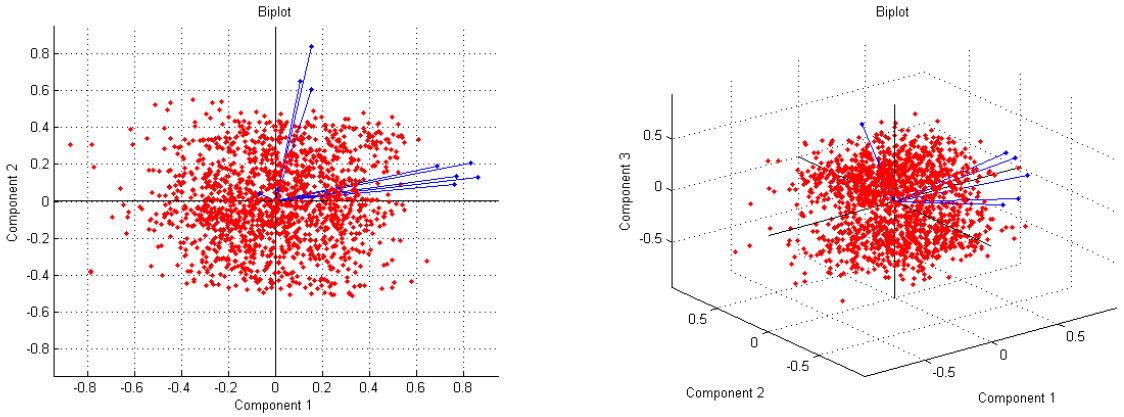
Na základě výsledku analýzy (Tabulka 6.8), lze určit, že **Faktorem 1** jsou vysvětleny proměnné zabývající se hodnocením testů (Test angličtina, matematika, sociální vědy, přírodní vědy, slovní zásoba). **Faktorem 2** jsou vysvětleny proměnné zabývající se poměrem v rodině (Vzdělání otce, matky, příjem rodiny). **Faktorem 3** je vysvětleno pohlaví. Stupeň podobnosti dvojčat je svojí velikostí nejvíce svázán s faktorem 1, avšak

tato proměnná je velmi slabě korelovaná. Tedy na hodnoty z původní datové matice nejsou touto proměnnou výrazně ovlivněny.

Cílem metody bylo také snížení dimenze vstupních dat. Vykreslením grafu komponentního skóre pro 2 nebo 3 proměnné, můžeme snadno vizualizovat data původního souboru (Obrázek 6.8). Spojením grafu faktorových zátěží a faktorového skóre do dvojného grafu, můžeme zjistit, který objekt je silně svázán s jakým faktorem. (Obrázek 6.9).



Obrázek 6.8.: Vizualizace matice dvojčat



Obrázek 6.9.: Vizualizace dvojného grafu

Tato metoda je implementována v aplikaci *Data Analysis Application* (příloha A). Ta celý proces zastřešuje a poskytuje grafické uživatelské rozhraní pro snadnější použití.

6.3. Shluková analýza v prostředí MATLAB

Používá se pro určení struktury ve znacích, tedy pro rozdělení objektů do kategorií. Shluková analýza se dá využít v mnoha odvětvích. Zde si popíšeme oblast sociálně ekonomickou, kde se bude zjišťovat podobnost krajů na základě jednotlivých znaků. Dále má využití v data miningových metodách, konkrétně bude ukázána aplikace v oblasti text miningu a klasifikace dokumentů.

Obecný postup použití metody

Máme datový soubor obsahující jednotlivá pozorování (objekty), které jsou popsány několika atributy.

Speciální formou datové matice je varianta pro text mining. Ta obsahuje vybrané dokumenty (objekty), které jsou popsány významovými slovy (atributy).

Na základě kapitoly 4.5 víme, že je potřeba zajistit splnění předpokladů. Pro zjištění vztahu mezi atributy (**multikolinearity**) je potřeba vypočítat determinant matice $X^T X$. Tato matice se však dá nahradit normovanou korelační maticí R

$$det(R) = \prod_{i=1}^{m} \lambda_i.$$

Pokud je determinant menší než 10^{-3}, poté se jedná o **silnou multikolinearitu** a je potřeba jí odstranit. K tomu lze využít metodu hlavních komponent (kapitola 4.3).

Po splnění předpokladů, zvolíme počet skupin (shluků) do kterých chceme objekty roztřídit. Samotné roztřídění provedeme hierarchickým nebo nehierarchickým způsobem. Více v kapitole 4.5.

Po zjištění shluků je lze vizualizovat ve formě grafu, nebo tabulky. V případě hierarchického shlukování lze využít dendrogram.

Příklad 6.3.1: Demografická data

Shluková analýza využívá data z příkladuPříklad 6.1.1. Jedná se o soubor popisující vliv sociálně ekonomických ukazatelů pro porovnání jednotlivých okresů v České republice (zdroj [3]). Matice obsahuje 76 objektů definovaných 11 atributy.

Základní průzkumová analýza je k dispozici v Příklad 6.1.1, takže tento krok zde vynecháme.

Cílem shlukové analýzy je rozdělit těchto 76 objektů do 3 skupin. V každé skupině budou objekty na které mají atributy podobný vliv.

Pro shlukování použijeme nehierarchickou metodu K-means. Výsledky metody vizualizujeme pomocí grafu a tabulky. Hierarchické shlukování zde využívat nebudeme, protože dendrogram, kterým se postup vizualizuje, se stává pro 76 objektů hůře čitatelným.

V první fázi je potřeba zajistit splnění předpokladů, tedy zjistit možnou přítomnost multikolinearity. Vypočítáme determinant korelační matice. V MATLABu vypočítáme korelační matici pomocí funkce `corr()` a determinant následně funkcí `det()`.

```
R = corr(data);
detR = det(R);
```

Z výsledné hodnoty determinantu lze usoudit, že soubor obsahuje multikolinearitu.

$$\det(R) = 2,237 \cdot 10^{-8} < 10^{-3}.$$

Pro odstranění multikolinearity, je vhodné použít metodu na snížení rozměru a následně pracovat s takto upravenými daty. Pro redukci nastavíme 3 hlavní komponenty. Dále je vhodné provést normování (z-score).

