VŠB TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Shluková analýza: Základní myšlenky a algoritmy

Cluster analysis: Basic Concepts and Algorithms

Roman Macháček

Bakalářská práce

Vedoucí práce: Ing. Martina Litschmannová, Ph.D.

Ostrava, 2021

Abstrakt

Cílem práce je uvést čtenáře do problematiky shlukové analýzy s využitím praktických příkladů a ilustrací. První a druhá kapitola jsou zaměřeny na popis a analýzu datového souboru Iris, který bude využíván v průběhu práce. Kapitola věnovaná shlukové analýze začíná formulací úlohy a pokračuje uvedením hierarchických metod shlukování společně s vybranými metodami nehiearchického shlukování (k-means, DBSCAN). Poslední část práce je věnována měření kvality shlukování a aplikaci při hledání optimálního počtu shluků. Pro lepší pochopení jsou všechny metody nejdříve popsány intuitivně, poté formulovány matematickým aparátem a následně implementovány v jazyce R.

Klíčová slova: shluková analýza, hierarchické shlukování, nehierarchické shlukování, k-means, DBSCAN

Abstract

The aim of this work is to introduce the reader to the issues of cluster analysis using practical examples and illustrations. The first and second chapters are focused on the description and analysis of the Iris dataset, which will be used during the work. The chapter devoted to cluster analysis begins with the formulation of the task and continues with the introduction of hierarchical clustering methods together with selected methods of non-hierarchical clustering (k-means, DBSCAN). The last part of the work is devoted to the clustering quality measures and application in finding the optimal number of clusters. For a better understanding, all methods are first described intuitively, then formulated with a mathematical apparatus, and then implemented in R.

Keywords: cluster analysis, hierarchical clustering, non-hierarchical clustering, k-means, DBSCAN

Tímto děkuji vedoucí práce, Ing. Martině Litschmannové, Ph.D., za poskytnutí velkého množství času investovaného do rad, připomínek a celkové pomoci při psaní práce. Dále děkuji rodičům za poskytnutí klidu a pohody.

Obsah

Seznam zkratek a symbolů

C	Rozklad
$\bar{C}$	Centroid všech objektů
C_i	Shluk C_i tvořený objekty z C
$\bar{C}_i$	Centroid shluku C_i
MS_B	Mezishluková variabilita
MS_T	Celková variabilita rozkladu
MS_W	Vnitroshluková variabilita
Q_1	Dolní kvartil
Q_3	Horní kvartil
S	Průměrná šířka rozkladu
SS_B	Mezishlukový součet čtverců
SS_T	Celkový součet čtverců
SS_W	Vnitroshlukový součet čtverců
X	Vstupní matice dat
ch	Calinského-Harabaszův index
$d(\mathbf{x}, \mathbf{y})$	Vzdálenost objektů $\mathbf{x}, \mathbf{y}$
$d_e(\mathbf{x}, \mathbf{y})$	Euklidovská vzdálenost objektů $\mathbf{x}, \mathbf{y}$
db	Daviesův-Bouldinův index
dis	Míra nepodobnosti
du	Dunnův index
s	Siluetová funkce
sim	Míra podobnosti
$\mathbf{x}$	Objekt $\mathbf{x}$
x_i	itý-atribut objektu $\mathbf{x}$
$\|C_i\|$	Počet objektů ve shluku C_i
$\delta(C_x, C_y)$	Vzdálenost shluků C_x, C_y

Seznam obrázků

Seznam tabulek

Kapitola 1

Úvod

Žijeme v informačním věku. Toto historické období začalo ve 20. století, jehož počátek je spojen s vynálezem tranzistoru. Hlavním znakem tohoto období jsou informační technologie, konkrétněji počítače. S technologickým pokrokem vznikla možnost imitace určitých aspektů naší inteligence pomocí strojů, tj. umělá inteligence.

Podkategorií umělé inteligence je strojové učení, které má počátky v 50. letech 20. století. Jedná se o soubor algoritmů a matematických modelů, jejichž cílem je pochopení vnitřní struktury dat, díky nimž se následně vytvářejí predikce na datech nových. Je proto logické, že velká část modelů strojového učení vychází z modelů statistických. S příchodem internetu se přístup k datům zjednodušil a množství dat mnohonásobně zvětšilo. Důsledkem je obrovská akcelerace této oblasti.

Cílem práce je seznámit čtenáře s jednou z částí statistických metod strojového učení, shlukovou analýzou, konkrétně spadající pod vícerozměrnou statistickou analýzu.

Shluková analýza je využívána například při taxonomii, tj. klasifikaci organismů, kde biologové jsou na základě DNA schopni určit shluky, chceme-li druhy organismů s podobnými genetickými rysy. Lékaři využívají shluky k nalezení pacientů s podobnými symptomy, na základě čehož je uzpůsobena léčba. Dalším příkladem je smart advertising, který využivá shluků uživatelů se stejnými zájmy pro volbu reklamy, podobným způsobem funguje také doporučování hudby a filmů. Shluková analýzá je také využívána při detekci spamu a analýze dokumentů.

Můžeme tedy říci, že shluková analýzá na základě atributů souboru přiřadí jednotlivým záznamům shluk, do kterého patří. Výsledné shluky jsou tak oproti původnímu souboru konzistentnější z hlediska atributů, které sledujeme.

Shluková analýza je tvořena řadou metod, které mají stejný cíl, ale dosahují jej různými postupy. V této práci se zaměříme na základní z těchto metod, přičemž vše budeme prezentovat na modelovém příkladu se známým souborem Iris. V prní částí práce se s tímto datovým souborem seznámíme. Následně se budeme zabývat shlukovou analýzou, definicí úlohy shlukování, metodami shlukové analýzy a měřením kvality shlukování.

V praxi se používají knihovny s již implementovanými, optimalizovanými algoritmy. V této práci je každá z metod popsána nejdříve intuitivně, poté zformulována matematickým aparátem a následně implementována v software R. Výstupy implementovaných funkcí jsou často graficky zobrazeny pomocí knihoven *ggplot* a *factoextra*. V příloze se nachází každá z implementací jejíž výstup je srovnán s výstupy funkcí z odpovídajících knihoven v R.

1.1 Dataset Iris

Jak již bylo uvedeno, principy shlukové analýzy budeme demonstrovat na jednom z nejznámějších datových souborů, datovém souboru (*datasetu*) Iris. Vytvořil jej statistik a biolog Ronald Fisher v roce 1936 a využil jej ve svém vědeckém článku zaměřeném na lineární diskriminační analýzu.

Datový soubor obsahuje záznamy o 50 květech třech druhů kosatců: Setosa, Virginica a Versicolor (viz obrázek 1.1). Každý záznam je tvořen délkou a šířkou okvětního (*petal*) a kališního (*sepal*) lístku v jednotkách centrimetrů. Záznam také obsahuje druh kosatce, ke kterému se měření vztahuje. Jednotlivé atributy a jejich odpovídající názvy proměnných v souboru můžeme vidět v tabulce 1.1.

Tabulka 1.1: Označení atributů kosatců v souboru Iris

Atribut	Název proměnné
Délka kališního lístku (cm)	sepal_l
Šířka kališního lístku (cm)	sepal_w
Délka okvětního lístku (cm)	petal_l
Šířka okvětního lístku (cm)	petal_w

Jak již bylo uvedeno, počet záznamů v souboru je 150. Našim cílem bude pomoci shlukové analýzy roztřídit kosatce do tří druhů (shluků) za předpokladu, že jejich druh neznáme. Následně znalost biologického druhu kosatců využijeme k posouzení kvality shlukování.

Obrázek 1.1: Druhy kosatců zařazených do souboru Iris, zleva - Setosa (Zdroj: [O1]), Versicolor (Zdroj: [O2]) a Virginica (Zdroj: [O3]).

Kapitola 2

Analýza dat

Při práci s novým datovým souborem je prvním krokem jeho explorační analýza, poskytující informace o atributech s nimiž následně pracujeme. Lze očekávat, že jednotlivé druhy kosatců se ve sledovaných atributech liší, nás zajímá jak moc. Jaké je rozložení atributů? Obsahuje datový soubor statistické jednotky, dále jen záznamy, které jsou neúplné? Jaké jsou extrémy? Na tyto otázky potřebujeme nalézt odpovědi, abychom si udělali představu o datovém souboru.

Následně se můžeme ptát na důmyslnější otázky a formulovat hypotézy. Znalost souboru a metod, se kterými se seznámíme, nám umožní lepší orientaci v postupu, který zvolit pro řešení problému.

2.1 Příprava datového souboru

Pro první seznámení se podívejme na vzorek dat ze souboru. Využijeme názvu proměnných, které jsme zavedli v tabulce 1.1. Vzorek dat ze souboru můžeme vidět v tabulce 2.1.

Tabulka 2.1: Ukázka několika záznamů datového souboru Iris

ID	sepal_l	sepal_w	petal_l	petal_w	druh
1	5,1	3,5	1,4	0,2	Setosa
2	4,9	3,0	1,4	0,2	Setosa
3	4,7	3,2	1,3	0,2	Setosa

Z tabulky 2.1 se zdá, že minimálně u prvních tří kosatců je kališní lístek větší i širší, než lístek okvětní. Dalším pozorováním je poměr délky a šířky lístku. U kališního lístku tento poměr činí cca 2:1, u lístku okvětního cca 7:1, tj. délka lístku je sedminásobně větší než jeho šířka.

Tyto vlastnosti jsme ovšem určili na základě vzorku dat z tabulky 2.1, který může být tvořen extrémy souboru. Zároveň si musíme uvědomit, že uvedený vzorek se vztahuje pouze ke kosatcům Setosa. Z těchto důvodů je vhodné pracovat se statistickými charakteristikami celého souboru se kterým pracujeme.

2.1.1 Chybějící hodnoty

Před popisem souboru je nutno zkontrolovat záznamy a jejich hodnoty. Hodnota, která je z nějakého důvodu neznámá (NA) se nazývá chybějící. Záznamy, které obsahují chybějící hodnotu jsou problémové. Položme si otázku, co by nastalo, kdyby některé z atributů obsahovaly chybějící hodnoty.

Kupříkladu potřebujeme spočítat průměr jednotlivých atributů v souboru. Jelikož každý z atributů může obsahovat chybějící hodnoty v jiném záznamu, vztahovaly by se průměry na odlišné datové soubory. V takovém případě je vhodné, pokud se tímto nepřípravíme o mnoho záznamů, odebrat záznamy s chybějícími hodnotami, popř. zvážit jejich nahrazení průměrem nebo mediánem.

V software *R* můžeme využít příkazu: is.na(data) k identifikaci chybějících údajů. Bylo zjištěno, že soubor Iris neobsahuje žádný záznam s chybějící hodnotou, není tedy potřeba odebírat žádné záznamy.

2.1.2 Odlehlá pozorování

Další komplikací při analýze dat může způsobit výskyt tzv. odlehlých pozorování, tj. záznamů, které v daném atributu vykazují značně odlišné hodnoty oproti ostatním záznamům. Problém ilustrujeme opět na průměru. Pokud si počítáme průměr známek, můžeme pozorovat, že vliv známky na průměr rostl se vzdáleností od průměru. Například průměr čtyřkaře tolik neovlivní pětka, jako průměr jedničkáře. Odlehlá pozorování mají tendenci drasticky měnit průměr.

Jak tato pozorování určit? Jednou z metod, které slouží k identifikaci odlehlých pozorování je metoda vnitřních hradeb. Označme x_p $100p\%$ kvantil. Je zřejmé, že v intervalu $\langle x_{0,25}, x_{0,75}\rangle$ leží alespoň polovina dat.

Pokud tento interval rozšíříme symetricky o $1,5$ násobek interkvartilového rozpětí ($x_{0,75} - x_{0,25}$), získáme tzv. vnitřní hradby. Mezi vnitřními hradbami by měla ležet většina dat. Data, která leží mimo vnitřní hradby označíme jako odlehlá pozorování.

Komplexní analýza odlehlých pozorování v našem případě nemá smysl, jelikož každý druh má jiné statistické charakteristiky. Například kosatec druhu Setosa může mít vyšší průměr určitého atributu než ostatní druhy kosatců, ovšem to se z komplexní analýzy nedozvíme.

Odlehlá pozorování má tedy smysl hledat u jednotlivých druhů kosatců. Výskyt odlehlých pozorování u jednotlivých atributů druhů kosatců vidíme v tabulce 2.2.

Tabulka 2.2: Počet odlehlých pozorování atributů dle druhů kosatců

Druh	sepal_l	sepal_w	petal_l	petal_w
Setosa	0	0	4	2
Versicolor	0	0	1	0
Virginica	1	3	0	0

Z tabulky 2.2 vidíme, že odlehlá pozorování se vyskytují, převážně u kosatců druhu Setosa. Můžeme tedy všechny záznamy obsahující tato odlehlá pozorování odebrat? Počty odlehlých pozorování nám neřeknou nic o jednotlivých odlehlých pozorováních, proto je lepší využít krabicových grafů, viz obrázek 2.1.

Jaký je dopad odlehlých pozorování na krabicový graf? Krabicový graf se graficky „zplošťuje“ s rostoucí vzdáleností odlehlého pozorování od průměru daného atributu. Jelikož krabicové grafy na obrázku 2.1 nepůsobí „zploštěle“ a záznamů je pouze 50, rozhodli jsme se ponechat tyto záznamy v souboru. Otázky, které si můžeme klást, souvisí se vznikem těchto odlehlých pozorování.

2.2 Výběrové charakteristiky

Přesnější představu o chování jednotlivých atributů nám dávají jejich výběrové charakteristiky, což jsou hodnoty popisující datový soubor se kterým pracujeme. Každá z výběrových charakteristik nám poskytuje nový pohled na soubor. Stejně jako u odlehlých pozorování nás zajímají druhové charakteristiky atributů, které můžeme vidět v tabulce 2.3.

Prvními charakteristikami jsou maximum a minimum, díky kterým získáme rozmezí hodnot atributů. Po přidání průměru, získáme hrubou představu o rozložení hodnot atributu. Pro upřesnění představy o rozložení hodnot využijeme medián a kvartily.

Pro posouzení variability kolem průměru využijeme směrodatnou odchylku. Podílem směrodatné odchylky a průměru získáme variační koeficient, který je rovněž vhodnou mírou varibility.

Další charakteristikou kterou využijeme je koeficient šikmosti, popisující symetrii rozložení hodnot atributů. Nulová šikmost odpovídá hodnotám, které jsou symetricky rozprostřeny kolem průměru. Nenulové hodnoty odpovídají určité asymetrii hodnot, v případě kladné šikmosti je více než polovina dat menších než průměr, u záporné šikmosti je tomu naopak.

Poslední charakteristikou je koeficient špičatosti, popisující špičatost rozdělení ve srovnání s rozdělením normálním (*Gaussovým*). Nulová hodnota odpovídá normálnímu rozdělení, s rostoucím koeficientem klesá vzdálenost hodnot od průměru, čemuž odpovídá větší špičatost. Zmíněné výběrové charakteristiky atributů jednotlivých druhů kosatců vidíme v tabulce 2.3.

Tabulka 2.3: Výběrové charakteristiky jednotlivých atributů dle druhu kosatce

sepal_l (cm)

	Setosa	Versicolor	Virginica
min	4,3	4,9	4,9
Q_1	4,80	5,60	6,23
průměr	5,01	5,94	6,59
medián	5,00	5,90	6,50
Q_3	5,20	6,30	6,90
max	5,8	7,0	7,9
sm. odch.	0,36	0,52	0,64
var. koef.	0,07	0,09	0,10
šikmost	0,12	0,10	0,11
špicatost	−0,35	−0,60	−0,09

sepal_w (cm)

	Setosa	Versicolor	Virginica
min	2,3	2,0	2,2
Q_1	3,13	2,53	2,80
průměr	3,42	2,77	2,97
medián	3,40	2,80	3,00
Q_3	3,68	3,00	3,18
max	4,4	3,4	3,8
sm. odch.	0,39	0,32	0,33
var. koef.	0,11	0,11	0,11
šikmost	0,10	−0,35	0,36
špicatost	0,69	−0,45	0,52

petal_l (cm)

	Setosa	Versicolor	Virginica
min	1,0	3,0	4,5
Q_1	1,40	4,00	5,10
průměr	1,46	4,26	5,55
medián	1,50	4,35	5,55
Q_3	1,58	4,60	5,88
max	1,9	5,1	6,9
sm. odch.	0,18	0,47	0,56
var. koef.	0,12	0,11	0,10
šikmost	0,07	−0,59	0,53
špicatost	0,81	−0,07	−0,26

petal_w (cm)

	Setosa	Versicolor	Virginica
min	0,1	1,0	1,4
Q_1	0,20	1,20	1,80
průměr	0,24	1,33	2,03
medián	0,20	1,30	2,00
Q_3	0,30	1,50	2,30
max	0,6	1,8	2,5
sm. odch.	0,11	0,20	0,28
var. koef.	0,44	0,15	0,14
šikmost	1,16	−0,03	−0,13
špicatost	1,30	−0,49	−0,66

Využijeme tabulky 2.3 k analýze jednoho z atributů, v tomto případě délky okvětního lístku (*petal_l*) kosatce druhu Setosa. Změřená délka okvětních lístků kosatce Setosa se pohybuje v rozpětí 1,0 cm až 1,9 cm. Zajímavým pozorováním je, že rozpětí délky okvětního lístku kosatce Setosa nezasahuje do rozpětí zbylých dvou druhů, jejichž rozpětí se částečně překrývá.

Průměrná délka okvětních lístků kosatce Setosa je 1,46 cm. Také zde můžeme zpozorovat odlišnost průměrné velikosti délky okvětních lístků každého druhu kosatce. Směrodatná odchylka délky okvětního lístku kosatce Setosa činí 0,18 cm. Díky variačnímu koeficientu (12 %) můžeme očekávat, že hodnoty jsou u Setosy soustředěny v blízkosti průměru.

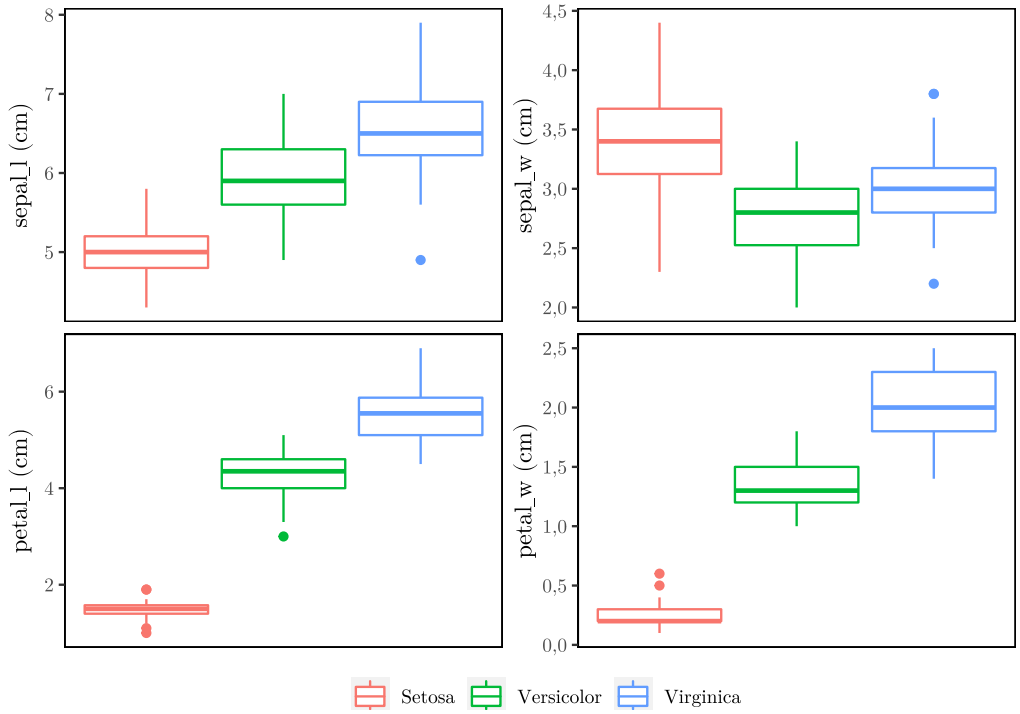
U poloviny testovaných vzorků kosatce Setosa délka okvětních lístků nepřekročila 1,50 cm. V polovině měření se tato délka kosatce Setosa pohybovala v rozmezí 1,40 cm až 1,58 cm. Tyto charakteristiky nám přesněji vymezily rozložení hodnot délky okvětních lístků kosatce druhu Setosa. Obdobným způsobem bychom mohli analyzovat i ostatní atributy jednotlivých druhů kosatců.