Pro shlukování metodou K-means je nastavena čtvercová eukleidovská vzdálenost, kterou metoda minimalizuje. Dále nastavíme zobrazení výstupů. Necháme si vypsat datový soubor doplněný o indikátor shluku, souřadnice centroidů. Dále si necháme vykreslit bodový graf s rozlišením jednotlivých shluků. Shlukovat budeme do tří skupin.

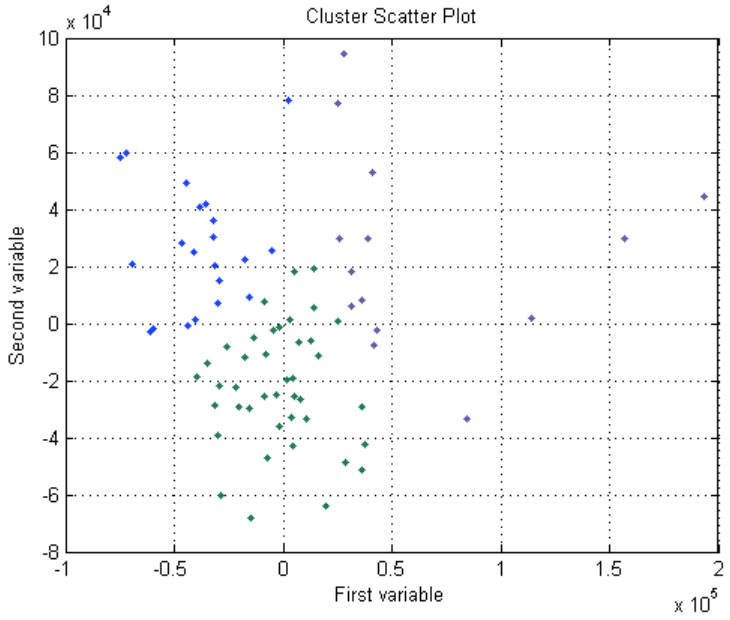
```
data = zscore(data);
[idx, centroids] = kmeans(data, 3,'Distance','sqeuclidean');
```

Princip metody spočívá ve stanovení počátečních centroidů, porovnávání vzdáleností jednotlivých objektů od centroidů a přiřazení do patřičného shluku. Z těchto shluků se vyberou nové centroidy a postup se opakuje. Po provedení dostatečného množství iterací se vytvořila nová skupina finálních centroidů (Tabulka 6.9), podle kterých jsou už prvky klasifikovány s dostatečnou přesností.

Centroid/Proměnná	PC1	PC2	PC3
První	-1783,1	-22276	309,76
Druhý	63658	24977	61,487
Třetí	-38958	26839	-645,76

Tabulka 6.9.: Centroidy shlukování k-means matice okresů

Pro zobrazení v grafu, je potřeba nejprve snížit rozměr dat. Vzhledem k použití redukce dimenze pro zajištění multikolinearity máme rozměr dat snížený a lze rovnou vykreslit graf. Pokud bychom redukci dimenze neprováděli na začátku, bylo by vhodné ji použít v této fázi. Výsledek na obrázku 6.10



Obrázek 6.10.: Vizualizace shluků matice okresů

Další možností reprezentace shluků je ve formě tabulky, kde každému objektu bude přiřazeno číslo shluku. Takto upravená tabulka je zobrazena níže (Tabulka 6.10). V ní se nacházejí přesné názvy okresů, které jsou podle vlastností (proměnných), nejpodobnější.

Tato metoda je implementována v aplikaci *Data Analysis Application* (příloha A). Ta celý proces zastřešuje a poskytuje grafické uživatelské rozhraní pro snadnější použití.

1. shluk		2. shluk	3. shluk
Beroun	Ústí nad Labem	Kladno	Benešov
Kolín	Česká Lípa	Praha-východ	Příbram
Kutná Hora	Jablonec nad Nisou	Plzeň-město	České Budějovice
Mělník	Semily	Liberec	Český Krumlov
Mladá Boleslav	Jičín	Hradec Králové	Jindřichův Hradec
Nymburk	Náchod	Pardubice	Prachatice
Praha-západ	Trutnov	Brno-město	Tábor
Rakovník	Rychnov nad Kněžnou	Brno-venkov	Klatovy
Písek	Chrudim	Olomouc	Plzeň-sever
Strakonice	Blansko	Zlín	Tachov
Domažlice	Břeclav	Frýdek-Místek	Karlovy Vary
Plzeň-jih	Hodonín	Karviná	Svitavy
Rokycany	Vyškov	Opava	Ústí nad Orlicí
Cheb	Jeseník	Ostrava-město	Havlíčkův Brod
Sokolov	Prostějov		Jihlava
Děčín	Přerov		Pelhřimov
Chomutov	Kroměříž		Třebíč
Litoměřice	Uherské Hradiště		Žďár nad Sázavou
Louny	Vsetín		Znojmo
Most	Nový Jičín		Šumperk
Teplice			Bruntál

Tabulka 6.10.: Shluky matice okresů

Příklad 6.3.2: Text mining a klasifikace dokumentů

Pro použití shlukové analýzy v text miningu je potřeba nejprve zpracovat databázi dokumentů. U všech dokumentů je nutné provést předúpravu. Vstupními dokumenty jsou články zabývající se metodou PCA nebo rozpoznáváním obrazu. Konkrétně se jedná o články [2], [10], [1], [23], [21]. Výstupem předúpravy je matice obsahující m dokumentů (objektů) a n klíčových slov. Hodnota následně reprezentuje poměr výskytu daného klíčového slova v patřičném dokumentu.

Předúprava dat spočívá v načtení všech dokumentů v textovém formátu. Dále se každý dokument rozseká na jednotlivé řetězce. Z těchto řetězců se odstraní přípony, předpony,

množné tvary a vezmou se pouze kořeny slov. Z nich se sestaví databáze klíčových slov a odstraní se duplicitní záznamy.

```
str = regexprep(token,'[\.\,]','');
str = regexprep(str,'ees\s','ee ');
str = regexprep(str,'s\s',' ');
result = regexpi(str, '[A-Za-z]+', 'match');
result = lower(result);
...
termDB = unique(termDB);
```

Dále se z databáze klíčových slov odstraní tzv. *stop words*, neboli terminální slova. Výsledkem je databáze slov s informačním významem. Z těch se spočítá tabulka četností pro jednotlivé dokumenty, tzv. **Frequency Document-Term Matrix**. Tato matice slouží jako zdrojová pro další výpočty.