2.3 Vizualizace

Výběrové charakteristiky jsme využili k popsání atributů druhů kosatců v souboru. K rozšíření, nebo doplnění naší vědomosti o atributech, se kterými v souboru pracujeme využijeme vizualizace, v našem případě se zaměříme na krabicové grafy a grafy hustot pravděpodobností. Každý z těchto grafů nám pomůže při analýze druhů kosatců a v hledání jejich odlišností.

2.3.1 Krabicové grafy

Krabicové grafy umožňují rychlé vizuální srovnání druhů kosatců z hlediska jednotlivých pozorovaných atributů. Z krabicových grafů můžeme vyčíst některé výběrové charakteristiky jako je maximum, minimum, medián, 1. a 3. kvartil. Tyto grafy se také využívají pro detekci odlehlých pozorování, čehož jsme využili v kapitole 2.1.2.



Obrázek 2.1: Srovnání atributů kosatců dle jejich druhů (krabicové grafy)

Z obrázku 2.1 vidíme, že kosatec druhu Setosa se od zbylých dvou druhů kosatců liší, což lze nejlépe vidět ve vizualizaci délky a šířky okvětního lístku. Kosatce Versicolor a Virginica se zdají hůře rozlišitelné.

Ve vizualizaci šířky kališního lístku vidíme větší překrytí krabicových grafů, než-li u zbylých vizualizací. Z toho můžeme konstatovat, že atributy kališního lístku jednotlivých druhů se neliší tolik, jako například u délky okvětních lístků.

Srovnáním krabicových grafů zjistíme, že kosatec druhu Virginica dosahuje typicky nejvyšších hodnot jednotlivých atributů, zatímco kosatec druhu Setosa hodnot nejnižších, vyjímkou je vizualizace šířky kališního lístku.

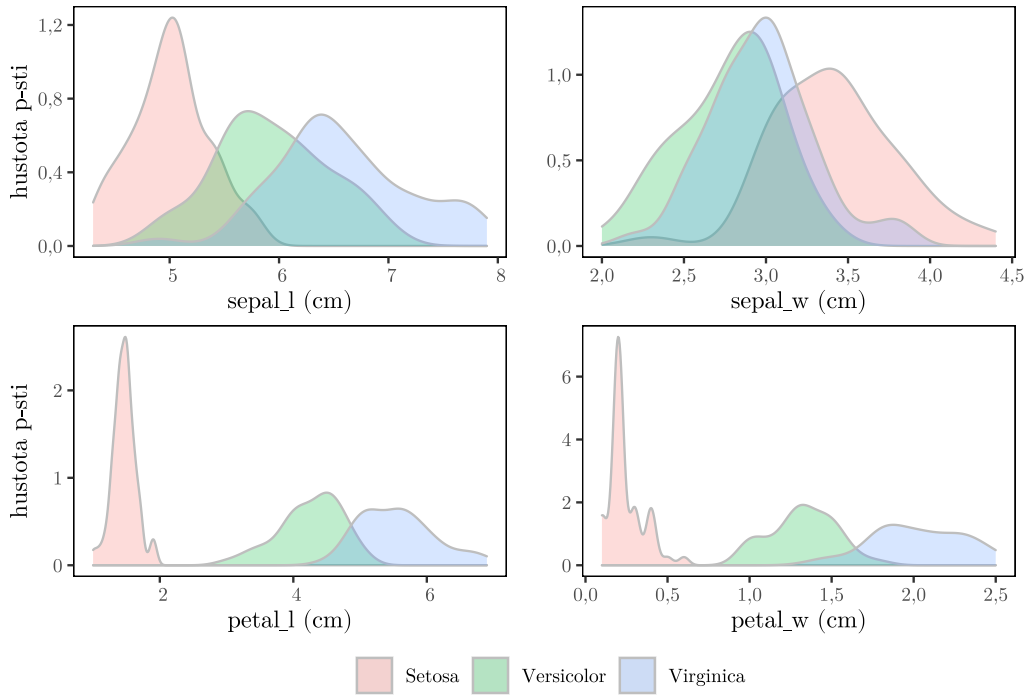
Poslední poznámkou je podobnost krabicových grafů délky a šířky okvětních lístků. Intuitivně nám tyto atributy poskytují podobné informace. Touto podobností bychom se zabývali více pokud bychom chtěli redukovat dimenzionalitu souboru.

2.3.2 Hustota pravděpodobnosti

Další z možností, kterou lze využít ke srovnání jednotlivých atributů je graf hustoty pravděpodobnosti. Analýzu statistických charakteristik si můžeme přečíst znovu a představit si dopad jednotlivých charakteristik na výslednou hustotu pravděpodobnosti atributů.

Očekáváme například, že u délky okvětního lístku bude špičatost u druhu Setosa velká a výsledný graf bude oddělen od zbylých dvou druhů kosatců, které se budou překrývat.

Nyní využijeme zbylých charakteristik z tabulky 2.3 k posouzení normality délky okvětních lístků. Špičatost i šikmost délky okvětních lístků druhu Setosa leží v intervalu $(-2, 2)$, tudíž lze očekávat, že délka okvětních lístků pro druh kosatce Setosa by mohla mít normální rozdělení, což můžeme posoudit i graficky z obrázku 2.2. Pro exaktní posouzení normality by bylo třeba využít některý z testů normality, např. Shapirův-Wilkův test.



Obrázek 2.2: Srovnání atributů kosatců dle jejich druhů (hustoty pravděpodobnosti)

Graf hustot pravděpodobností na obrázku 2.2 nám umožní rychle porovnat rozložení jednotlivých atributů druhů kosatců. Například u šířky kališních lístků se hustoty pravděpodobností překrývají, což z tabulky 2.3 nelze rychle určit.

Informace, které jsme o datasetu Iris získali nám nyní pomohou lépe pochopit mechanismy metod shlukové analýzy.

Kapitola 3

Shluková analýza

Shluková analýza je souhrn metod zabývajících se vyšetřováním podobnosti vícerozměrných objektů (tj. objektů charakterizovaných alespoň dvěma atributy) a jejich klasifikací do tzv. shluků (*clusterů*). Cílem shlukové analýzy je empirická klasifikace objektů, zjednodušení struktury dat a identifikace vztahů mezi objekty.

Algoritmů řešících shlukování je mnoho, částečným důvodem je volnost, kterou poskytují pojmy se kterými ve shlukové analýze pracujeme. S příklady shlukování jsme se již setkali v úvodní kapitole, a proto se rovnou podíváme na formulaci úlohy shlukování.

Označme X naši vstupní matici dat o rozměrech $n \times m$. Tato matice obsahuje n objektů $\mathbf{x}_i$ charakterizovaných m atributy, $\mathbf{x}_i = (x_{i,1}, ..., x_{i,m})$. Dále označme $C = \{\mathbf{x}_i, ..., \mathbf{x}_n\}$ jako množinu všech objektů matice X.

Shlukování je úloha, při které vytváříme rozklad množiny C. Tento rozklad je množina neprázdných, navzájem disjunktních podmožin shluků C_i, jejichž sjednocením je C. Požadavek navzájem disjunktních podmnožin zaručí výskyt jednoho objektu právě v jednom shluku. Vytváříme tedy shluky $C_1, ..., C_k$ $(k \leq n)$ splňující následující podmínky:

$$\begin{aligned}
&\text{(i)} \quad C_i \subset C \quad \forall i = \{1, \dots, k\}, \text{ kde } C_i \neq \emptyset \\
&\text{(ii)} \quad C_i \cap C_j = \emptyset \quad \forall i, j \in \{1, \dots, k\}, \text{ kde } i \neq j \\
&\text{(iii)} \quad C = C_1 \cup \dots \cup C_k
\end{aligned}$$

Ovšem ne každý rozklad je vhodným řešením úlohy. My hledáme rozklad maximalizující podobnost objektů ve shluku a minimalizující podobnost objektů shluků odlišných.

3.1 Vzdálenosti

Celé shlukování pracuje se vzdálenostmi. Vzdálenosti jsou definovány na základě míry nepodobnosti objektů. Definice těchto pojmů lze nalézt v řadě literatury, například: [1], [2] a [3].