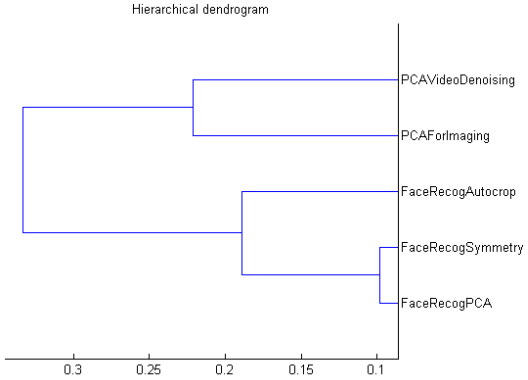
Pro hierarchické shlukování je potřeba spočítat vzdálenost mezi dvojicemi objektů. Jako metrika se použije kosinova vzdálenost (viz kapitola 4.5). Ta je potřeba definovat ručně, jinak počítá výchozí - eukleidovská vzdálenost. Na základě vzdáleností se provede propojení objektů. Jako metoda propojení může být použit nejbližší/nejvzdálenější soused, Wardova metoda, a další. Z této informace lze jednoduše vykreslit dendrogram. Volitelně lze také spočítat indexy shluků.

```
distObj = pdist(data, 'cosine');
linkObj = linkage(distObj, link);
idx = cluster(linkObj, 'maxclust', 2);
dendrogram(linkObj, 'Labels', labels, 'Orientation','left');
```

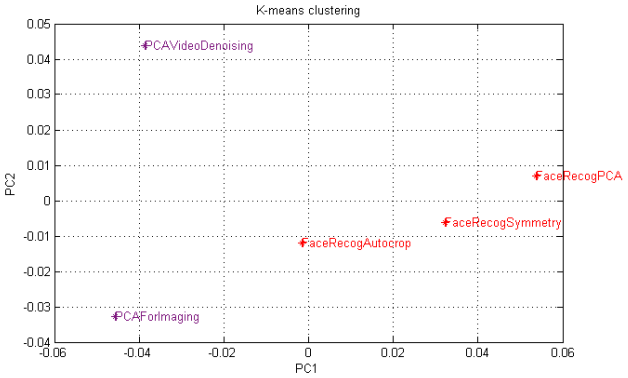
Pro nehierarchické shlukování a metodu K-Means se využije metoda `kmeans()`. U té je nutno specifikovat vstupní data, počet shluků a metriku nastavit na kosinovu. Výsledkem je vektor indexů, určující zařazení jednotlivého objektu do patřičného shluku. Textový výpis jednotlivých skupin nepotřebuje další úpravy. Pro grafické zobrazení však ještě nutné provést redukci dimenze na 2 nebo 3 komponenty, pro přesnější vizualizaci. Využije se k tomu již popsaná metoda PCA. Výsledné komponenty se postupně zobrazí do grafu, kde budou jednotlivé skupiny barevně rozlišeny.

```
idx = kmeans(data, 2, 'Distance', 'cosine');
```

Výsledkem hierarchického shlukování je dendrogram (Obrázek 6.11) zobrazující podobnost jednotlivých textových dokumentů. Na tomto základě lze rozhodnout, že klasifikaci dokumentů do dvou shluků provedl správně. V jednom shluku jsou dokumenty zabývající se primárně metodou PCA a v druhém shluku jsou dokumenty zabývající se primárně rozpoznáváním obličeje. Tuto skutečnost popisuje i scatter plot (Obrázek 6.12) zobrazující výsledek nehierarchického shlukování.



Obrázek 6.11.: Dendrogram



Obrázek 6.12.: K-means graf

Tato metoda je implementována v aplikaci *Document Clustering Application* (příloha C). Ta celý proces zastřešuje a poskytuje grafické uživatelské rozhraní pro snadnější použití.

7. Shrnutí výsledků

V rámci práce se povedlo vytvořit tři nezávislé aplikace v prostředí MATLAB. Všechny aplikace jsou vytvořeny s využitím objektově orientovaných principů.

První z těchto aplikací je *Data Analysis Application*. Jedná se o komplexní aplikaci pro průzkumovou analýzu vícerozměrných dat. V rámci této aplikace jsou implementovány všechny tři vybrané metody vícerozměrných dat (tj. Analýza hlavních komponent, Faktorová analýza, Shluková analýza). Navíc, nad rámec této práce, se povedlo implementovat základní deskriptivní statistiky, test vícerozměrné normality dat a nástroj pro předúpravu vstupních dat (odstranění chybějících hodnot, normování). Aplikace dále umožňuje export grafů a tabulek. Nevýhodou je absence dalších statistických metod, které navíc nebyly ani součástí této práce, a nemožnost plně využít potenciál MATLABovských příkazů.

Druhou aplikací je *Image Compression Application*. Jedná se o aplikaci pro kompresi obrazových dat. Ta je založena na metodě hlavních komponent. Aplikace umožňuje kompresi barevného i černobílého obrázku. Součástí je i náhled komprimovaného obrázku a jeho export. Nevýhodou je absence dalších komprimačních metod, které však nebyly obsahem této práce.

Třetí aplikací je *Document Clustering Application*. Jedná se o aplikaci pro kategorizaci dokumentů do skupin, na základě jejich obsahu. Využívá se zde metody shlukové analýzy a tvoří součást oblasti zvané Text mining. Aplikace umožňuje načíst adresář s dokumenty a na základě extrakce významových slov s využitím metod shlukové analýzy je roztřídit do předem stanovených skupin. Tabulky i grafy lze exportovat. Nevýhodou aplikace je, že zabírá jen malou část oblasti Text miningu.

Výsledky této práce přinesly pokročilou znalost programování v prostředí MATLAB.