Míra podobnosti (Similarity)

Míra podobnosti je funkce $sim : X \times X \to I \subset \mathbb{R}$ přiřazující každé dvojici objektů $\mathbf{x}, \mathbf{y} \in X$ číslo $sim(\mathbf{x}, \mathbf{y}) \in I$, pro které platí:

$$\begin{aligned} &\text{(i)} \quad sim(\mathbf{x}, \mathbf{y}) \leq \sup I \\ &\text{(ii)} \quad sim(\mathbf{x}, \mathbf{y}) = sim(\mathbf{y}, \mathbf{x}) && \text{(symetrie)} \\ &\text{(iii)} \quad sim(\mathbf{x}, \mathbf{y}) = \sup I \iff \mathbf{x} = \mathbf{y} && \text{(identita)} \end{aligned}$$

Čím jsou si objekty podobnější, tím je míra podobnosti *sim* větší. Požadavek (i) omezuje míru podobnosti *sim* shora. Díky tomuto omezení jsme schopni porovnat míru podobnosti objektů $\mathbf{x}, \mathbf{y} \in X$. Největší míra podobnosti odpovídá podobnosti dvou stejných objektů, tj. identitě.

Míra nepodobnosti (Dissimilarity)

Míra nepodobnosti je funkce $dis : X \times X \to I \subset \mathbb{R}$ přiřazující každé dvojici objektů $\mathbf{x}, \mathbf{y} \in X$ číslo $dis(\mathbf{x}, \mathbf{y}) \in I$, pro které platí:

$$\begin{aligned} &\text{(i)} \quad dis(\mathbf{x}, \mathbf{y}) \geq \inf I \\ &\text{(ii)} \quad dis(\mathbf{x}, \mathbf{y}) = dis(\mathbf{y}, \mathbf{x}) && \text{(symetrie)} \\ &\text{(iii)} \quad dis(\mathbf{x}, \mathbf{y}) = \inf I \iff \mathbf{x} = \mathbf{y} && \text{(identita)} \end{aligned}$$

Čím jsou si objekty podobnější, tím je míra nepodobnosti *dis* nižší. Opět zde máme požadavek (i) omezující míru nepodobnosti *dis*, tentokrát zdola.

V praxi se pracuje s mírou nepodobnosti, jedním z důvodů je lepší interpretace a intuitivní analogie se vzdáleností, kterou nyní na základě míry nepodobnosti můžeme definovat.

Vzdálenost (Distance)

Vzdálenost je specifickou mírou nepodobnosti $d : X \times X \to I = \langle 0, \infty)$ přiřazující každé dvojici objektů $\mathbf{x}, \mathbf{y} \in X$ číslo $d(\mathbf{x}, \mathbf{y}) \in I$, pro které platí:

$$\begin{aligned} &\text{(i)} \quad d(\mathbf{x}, \mathbf{y}) \geq 0 && \text{(nezápornost)} \\ &\text{(ii)} \quad d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}) && \text{(symetrie)} \\ &\text{(iii)} \quad d(\mathbf{x}, \mathbf{y}) = 0 \iff \mathbf{x} = \mathbf{y} && \text{(identita)} \\ &\text{(iv)} \quad d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) && \text{(troj. nerovnost)} \end{aligned}$$

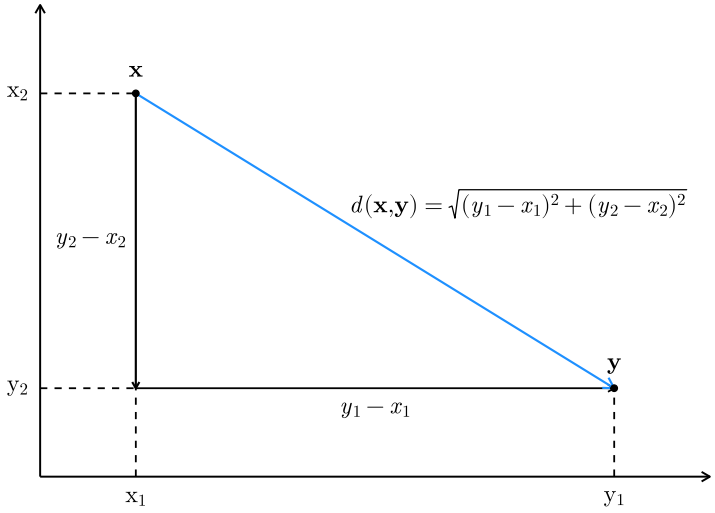
Díky tomu, že $I = \langle 0, \infty)$, je $\inf I = 0$. Dále vidíme nový požadavek (iv) na vzdálenost, trojúhelníkovou nerovnost. Vzdáleností je mnoho, proto se zaměříme na vzdálenosti, se kterými se můžeme nejčastěji setkat v praxi. Srovnání těchto a mnoha jiných vzdáleností při shlukování se zabývá např. [4].

Euklidovská vzdálenost

S Euklidovskou vzdáleností pracujeme již od základní školy. Jedná se o geometrickou vzdálenost v $\mathbb{R}^d$, zobecňující Pythagorovu větu. Na tuto vzdálenost se tedy můžeme dívat jako na nejkratší vzdálenost mezi dvěma body, chceme-li objekty, v prostoru.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

Euklidovská vzdálenost je ovlivněna měřítkem jednotlivých atributů. Dojde zde k podobnému problému jako při výpočtu průměru, vzdálenost je před ovlivněna rozpětím jednotlivých atributů. Euklidovskou vzdálenost v $\mathbb{R}^2$ můžeme vidět na obrázku 3.1.



Obrázek 3.1: Euklidovská vzdálenost objektů $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$

Z důvodu citlivosti Euklidovské vzdálenosti na měřítka atributů je někdy vhodné data před samotným shlukováním standardizovat. Jako standardizace dat se označuje převod dat (proměnných, chceme-li atributů) na stejné měřítko, což eliminuje vliv skutečného rozsahu a velikosti příslušných proměnných. V praxi se nejčastěji používá standardizace směrodatnou odchylkou, kdy se standardizovaná hodnota y proměnné x získá tak, že se od původní hodnoty x odečte její průměr $\bar{x}$ a tento rozdíl se vydělí směrodatnou odchylkou proměnné s_x:

$$y = \frac{x - \bar{x}}{s_x}$$

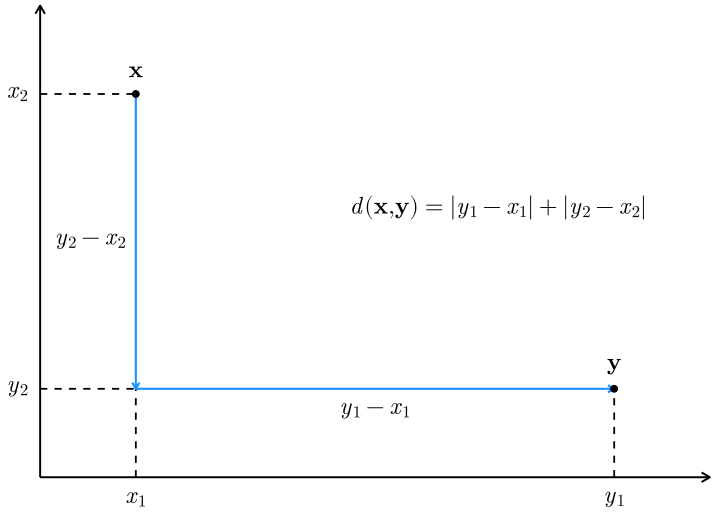
Jak již bylo zmíněno, výhodou standardizace je eliminace vlivu skutečného rozsahu a velikosti jednotlivých proměnných, na druhou stranu bychom měli mít na paměti i to, že standardizace může vést ke ztrátě informace a tím i k horším výsledkům klasifikace a přistupovat k ní proto až po důsledném promyšlení. Více lze najít např. v [5]. Atributy datasetu Iris nemají drasticky lišící se rozpětí, proto jednotlivé atributy standardizovat nebudeme.

Manhattanská vzdálenost

Linearizací vzdálenosti Euklidovské získáváme vzdálenost Manhattenskou, jejíž inspirace vychází z orientací ulic v Manhattnu. Tato vzdálenost je definovaná jako součet vzdáleností mezi jednotlivými, odpovídajícími atributy.

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} |x_i - y_i|$$

Díky linearizaci snížíme dopad atributů s větším rozptylem, tudíž zrovnoměrníme dopad jednotlivých atributů na výslednou vzdálenost. Tuto vzdálenost je tedy vhodné použít při velmi velkém počtu atributů [6]. Nevýhodou je závislost vzdálenosti na konkrétní rotaci souřadnicového systému. Tuto vzdálenost v $\mathbb{R}^2$ můžeme vidět na obrázku 3.2.



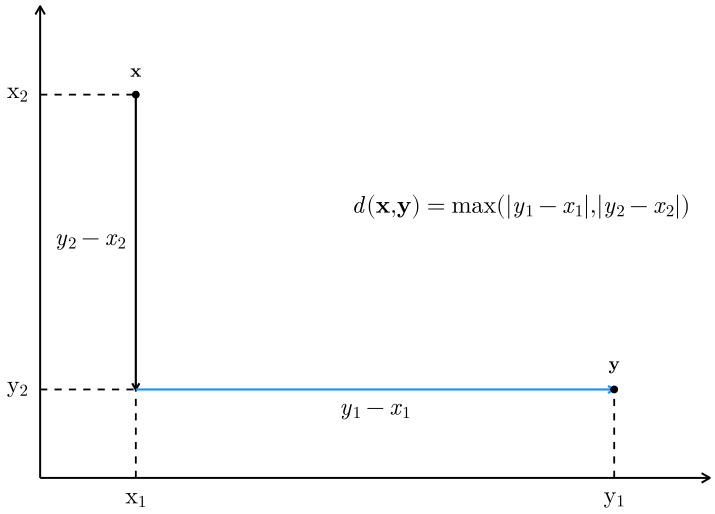
Obrázek 3.2: Manhattanská vzdálenost objektů $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$

Maximální vzdálenost

Tato vzdálenost je definována jako maximální vzdálenost příslušných atributů.

$$d(\mathbf{x}, \mathbf{y}) = \max_i |x_i - y_i|$$

Maximální vzdálenost je vhodné použít, pokud objekty považujeme za odlišné, liší-li se alespoň v jednom z atributů. S touto vzdáleností se můžeme setkat také pod názvem Čebyševova vzdálenost. Vizualizaci maximální vzdálenosti v $\mathbb{R}^2$ můžeme vidět na obrázku 3.3.



Obrázek 3.3: Maximální vzdálenost objektů $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$

Minkowského vzdálenost

Zobecněním všech zmíněných vzdáleností je Minkowského vzdálenost

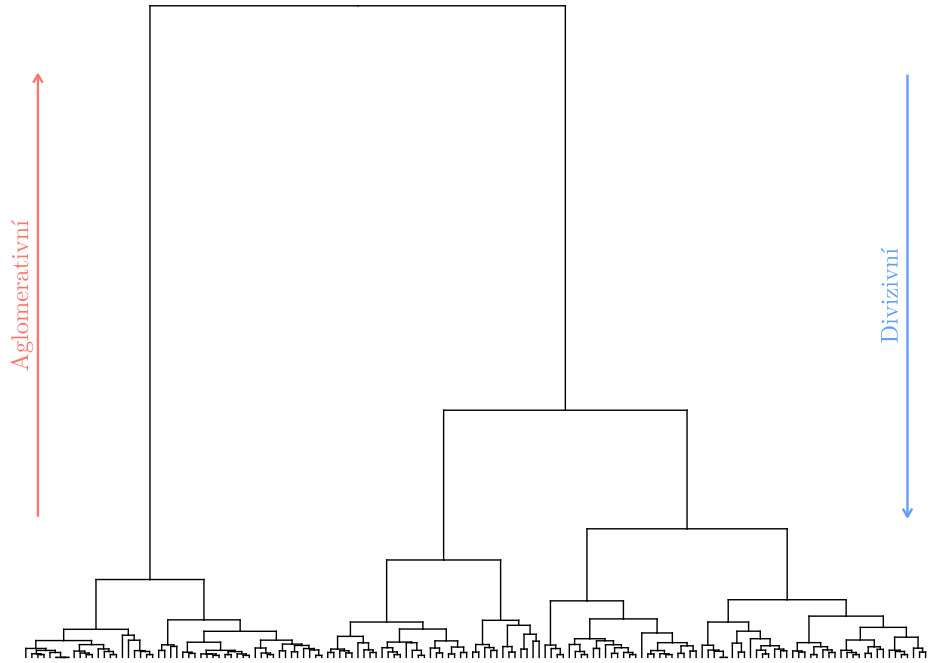
$$d(\mathbf{x}, \mathbf{y}) = \sqrt[p]{\left(\sum_{n=1}^{m} |x_i - y_i|^p\right)},$$

kde pro $p = 1$ dostáváme vzdálenost Manhattanskou, pro $p = 2$ Euklidovskou vzdálenost a pro $p = \infty$ vzdálenost maximální.

3.2 Rozdělení metod

Již jsme zmínili množství algoritmů řešících shlukování. Pro snažší orientaci se algoritmy dělí dle principu a výsledného výstupu, který produkují. Základní typy shlukování jsou hierarchické a nehierarchické.

Hierarchické shlukování vytváří iterativním způsobem rozklady, které společně tvoří hierarchický strom, tzv. dendrogram. Stromová struktura poskytuje lepší orientaci při práci se shluky. Hierarchické metody dále můžeme rozdělit dle směru tvorby dendrogramu, na aglomerativní a divizivní. Dendrogram i se směry tvorby můžeme vidět na obrázku 3.4.



Obrázek 3.4: Dendrogram datasetu Iris s vyznačenými směry tvorby

Dalším typem je shlukování nehierarchické. U tohoto typu se snažíme minimalizovat již zmíněnou míru nepodobnosti. Z tohoto důvodu je nehierchické shlukování komplexnější. Diverzita algoritmů je zde způsobena volností, kterou postkytuje volba míry nepodobnosti.

Metody spadající pod nehierarchické shlukování typicky pracují se vstupním kritériem, například předem určeným počtem shluků a funkcí nepodobnosti, na jejimž základě každou iterací upravují, zpřesňují svou klasifikaci objektů do shluků.

My se podíváme na hieararchické i nehierarchické shlukování, u nehierarchického shlukování se zaměříme na metody: k-means a DBSCAN. Mezi další, používané metody patří například: Affinity propagation, Mean-shift, OPTICS, Expectation maximization a Spectral clustering (viz [7]).

3.3 Hierarchické shlukování

Jak již bylo zmíněno, hierarchické shlukování pracuje na základě iterativních rozkladů. Tyto rozklady jsou buď tvořeny postupným slučováním shluků (aglomerativní metody), nebo postupným dělením shluků (divizivní metody). Publikací popisujících hierarchické shlukování je mnoho, uveďme například [5], [8], [9] a [10].

Označme počet shluků v rozkladu k. U aglomerativních metod tvoří počáteční rozklad shluky obsahující pouze jeden objekt ($k = n$). Dále se iterativním způsobem slučují nejpodobnější shluky. Algoritmus končí při $k = 1$, čímž dostáváme strom, který se s každou iterací zužuje.

Naopak u divizivních metod je počáteční rozklad tvořen jediným shlukem, tj. celou množinou C ($k = 1$). Dále se iterativním způsobem oddělují nejméně podobné shluky. Algoritmus končí při $k = n$, čímž dostáváme strom, který se s každou iterací rozšiřuje.

Pro zvýšení rychlosti algoritmy typicky pracují se symetrickou maticí vzdáleností D o rozměrech $n \times n$ tvořenou prvky matice $D_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$. Tato matice je tedy tvořena vzdálenostmi všech dvojic objektů matice X.

My budeme pracovat s aglomerativními metodami. Již jsme definovali vzdálenosti mezi objekty. Pozorný čtenář si ovšem mohl všimnout problému: není zatím definován způsob, jakým budeme slučovat/oddělovat shluky mezi sebou.

3.3.1 Vzdálenosti mezi shluky

Vzdálenosti mezi shluky, také označovány jako vazby (*linkage*), jsou mírou nepodobnosti $\delta : C \times C \to I = \langle 0, \infty)$ přiřazující každé dvojici shluků $C_x, C_y \in C$ číslo $\delta(C_x, C_y) \in I$, pro které platí:

$$\begin{aligned} &\text{(i)} \quad \delta(C_x, C_y) \geq 0 && \text{(nezápornost)} \\ &\text{(ii)} \quad \delta(C_x, C_y) = \delta(C_x, C_y) && \text{(symetrie)} \\ &\text{(iii)} \quad \delta(C_x, C_y) = 0 \iff C_x = C_y && \text{(identita)} \end{aligned}$$

Existuje mnoho metod definujících vzdálenosti mezi shluky, my se podíváme na základní z nich. Při každé z těchto vzdáleností uvedeme její dopad na matici vzdáleností D při sloučení dvou shluků s příkladem výpočtu a grafickou interpretací metody.

Shlukování jednotlivými metodami budeme demonstrovat na shlucích C_x, C_y, obsahujících objekty tvořené dvěma atributy, jejichž hodnoty jsou uvedeny v tabulce 3.1.

Tabulka 3.1: Popis objektů ve shlucích C_x, C_y (modelový příklad)

$C_x = C_{abcd}$		
Objekt	1.atribut	2.atribut
a	0,95	1,30
b	1,20	0,85
c	1,30	1,40
d	0,85	0,70

$C_y = C_{efg}$		
Objekt	1.atribut	2.atribut
e	3,50	3,00
f	3,00	2,60
g	3,30	2,30

Metoda nejbližšího souseda

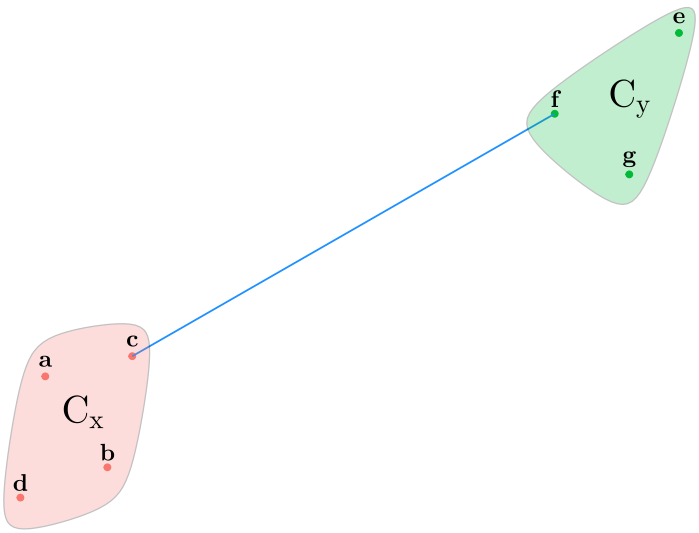
Uvažujme libovolnou vzdálenost d. Metoda nejbližšího souseda definuje vzdálenost dvou shluků C_x, C_y jako minimální vzdálenost objektů $\mathbf{x} \in C_x, \mathbf{y} \in C_y$.

$$\delta(C_x, C_y) = \min_{\mathbf{x} \in C_x, \mathbf{y} \in C_y} d(\mathbf{x}, \mathbf{y})$$

Sloučení shluků C_x, C_y vede ke změně matice vzdáleností, přičemž vzdálenost sloučených shluků $C_x \cup C_y$ od shluku C_z určíme dle vztahu

$$\delta(C_x \cup C_y, C_z) = \min(\delta(C_x, C_z), \delta(C_y, C_z)).$$

Metoda nejbližšího souseda má tendenci vytvářet podlouhlé shluky. Tento jev je nazýván řetězením (*chaining*) a je hlavním důvodem, proč se s touto metodou v praxi nesetkáme příliš často. Ilustraci zjištění vzdálenosti shluků metodou nejbližšího souseda můžeme vidět na obrázku 3.5.