Jedná se především o vývoj objektově orientovaných komponent, vývoj grafického rozhraní a správné použití již implementovaných funkcí a interpretace jejich výsledků. Výsledky dále přinesly základní znalost oblasti vícerozměrných dat a různé možnosti jejich využití. Konkrétně princip a využití metod pro redukci rozměru dat (Analýza hlavních komponent, Faktorová analýza), hledání struktur a shluků v datech (Shluková analýza).

8. Závěry a doporučení

Po zhodnocení vlastního řešení (třech vytvořených aplikací) z hlediska rozsahu zkoumané oblasti, dodatečných funkcí a možnosti ovládání jsem dospěl z následujícím závěrům.

Aplikace *Data Analysis Application* poskytuje základní možnosti úpravy dat, vizualizace dat a export výsledků obdobný jako profesionální software (např. *IBM SPSS Statistics*). V porovnání se softwary jako R jazyk, nabízí MATLAB srovnatelně výkonné funkce. Tato implementovaná aplikace však nedokáže využít všechny dostupné parametry a výstupy daných funkcí. Do budoucna by bylo vhodné aplikaci rozšířit o další metody vícerozměrné analýzy, jako o Diskriminační analýzu, Logistickou regresi, multidimenzionální škálování, aj.

Aplikace *Image Compression Application* poskytuje základní kompresi pomocí metody hlavních komponent. Zde by do budoucna bylo vhodné aplikaci rozšířit o další kompresní metody.

Aplikace *Document Clustering Application* poskytuje dobrý nástroj pro shlukování dokumentů, avšak z hlediska oblasti Text miningu, zabírá jen malou podmnožinu. Do budoucna by bylo vhodné zamyslet se nad dalšími možnostmi analýzy textu v prostředí MATLAB a případné rozšíření této aplikace.

Na závěr bych doporučil využití obou typů softwaru. Pokud jako uživatel potřebujeme základní práci s těmito metodami, poté bych doporučil grafickou variantu *SPSS Statistics*, nebo *Data Analysis Application*. Tím získáme rychle základní charakteristiky. Pokud jako uživatel potřebujeme speciální nastavení metod, nebo speciální výstupy, pak bych doporučil využít jazyk R, nebo samotný MATLAB.

Seznam použité literatury

[1] BARAHATE, S. R.; SATURWAR, J. *Face recognition using PCA based algorithm and neural network.* In: Proceedings of the International Conference and Workshop on Emerging Trends in Technology. ACM, 2010. p. 249-252.

[2] BHUJLE, H.; CHAUDHURI, S. *PCA based video denoising in a non-local means framework.* In: Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing. ACM, 2012. p. 64.

[3] *Český statistický úřad, Veřejná databáze* [online]. [cit. 2016-09-26]. Dostupné z: `https://vdb.czso.cz/vdbvo2/faces/cs/index.jsf`

[4] HEBÁK, P. *Vícerozměrné statistické metody [1].* Praha: Informatorium, 2007. ISBN 978-80-7333-056-9

[5] HEBÁK, P. *Vícerozměrné statistické metody [3].* Praha: Informatorium, 2005. ISBN 80-7333-039-3

[6] HENDL, J. *Přehled statistických metod a zpracování dat.* Portál Praha, 2004. ISBN 80-7178-820-1

[7] HANSELMAN, D. C.; LITTLEFIELD, B. *Mastering MATLAB 7.* Upper Saddle River, NJ: Pearson/Prentice Hall, c2005. ISBN 0-13-143018-1

[8] HLADNIK, A. *Image compression and face recognition: Two image processing applications of principal component analysis.* International Circular of Graphic Education and Research, 2013, 6: 56-61.

[9] HÖPPNER, F. *Fuzzy cluster analysis: methods for classification, data analysis, and image recognition.* Chichester; New York: J. Wiley, c1999. ISBN 0-471-98864-2

[10] KARMAKAR, D.; MURTHY, C. A. *Face Recognition using Face-Autocropping and Facial Feature Points Extraction.* In: Proceedings of the 2nd International Conference on Perception and Machine Intelligence. ACM, 2015. p. 116-122.

[11] KONCHADY, M. *Text mining application programming.* Boston, Mass.: Charles River Media, 2006. Charles River Media programming series. ISBN 1-58450-460-9

[12] LOEHLIN, J. C.; R. C. NICHOLS. *The National Merit Twin Study* [online]. Harvard Dataverse, 2013 [cit. 2016-12-18]. DOI: 1902.1/13913. Dostupné z: `http://hdl.handle.net/1902.1/13913`

[13] MARTINEZ, W. L.; MARTINEZ, A. R. *Exploratory data analysis with MATLAB.* Boca Raton, Fla.: Chapman & Hall/CRC, c2005. ISBN 1-58488-366-9

[14] MELOUN, M.; MILITKÝ, J. *Statistická analýza experimentálních dat.* Praha: Academia, 2004. ISBN 80-200-1254-0

[15] MYATT, G. J. *Making sense of data: A practical guide to exploratory data analysis and data mining.* Hoboken, N.J.: Wiley-Interscience, c2007. ISBN 978-0-470-07471-8

[16] NARAYAN, C. G. *Multivariate statistical analysis.* Marcel Dekker, Inc., 2004. ISBN: 0-8247-4713-5

[17] ROHN, J.*Lineární algebra a optimalizace.* Praha: Karolinum, 2004. ISBN 80-246-0932-0.

[18] ŘEZANKOVÁ, H.; HÚSEK, D.; SNÁŠEL, V. *Shluková analýza dat.* Praha: Professional Publishing, 2007. ISBN 978-80-86946-26-9

[19] SANTO, R. do E. *Principal Component Analysis applied to digital image compression.* Einstein (São Paulo), 2012, 10.2: 135-139.

[20] SHAHLA, R. M.; FAISAL, Z.; SHUMILA, R. *Multivariate Normality: A Graphical Approach.* Middle-East Journal of Scientific Research 13 (2). IDOSI Publications, 2013, 2013, 254-263. DOI: 10.5829/idosi.mejsr.2013.13.2.1746. ISSN 1990-9233.

[21] SINGH, A. K.; NANDI, G. Ch. *Face recognition using facial symmetry.* In: Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology. ACM, 2012. p. 550-554.