Obrázek 3.5: Ilustrace zjištění vzdálenosti shluků C_x, C_y metodou nejbližšího souseda

Pro ukázku konkrétní aplikace metody nejbližšího souseda si ukážeme jak určit Euklidov-

skou vzdálenost pro shluky C_x, C_y. Z obrázku 3.5 je zřejmé, že použijeme-li metodu nebližšího souseda, získáme:

$$\delta(C_x, C_y) = \min_{\mathbf{x}\in C_x, \mathbf{y}\in C_y} d_e(\mathbf{x}, \mathbf{y}) = d_e(\mathbf{c}, \mathbf{f}) = \sqrt{(f_1 - c_1)^2 + (f_2 - c_2)^2} \doteq 2,08.$$

Je vhodné poznamenat, že bez Obrázku 3.5 by bylo nutné vypočítat všechny vzdálenosti mezi objekty ve shlucích C_x, C_y a vybrat nejmenší z nich.

Příklad

Nyní si ukážeme postup hierarchického shlukování metodou nejbližšího souseda s Euklidovskou vzdáleností. Vstupní matice dat X odpovídá hodnotám uvedeným v Tabulce 3.1. Odpovídající matice vzdáleností D je tvořena Euklidovskými vzdálenostmi mezi všemi dvojicemi shluků v X. Vzhledem k symetrii vzdálenosti jde o symetrickou matici.

Počáteční shluky jsou tvořeny pouze jedním objektem, tudíž vzdálenosti mezi shluky jsou Euklidovskými vzdálenostmi mezi odpovídajícími objekty $\delta(C_a, C_b) = d_e(a, b)$.

	C_a	C_b	C_c	C_d	C_e	C_f	C_g
C_a	0	0,514782	0,364006	0,608276	3,064719	2,427447	2,553919
C_b	0,514782	0	0,559017	0,380789	3,148412	2,510478	2,551960
C_c	0,364006	0,559017	0	0,832166	2,720294	2,080865	2,193171
C_d	0,608276	0,380789	0,832166	0	3,508917	2,869233	2,926175
C_e	3,064719	3,148412	2,720294	3,508917	0	0,640312	0,728011
C_f	2,427447	2,510478	2,080865	2,869233	0,640312	0	0,424264
C_g	2,553919	2,551960	2,193171	2,926175	0,728011	0,424264	0

Všimněme si, že matice vzdálenosti je maticí symetrickou. Proto se často uvádí pouze horní trojúhelníková část této matice. Nyní provedeme sloučení shluků na základě δ, hledáme nejmenší vzdálenost jednotlivých dvojic různých shluků v matici vzdáleností D. Nejmenší vzdálenost je mezi shluky C_a, C_c ($\delta(C_a, C_c) = 0,364006$), tudíž tyto shluky sloučíme.

Po sloučení $C_a \cup C_c$ dojde ke změně matice D, přičemž vzdálenost nového shluku $C_a \cup C_c$ od shluků ostatních určíme dle vztahu: $\delta(C_a \cup C_c, C_z) = \min(\delta(C_a, C_z), \delta(C_c, C_z))$. Pro příklad spočteme novou vzdálenost po sloučení $C_a \cup C_c$ od shluku C_b.

$$\delta(C_{ac}, C_b) = \delta(C_a \cup C_c, C_b) = \min(\delta(C_a, C_b), \delta(C_c, C_b)) = \min(0{,}514782, 0{,}559017) = 0{,}514782$$

Označme červeně přepočtené vzdálenosti s nově sloučeným shlukem. Po přepočtení všech

těchto vzdáleností získáme:

$$\begin{array}{c} \\ C_{ac} \\ C_b \\ C_d \\ C_e \\ C_f \\ C_g \end{array} \begin{array}{c} \begin{array}{cccccc} C_{ac} & C_b & C_d & C_e & C_f & C_g \end{array} \\ \begin{bmatrix} 0 & 0,514782 & 0,608276 & 2,720294 & 2,080865 & 2,193171 \\ 0,514782 & 0 & 0,380789 & 3,148412 & 2,510478 & 2,551960 \\ 0,608276 & 0,380789 & 0 & 3,508917 & 2,869233 & 2,926175 \\ 2,720294 & 3,148412 & 3,508917 & 0 & 0,640312 & 0,728011 \\ 2,080865 & 2,510478 & 2,869233 & 0,640312 & 0 & 0,424264 \\ 2,193171 & 2,551960 & 2,926175 & 0,728011 & 0,424264 & 0 \end{bmatrix} \end{array}$$

Nejmenší vzdálenost je $\delta(C_b, C_d) = 0{,}380789$. Po sloučení $C_b \cup C_d$ získáme:

$$\begin{array}{c} \\ C_{ac} \\ C_{bd} \\ C_e \\ C_f \\ C_g \end{array} \begin{array}{c} \begin{array}{ccccc} C_{ac} & C_{bd} & C_e & C_f & C_g \end{array} \\ \begin{bmatrix} 0 & 0{,}514782 & 2{,}720294 & 2{,}080865 & 2{,}193171 \\ 0{,}514782 & 0 & 3{,}148412 & 2{,}510478 & 2{,}551960 \\ 2{,}720294 & 3{,}148412 & 0 & 0{,}640312 & 0{,}728011 \\ 2{,}080865 & 2{,}510478 & 0{,}640312 & 0 & 0{,}424264 \\ 2{,}193171 & 2{,}551960 & 0{,}728011 & 0{,}424264 & 0 \end{bmatrix} \end{array}$$

Nejmenší vzdálenost je $\delta(C_f, C_g) = 0{,}424264$. Po sloučení $C_f \cup C_g$ získáme:

$$\begin{array}{c} \\ C_{ac} \\ C_{bd} \\ C_e \\ C_{fg} \end{array} \begin{array}{c} \begin{array}{cccc} C_{ac} & C_{bd} & C_e & C_{fg} \end{array} \\ \begin{bmatrix} 0 & 0{,}514782 & 2{,}720294 & 2{,}080865 \\ 0{,}514782 & 0 & 3{,}148412 & 2{,}510478 \\ 2{,}720294 & 0{,}514782 & 2{,}720294 & 0{,}640312 \\ 2{,}080865 & 2{,}510478 & 0{,}640312 & 0 \end{bmatrix} \end{array}$$

Nejmenší vzdálenost je $\delta(C_{ac}, C_{bd}) = 0{,}514782$. Po sloučení $C_{ac} \cup C_{bd}$ získáme:

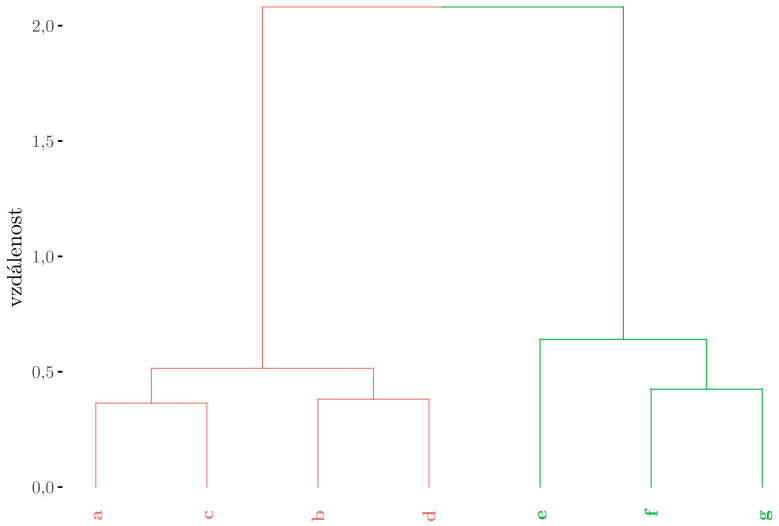
$$\begin{array}{c} \\ C_{abcd} \\ C_e \\ C_{fg} \end{array} \begin{array}{c} \begin{array}{ccc} C_{abcd} & C_e & C_{fg} \end{array} \\ \begin{bmatrix} 0 & 2{,}720294 & 2{,}080865 \\ 2{,}720294 & 0 & 0{,}640312 \\ 2{,}080865 & 0{,}640312 & 0 \end{bmatrix} \end{array}$$

Nejmenší vzdálenost je $\delta(C_e, C_{fg}) = 0{,}640312$. Po sloučení $C_e \cup C_{fg}$ získáme:

$$\begin{array}{c} \\ C_{abcd} \\ C_{efg} \end{array} \begin{array}{c} \begin{array}{cc} C_{abcd} & C_{efg} \end{array} \\ \begin{bmatrix} 0 & 2{,}080865 \\ 2{,}080865 & 0 \end{bmatrix} \end{array}$$

Pro poslední iteraci je nejmenší vzdálenost $\delta(C_{abcd}, C_{efg}) = 2{,}080865$. Na základě provedeného iterovaného postupu můžeme sestavit dendrogram. Vzdálenost, ve které se shluky sloučily

odpovídá vzdálenosti δ mezi slučovanými shluky. Výsledný dendrogram můžeme vidět na obrázku 3.6.



Obrázek 3.6: Výsledný dendrogram vytvořený metodou nejbližšího souseda

Jediným rozdílem u dalších metod hierarchického shlukování je změna matice D na základě příslušné volby, jak určit vzdálenost shluků δ, proto pro ostatní metody nebude příklad uveden.

Metoda nejvzdálenějšího souseda

Uvažujme libovolnou vzdálenost d. Metoda nejvzdálenějšího souseda definuje vzdálenost dvou shluků C_x, C_y jako maximální vzdálenost objektů $\mathbf{x} \in C_x, \mathbf{y} \in C_y$.

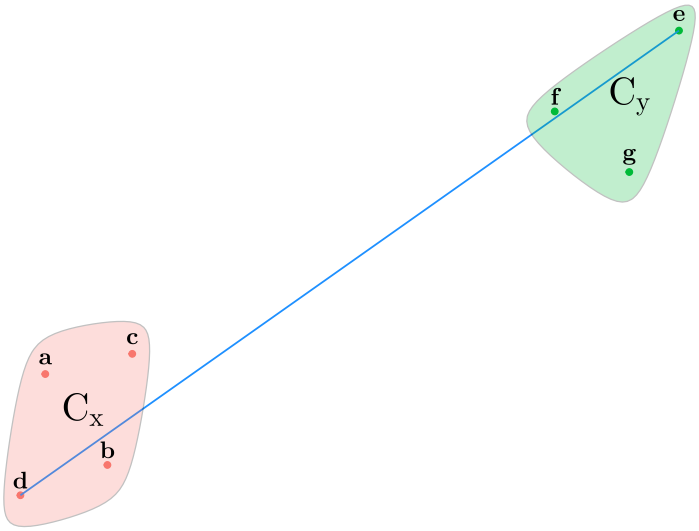
$$\delta(C_x, C_y) = \max_{\mathbf{x} \in C_x, \mathbf{y} \in C_y} d(\mathbf{x}, \mathbf{y})$$

Sloučení shluků C_x, C_y vede ke změně matice vzdáleností, přičemž vzdálenost sloučených shluků $C_x \cup C_y$ od shluku C_z určíme dle vztahu

$$\delta(C_x \cup C_y, C_z) = \max(\delta(C_x, C_z), \delta(C_y, C_z)).$$

Oproti metodě nejbližšího souseda zde nedochází k řetězení. Metoda nejvzdálenějšího souseda tvoří typicky kompaktní, kruhové shluky.

Vzniká zde ovšem problém, kdy určitý objekt může být blíže objektům z odlišných shluků než objektům z vlastního shluku. Ilustraci zjištění vzdálenosti shluků metodou nejvzdálenějšího souseda můžeme vidět na obrázku 3.7.



Obrázek 3.7: Ilustrace zjištění vzdálenosti shluků C_x, C_y metodou nejbližšího souseda

Pro ukázku konkrétní aplikace metody nejvzdálenějšího souseda si ukážeme jak určit Euklidovskou vzdálenost pro shluky C_x, C_y. Z obrázku 3.7 je zřejmé, že použijeme-li metodu nejvzdálenějšího souseda, získáme

$$\delta(C_x, C_y) = \max_{\mathbf{x} \in C_x, \mathbf{y} \in C_y} d_e(\mathbf{x}, \mathbf{y}) = d_e(\mathbf{d}, \mathbf{e}) = \sqrt{(e_1 - d_1)^2 + (e_2 - d_2)^2} \doteq 3{,}51.$$

Metoda průměrné vazby

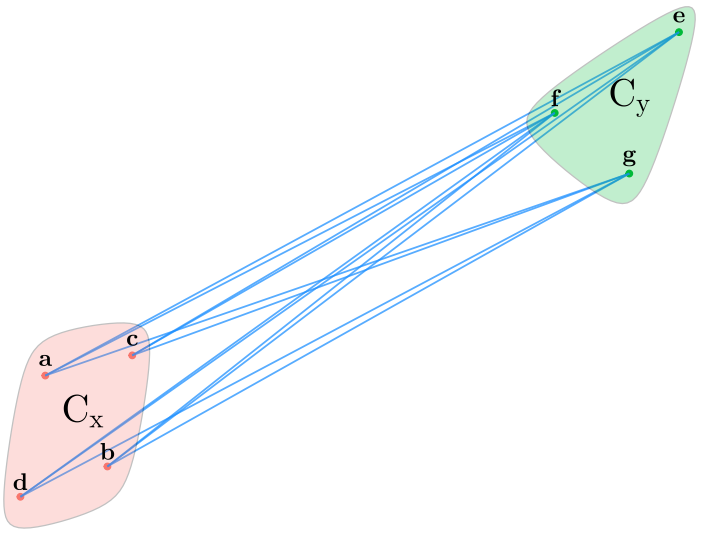
Uvažujme libovolnou vzdálenost d. Metoda průměrné vazby definuje vzdálenost dvou shluků C_x, C_y jako průměrnou vzdálenost všech dvojic objektů $\mathbf{x} \in C_x, \mathbf{y} \in C_y$.

$$\delta(C_x, C_y) = \frac{1}{|C_x||C_y|} \sum_{\mathbf{x} \in C_x} \sum_{\mathbf{y} \in C_y} d(\mathbf{x}, \mathbf{y})$$

Sloučení shluků C_x, C_y vede ke změně matice vzdáleností, přičemž vzdálenost sloučených shluků $C_x \cup C_y$ od shluku C_z určíme dle vztahu

$$\delta(C_x \cup C_y, C_z) = \frac{|C_x|\delta(C_x, C_z) + |C_y|\delta(C_y, C_z)}{|C_x| + |C_y|}.$$

Metoda průměrné vazby je kompromisem mezi metodou nejbližšího a nejvzdálenějšího souseda. Vytváří shluky všech tvarů, typicky kompaktní, kruhové. Ilustraci zjištění vzdálenosti shluků metodou průměrné vazby (bez zprůměrování) můžeme vidět na obrázku 3.8.



Obrázek 3.8: Ilustrace zjištění vzdálenosti shluků C_x, C_y metodou průměrné vazby

Pro ukázku konkrétní aplikace metody průměrné vazby si ukážeme jak určit vzdálenost pro shluky C_x, C_y. K výpočtu využijeme ilustraci vzdálenosti na obrázku 3.8. Dostáváme:

$$\begin{aligned}\delta(C_x, C_y) &= \frac{1}{|C_x||C_y|} \sum_{\mathbf{x} \in C_x} \sum_{\mathbf{y} \in C_y} d_e(\mathbf{x}, \mathbf{y}) \\ &= \frac{1}{4 \cdot 3} \left(d_e(\mathbf{a}, \mathbf{e}) + d_e(\mathbf{a}, \mathbf{f}) + d_e(\mathbf{a}, \mathbf{g}) + \cdots + d_e(\mathbf{d}, \mathbf{e}) + d_e(\mathbf{d}, \mathbf{f}) + d_e(\mathbf{d}, \mathbf{g})\right) \doteq 2{,}71.\end{aligned}$$

Centroidová metoda

Uvažujme Euklidovskou vzdálenost d_e. Centroidová metoda definuje vzdálenost dvou shluků C_x, C_y jako Euklidovskou vzdálenost středu shluků, tzv. centroidů $\bar{C}_x, \bar{C}_y$.

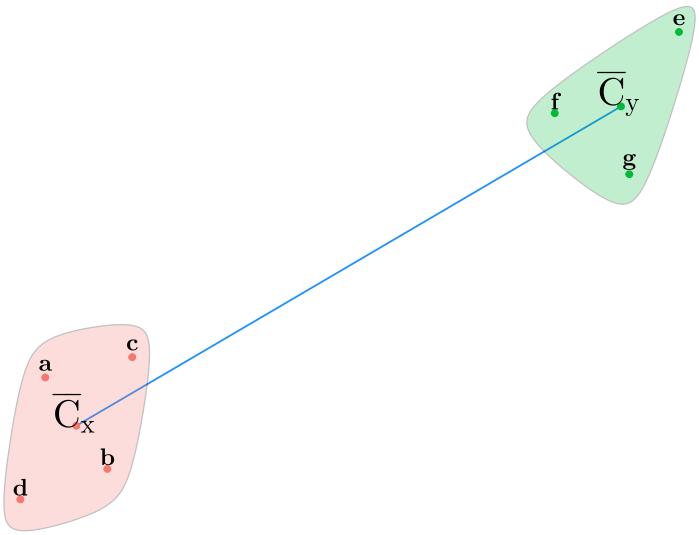
$$\bar{C}_x = \frac{1}{|C_x|} \sum_{\mathbf{x} \in C_x} \mathbf{x}, \quad \bar{C}_y = \frac{1}{|C_y|} \sum_{\mathbf{y} \in C_y} \mathbf{y}$$

$$\delta(C_x, C_y) = d_e\left(\bar{C}_x, \bar{C}_y\right)$$

Sloučení shluků C_x, C_y vede ke změně matice vzdáleností, přičemž vzdálenost sloučených shluků $C_x \cup C_y$ od shluku C_z určíme dle vztahu

$$\delta(C_x \cup C_y, C_z) = \sqrt{\frac{|C_x| \cdot \delta(C_x, C_z) + |C_y| \cdot \delta(C_y, C_z)}{|C_x| + |C_y|} - \frac{|C_x||C_y| \cdot \delta(C_x, C_y)}{(|C_x| + |C_y|)^2}}.$$

Je vhodné poznamenat, že software R pracuje se čtvercem vzdáleností d_e^2, tím se ovšem na principu shlukování nic zásadního nezmění. Centroidová metoda je jednoduchá, intuitivní a lehce implementovatelná. Vytváří shluky různých tvarů. Ilustraci zjištění vzdálenosti shluků centroidovou metodou můžeme vidět na obrázku 3.9.



Obrázek 3.9: Ilustrace zjištění vzdálenosti shluků C_x, C_y centroidovou metodou

Problém centroidové metody vzniká při slučování dvou shluků s rozdílnou velikostí. Při sloučení je nový centroid $\bar{C}_{xy}$ blíže shluku s větší velikostí. Pro $|C_x| \gg |C_y|$ se nový centroid téměř nepohne od původního centroidu $\bar{C}_x$.