[22] SOUKUP T., DAVIDSON I. *Visual data mining: techniques and tools for data visualization and mining.* Wiley Computer Publishing, 2002. ISBN: 0-471-14999-3

[23] STRAUCH, M.; GALIZIA, C. G. *Fast PCA for processing calcium-imaging data from the brain of Drosophila melanogaster.* BMC medical informatics and decision making, 2012, 12.1: 1.

[24] TURZÍK, D. *Matematika III: Základy optimalizace.* 2. vyd. Praha: Vysoká škola chemicko-technologická, 1994. ISBN 8070802057.

Příloha A.

Data Analysis Application - Uživatelská příručka

Požadavky

- MATLAB verze 2014b a vyšší
- Statistics and Machine Learning Toolbox^{TM} pro MATLAB

Instalace a spuštění

Instalace probíhá překopírováním adresáře `Data Analysis Application`, případně lze spustit přímo z CD, bez nutnosti kopírování.

Proces spuštění:

1. spustíme prostředí MATLAB
2. adresář *Data Analysis Application* zvolíme jako pracovní
3. aplikaci spustíme příkazem `run`

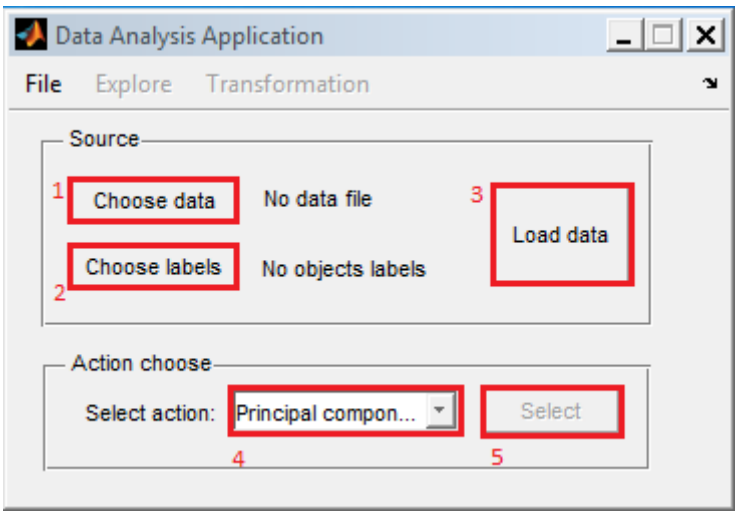
Načtení datového souboru a výběr analýzy

Po spuštění aplikace je nutné vybrat soubor s datovou maticí. Musí se jednat o textový soubor ve formátu ***.csv**, který obsahuje pouze numerické hodnoty oddělené čárkou. Výběr provedeme pomocí tlačítka **Choose data** (na obrázku označeno číslem 1).

Pokud chceme načíst názvy objektů, tedy názvy jednotlivých řádků, použijeme k tomu tlačítko **Choose labels** (na obrázku číslo 2). Zde se musí jednat o soubor obsahující numerické, nebo řetězcové hodnoty oddělené čárkou. Hodnoty musí být v jednom řádku a jejich počet musí korespondovat s počtem řádků datové matice.

Samotný import hodnot provedeme následovně pomocí tlačítka **Load Data** (na obrázku číslo 3).

Po načtení dat máme k dispozici výběr mezi **Analýzou hlavních komponent**, **Faktorovou analýzou** a **Shlukovou analýzou** (kombo box číslo 4). Po výběru jedné možnosti stiskneme tlačítko **Select** (na obrázku číslo 5).

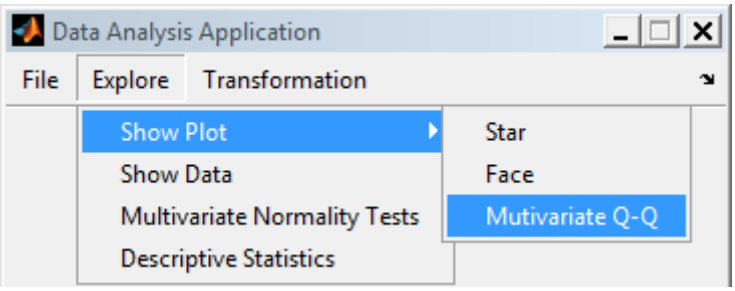


Explorace a transformace dat

Po načtení dat máme k dispozici explorační metody pro prvotní vizualizaci dat a transformační metody.

Explorační metody (menu **Explore**) dělíme na dvě skupiny. První skupinou je grafická vizualizace, druhou skupinou výpočet charakteristik. Samostatně stojí možnost pro zobrazení datové matice, ve formě tabulky (Show Data).

Do první skupiny patří ikonové grafy pro zjištění podobnosti proměnných, extrému atd. (Show Plots $->$ Star, Face). Dále grafické zjištění vícerozměrné normality (Show Plots $->$ Multivariate Q-Q), kde odchylky bodů od diagonály značí porušení normality.



Do druhé skupiny patří výpočet testů vícerozměrné normality (Multivariate Normality Tests). Ten vypočítá vícerozměrné šikmosti a špičatosti. Na jejich základě rozhodne o normalitě dat.

Normality tests

	1	2	3	4	5
1		Statistics	Test kriterium	p-value	Alfa
2	Skewness	1.0339e+03	326.4430	0	0.0500
3	Kurtosis	14.2574	1.6449	0	0.0500

Dále do druhé kategorie patří výpočet základních deskriptivních statistik pro všechny proměnné.

Transformační metody (menu **Transformation**) dovolují uživateli provést transformaci na interval (0,1) nebo převod na z-skóre a potom spustit konkrétní analýzu už s transformovanými hodnotami.

Analýza hlavních komponent

Po načtení dat a zvolení možnosti *Principal Component Analysis* se zobrazí nové okno pro výpočet analýzy hlavních komponent.

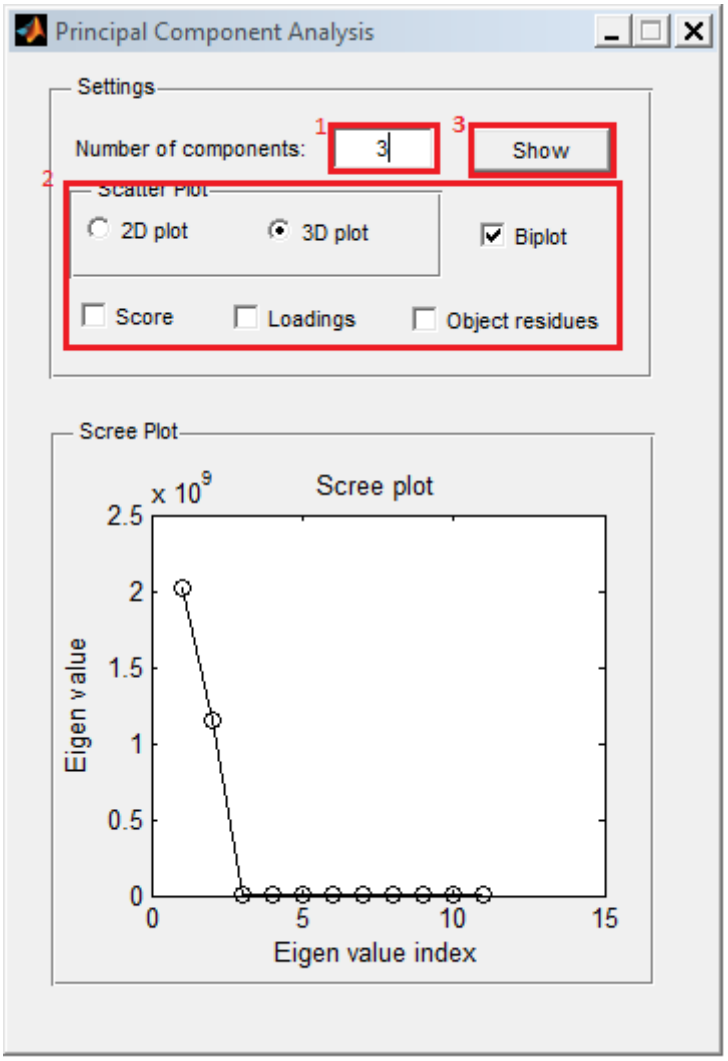
Okno se skládá z dvou částí. První část se zabývá nastavením metody. Druhá část zobrazuje graf úpatí vlastních čísel.

Nejprve se na základě grafu úpatí vlastních čísel zvolí počet hlavních komponent, které chceme vypočítat. To se zadá do patřičného pole (na obrázku číslo 1).

Dalším krokem je výběr možností, které chceme z analýzy získat (na obrázku číslo 2). Jedná se o možnost zobrazení dvojného grafu (Biplot) a bodového grafu komponentních

skóre (Scatter plot). Dále zobrazení hodnot komponentního skóre (Component score) a zátěží (Component loadings) ve formě tabulky. Nakonec možnost spočítat reziduální složku (Object residues)

Pokud je zadán počet hlavních komponent a jsou vybrány možnosti, které chceme spočítat, stačí stisknout tlačítko **Show** (na obrázku číslo 3).



Faktorová analýza

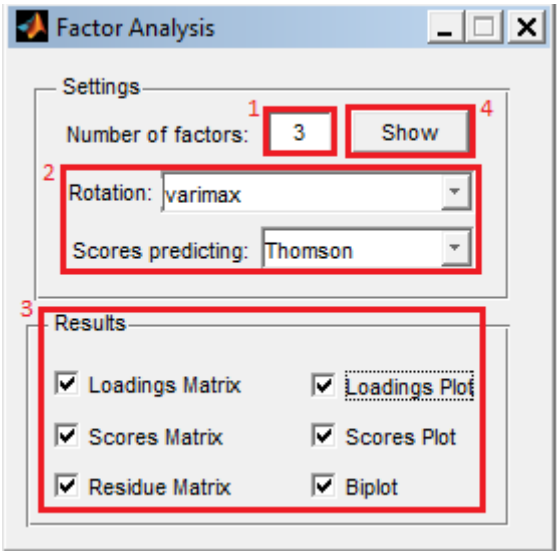
Po načtení dat a zvolení možnosti *Factor Analysis* se zobrazí nové okno pro výpočet faktorové analýzy.

Okno se skládá z dvou částí. První část se zabývá nastavením faktorové analýzy. Ve

druhé části se vybírají charakteristiky, které chceme z metody získat.

Pro výpočet se nejprve zadá počet faktorů (na obrázku číslo 1), dále se specifikuje faktorová rotace a metoda pro výpočet faktorového skóre (na obrázku číslo 2). Následně se vyberou charakteristiky, které chceme získat (na obrázku číslo 3). Jedná se o možnost zobrazení dvojného grafu (Biplot), grafů faktorového skóre (Scores Plot) a zátěží (Loadings Plot). Dále zobrazení hodnot faktorového skóre (Scores Matrix) a zátěží (Loadings Matrix) ve formě tabulky. Nakonec možnost spočítat reziduální složku (Residue Matrix)

Pokud je zadán počet faktorů a jsou vybrány možnosti, které chceme spočítat, stačí stisknout tlačítko **Show** (na obrázku číslo 4).



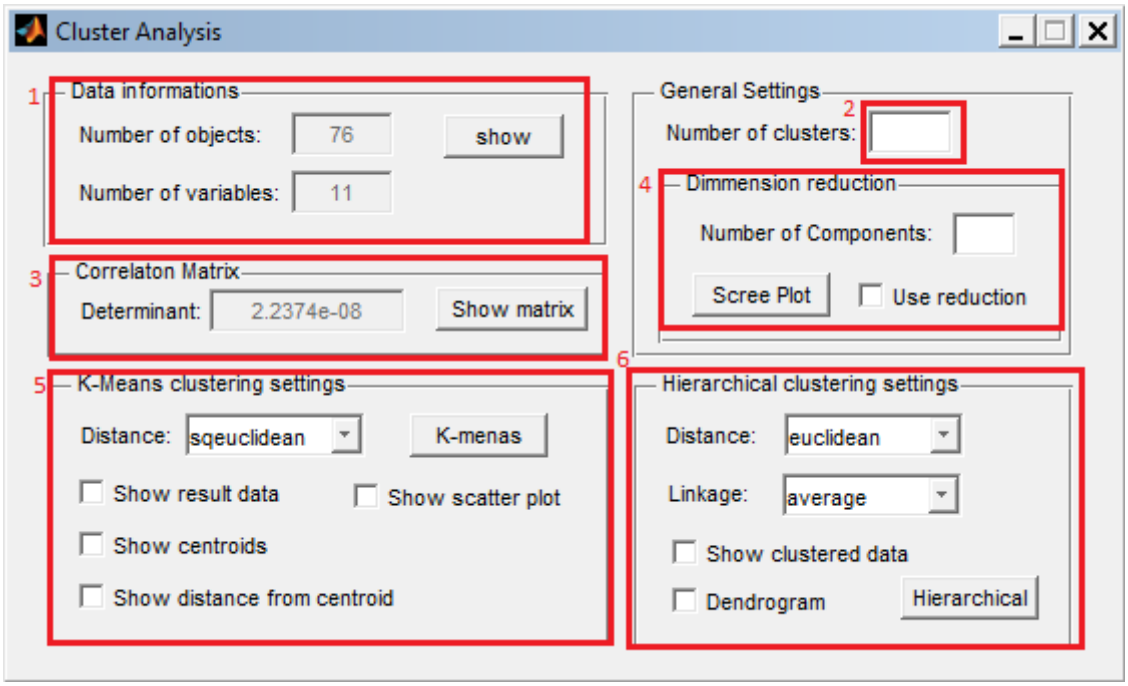
Shluková analýza

Po načtení dat a zvolení možnosti *Cluster Analysis* se zobrazí nové okno pro výpočet shlukové analýzy.

Okno se skládá z pěti částí, které jsou rozdělené podle funkcionality. První část, která je umístěna vlevo nahoře zobrazuje základní informace o datovém souboru s možností zobrazení dat v tabulace (na obrázku číslo 1). Druhá část, umístěná vlevo pod částí první, se zabývá korelační maticí vstupního datového souboru (na obrázku číslo 3). Třetí část, nacházející se vpravo nahoře, se zabývá společným nastavením hierarchického i nehierarchického shlukování (na obrázku číslo 2, 4). Ve spodní části okna se na levé straně nachází nastavení hierarchického shlukování a na pravé nastavení nehierarchického shlu-

kování.

Pro výpočet se nejprve prozkoumají vstupní data (na obrázku číslo 1) a určí se počet shluků (na obrázku číslo 2). V dalším kroku se podle determinantu korelační matice zjistí přítomnost multikoliearity (na obrázku číslo 3). Pokud je multikolinearita přítomna, provede se její odstranění ve formě metody hlavních komponent (na obrázku číslo 4). Dále se nastaví parametry hierarchického shlukování a provede se (na obrázku číslo 5). Nakonec se nastaví parametry nehierarchického shlukování a provede se (na obrázku číslo 6).



Zdrojové kódy

Zdrojové kódy celé aplikace jsou přiloženy na CD.

Příloha B.

Image Compression Application - Uživatelská příručka

Požadavky

- MATLAB verze 2016a a vyšší
- Statistics and Machine Learning Toolbox^{TM} pro MATLAB

Instalace a spuštění

Instalace probíhá překopírováním adresáře `Image Compression Application`, případně lze spustit přímo z CD, bez nutnosti kopírování.

Proces spuštění:

1. spustíme prostředí MATLAB
2. adresář *Image Compression Application* zvolíme jako pracovní
3. aplikaci spustíme příkazem `run`

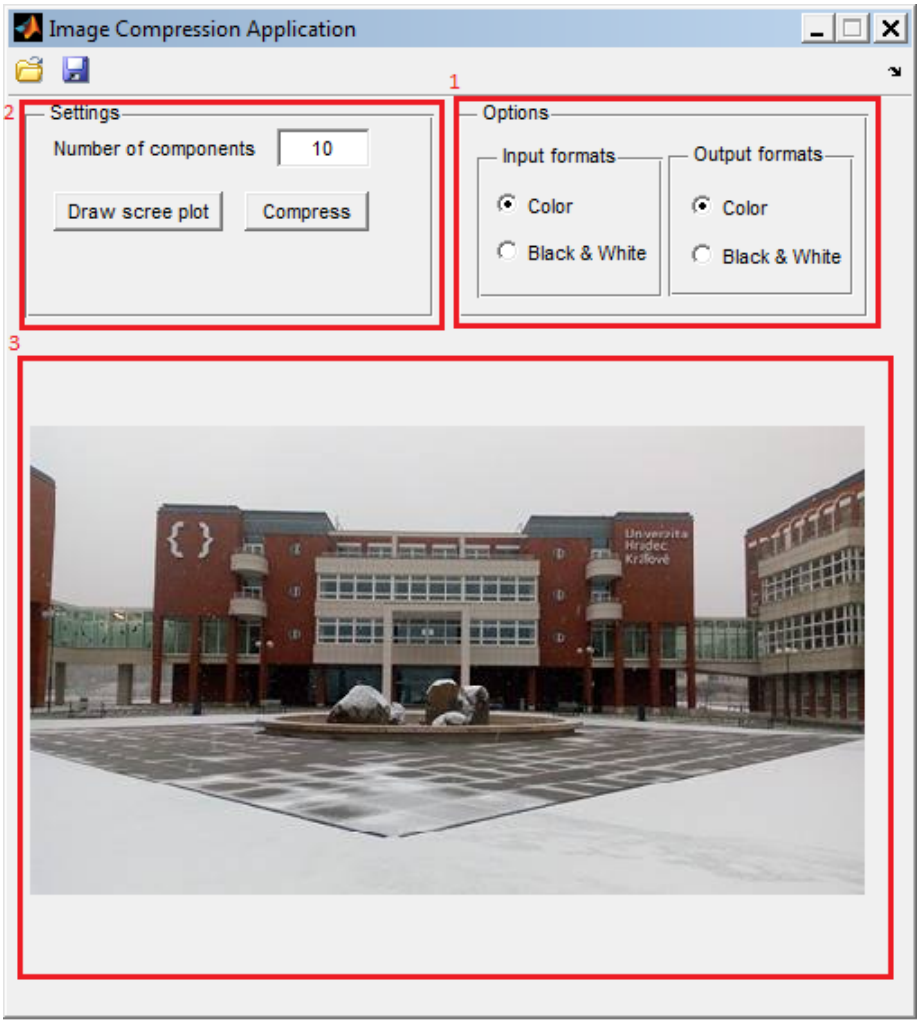
Komprese obrázku

Po spuštění aplikace je nutné vybrat obrázek ke komprimaci. To se provede tlačítkem **Open** nacházející se na horní liště. Po načtení se obrázek zobrazí v náhledové ploše (na obrázku číslo 3).

Dalším krokem je nastavení kompresního formátu. Určení zda se jedná o barevný nebo černobílý vstup a nastavení barevného, nebo černobílého výstupu. (na obrázku číslo 1).

Poznámka: Barevný výstup, funguje pouze v případě barevného vstupního obrázku.

Dále se určí počet komponent, na který se má zdrojový obrázek redukovat. Tlačítkem **Compress** se provede komprese (na obrázku číslo 2). Výsledný obrázek se zobrazí v náhledu (na obrázku číslo 3). Pro uložení komprimovaného obrázku se stiskne tlačítko **Save** nacházejícím se na horní liště.



Zdrojové kódy

Zdrojové kódy celé aplikace jsou přiloženy na CD.

Příloha C.

Document Clustering Application - Uživatelská příručka

Požadavky

- MATLAB verze 2014b a vyšší
- Statistics and Machine Learning Toolbox^{TM} pro MATLAB

Instalace a spuštění

Instalace probíhá překopírováním adresáře `Document Clustering Application`, případně lze spustit přímo z CD, bez nutnosti kopírování.

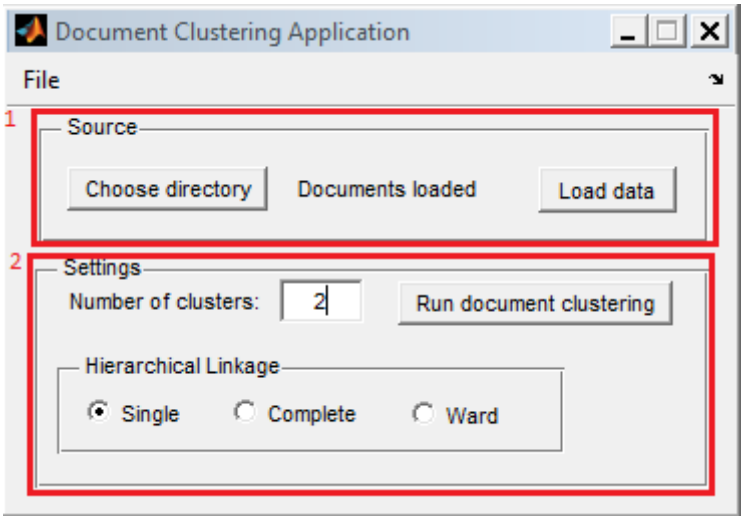
Proces spuštění:

1. spustíme prostředí MATLAB
2. adresář *Document Clustering Application* zvolíme jako pracovní
3. aplikaci spustíme příkazem `run`

Načtení datového souboru a výběr analýzy

Po spuštění aplikace je nutné vybrat adresář s dokumenty pro shlukování (na obrázku označeno číslem 1). Dokumenty musí být ve formátu ***.txt**. Načtení se provede stisknutím tlačítka **Load data**.

Dále je nutné specifikovat počet shluků a parametry hierarchického shlukování (na obrázku označeno číslem 2). Samotný proces shlukování se spustí tlačítkem **Run document clustering**. Výsledkem bude vizualizace grafická i maticová, zvlášť pro hierarchické a zvlášť pro nehierarchické shlukování.



Zdrojové kódy

Zdrojové kódy celé aplikace jsou přiloženy na CD.

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2016/2017

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai2-p)

Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Bc. Vojtas Pavel	Lhotka 188, Česká Třebová - Lhotka	I1500699

TÉMA ČESKY:

Vybrané metody analýzy vícerozměrných dat v MATLABu

TÉMA ANGLICKY:

Selected methods for multivariate data analysis in MATLAB

VEDOUCÍ PRÁCE:

doc. RNDr. Pavel Pražák, Ph.D. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Úvod
Cíl práce a metodika zpracování
Problematika vícerozměrných dat a oblasti využití
Vybrané metody analýzy vícerozměrných dat
  SVD rozklad a metoda hlavních komponent
  Faktorová analýza
  Shluková analýza
Metodologie zpracování dat v prostředí MATLAB
  Průzkumová analýza vícerozměrných dat
  Příprava a standardizace dat
  Interpretace výsledků
Aplikace vybraných metod analýzy vícerozměrných dat
  SVD a metoda hlavních komponent v prostředí MATLAB
  Faktorová analýza v prostředí MATLAB
  Shluková analýza v prostředí MATLAB
Shrnutí výsledků
Závěry a doporučení

SEZNAM DOPORUČENÉ LITERATURY:

1. HEBÁK, P. Vícerozměrné statistické metody [3]. Praha: Informatorium, 2005. ISBN 80-7333-039-3
2. HANSELMAN, D. C. a B. LITTLEFIELD. Mastering MATLAB 7. Upper Saddle River, NJ: Pearson/Prentice Hall, c2005. ISBN 0-13-143018-1
3. LEDOLTER, J. Data mining and business analytics with R. Hoboken, N.J: Wiley, 2013. ISBN 1118572157.
4. MARTINEZ, W. L. a A. R. MARTINEZ. Exploratory data analysis with MATLAB. Boca Raton, Fla.: Chapman & Hall/CRC, c2005. ISBN 1-58488-366-9
5. MELOUN M., J. MILITKÝ. Statistická analýza experimentálních dat. Praha: Academia, 2004. ISBN 80-200-1254-0
6. ŘEZANKOVÁ, H., D. HÚSEK a V. SNÁŠL. Shluková analýza dat. Praha: Professional Publishing, 2007. ISBN 978-80-86946-26-9

Podpis studenta: ..... Datum: 6.10.2016

Podpis vedoucího práce: ..... Datum: 6.10.2016