Dalším problémem této metody je tzv. inverze, kde při dalším iteračním kroku je vzdálenost slučovaných shluků menší, než byla vzdálenost slučovaných shluků v kroce předchozím. I přes tyto nedostatky je tato metoda v praxi často používaná.

Obecně je centroid reprezentantem shluku, v praxi se můžeme setkat také s tzv. medoidem, což je objekt shluku s nejmenší celkovou vzdáleností k objektům uvnitř shluku. My budeme nadále pracovat pouze s centroidy.

Pro ukázku konkrétní aplikace centroidové metody si ukážeme jak určit vzdálenost pro shluky C_x, C_y z modelového příkladu. Nejdříve určíme centroidy $\bar{C}_x, \bar{C}_y$.

$$\bar{C}_x = \frac{1}{|C_x|} \sum_{\mathbf{x} \in C_x} \mathbf{x} = \frac{1}{|C_x|} \left(\sum_{\mathbf{x} \in C_x} x_1, \sum_{\mathbf{x} \in C_x} x_2 \right) = (1{,}0750, 1{,}0625)$$

$$\bar{C}_y = \frac{1}{|C_y|} \sum_{\mathbf{y} \in C_y} \mathbf{y} = \frac{1}{|C_y|} \left(\sum_{\mathbf{y} \in C_x} y_1, \sum_{\mathbf{y} \in C_x} y_2 \right) \doteq (3{,}2667, 2{,}6333)$$

K výpočtu výsledné vzdálenosti shluků C_x, C_y využijeme ilustraci vzdálenosti na obrázku 3.9 a spočtených centroidů. Dostáváme:

$$\delta(C_x, C_y) = d_e\left(\bar{C}_x, \bar{C}_y\right) = \sqrt{\left(\bar{C}_{y_1} - \bar{C}_{x_1}\right)^2 + \left(\bar{C}_{y_2} - \bar{C}_{x_2}\right)^2} \doteq 2{,}70. \quad [\text{v R: } (2{,}70)^2]$$

Wardova metoda

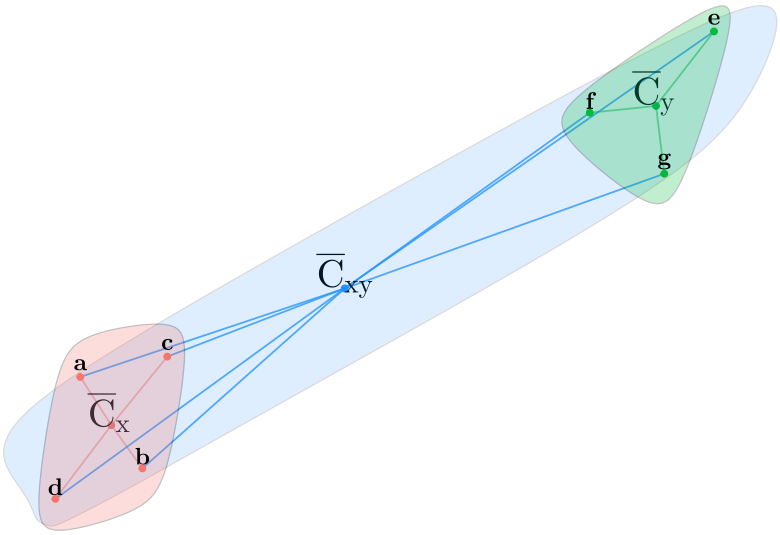
Uvažujme Euklidovskou vzdálenost d_e a středy shluků $\bar{C}_x, \bar{C}_y$. Centroid shluku, který by vznikl sloučením $C_x \cup C_y$ označme $\bar{C}_{xy}$. Wardova metoda definuje vzdálenost dvou shluků C_x, C_y jako odmocninu přírustku rozptylu po sloučení dvou shluků C_x, C_y vynásobenou konstantou.

$$\begin{aligned}\delta(C_x, C_y) &= \sqrt{2} \cdot \sqrt{\sum_{\mathbf{z} \in C_x \cup C_y} d_e^2\left(\mathbf{z}, \bar{C}_{xy}\right) - \sum_{\mathbf{x} \in C_x} d_e^2\left(\mathbf{x}, \bar{C}_x\right) - \sum_{\mathbf{y} \in C_y} d_e^2\left(\mathbf{y}, \bar{C}_y\right)} \\ &= \sqrt{\frac{2|C_x||C_y|}{|C_x| + |C_y|}} d_e(C_x, C_y)\end{aligned}$$

Konstanta $\sqrt{2}$ je nutná k zachování Euklidovské vzdálenosti mezi objekty, pokud jsou oba shluky tvořeny jediným objektem [8]. Sloučení shluků C_x, C_y vede ke změně matice vzdáleností, přičemž vzdálenost sloučených shluků $C_x \cup C_y$ od shluku C_z určíme dle vztahu

$$\delta(C_x \cup C_y, C_z) = \sqrt{\frac{(|C_x| + |C_z|)\delta(C_x, C_z) + (|C_y| + |C_z|)\delta(C_y, C_z) - |C_z|\delta(C_x, C_y)}{|C_x| + |C_y| + |C_z|}}.$$

Wardova metoda vytváří malé, kompaktní kruhové shluky s podobným počtem objektů. Okrajové objekty shluku jsou roztroušeny relativně volně. Ilustraci zjištění vzdálenosti shluků Wardovou metodou (bez odmocniny a vynásobení konstantou) můžeme vidět na obrázku 3.10.



Obrázek 3.10: Ilustrace zjištění vzdálenosti shluků C_x, C_y Wardovou metodou

Pro ukázku konkrétní aplikace metody Wardovy metody si ukážeme jak určit vzdálenost pro shluky C_x, C_y. K výpočtu využijeme ilustraci vzdálenosti na obrázku 3.10 a již vypočtených

centroidů. Dostáváme:

$$\delta(C_x, C_y) = \sqrt{\frac{2|C_x||C_y|}{|C_x| + |C_y|}} d_e(C_x, C_y)$$
$$\doteq 1{,}85 \cdot 2{,}7 = 5{,}00.$$

Přehled základních hierarchických metod

Ukázali jsme si základní metody hierarchického shlukování. Označme $\delta_{xy} = \delta(C_x, C_y)$. Poté pro přehlednost a rychlou referenci metod hierarchického shlukování můžeme využít tabulky 3.2.

Tabulka 3.2: Přehled metod hierarchického shlukování

Metoda	Vzdálenost shluků $\delta(C_x, C_y)$	Úprava vzdálenosti po sloučení shluků $\delta(C_x \cup C_y, C_z)$
Nejbliž. s.	$\min_{\mathbf{x}\in C_x, \mathbf{y}\in C_y} d(\mathbf{x}, \mathbf{y})$	$\min(\delta_{xz}, \delta_{yz})$
Nejvzdál. s.	$\max_{\mathbf{x}\in C_x, \mathbf{y}\in C_y} d(x, y)$	$\max(\delta_{xz}, \delta_{yz})$
Průměr. v.	$\frac{1}{\lvert C_x\rvert\lvert C_y\rvert} \sum_{\mathbf{x}\in C_x} \sum_{\mathbf{y}\in C_y} d(\mathbf{x}, \mathbf{y})$	$\frac{\lvert C_x\rvert\delta_{x,z} + \lvert C_y\rvert\delta_{y,z}}{\lvert C_x\rvert + \lvert C_y\rvert}$
Centroid.	$d_e\left(\bar{C_x}, \bar{C_y}\right)$	$\sqrt{\frac{\lvert C_x\rvert\delta_{x,z} + \lvert C_y\rvert\delta_{y,z}}{\lvert C_x\rvert + \lvert C_y\rvert} - \frac{\lvert C_x\rvert\lvert C_y\rvert\delta_{xy}}{(\lvert C_x\rvert + \lvert C_y\rvert)^2}}$
Wardova	$\sqrt{\frac{2\lvert C_x\rvert\lvert C_y\rvert}{\lvert C_x\rvert + \lvert C_y\rvert}} d_e(C_x, C_y)$	$\sqrt{\frac{(\lvert C_x\rvert + \lvert C_z\rvert)\delta_{xz} + (\lvert C_y\rvert + \lvert C_z\rvert)\delta_{yz} - \lvert C_z\rvert\delta_{xy}}{\lvert C_x\rvert + \lvert C_y\rvert + \lvert C_z\rvert}}$

Volba metody závisí na konkrétním problému. Typicky se porovná několik metod a na základě vizualizace, například dendrogramu, se vyberou metody produkující zajímavé výsledky. Volbu metody také určuje tvar shluků, který si představujeme pro daný problém jako optimální.

3.3.2 Implementace v software R

Pro ukázku, jak lze hierarchické shlukování provést s využitím software R, implementujeme hierarchické shlukování aglomerativní s metodou nejbližšího a nejvzdálenějšího souseda. Algoritmus aglomerativního shlukování následně aplikujeme na dataset Iris.

Hlavní princip algoritmu spočívá v postupném slučování nejbližších shluků. Algoritmus postupně iteruje a mění matici vzdáleností D. K ukončení dochází při dosažení počtu shluků k, který hledáme. Jednu iteraci algoritmu můžeme shrnout ve čtyřech základních krocích.

(1) Nalezení nejbližších shluků C_x, C_y

(2) Vytvoření shluku $C_x \cup C_y$

(3) Přepočet matice vzdálenosti D

Vstupem algoritmu je datová matice (data), metoda shlukování (vazba), vzdálenost d a již zmíněný počet shluků k. Výstupem je vektor, jehož složky odpovídají přiřazení původních objektů do nalezených shluků.

```
## data: vstupni matice dat,  k: pocet hledanych shluku,
## vazba: (single = nejblizsi, complete = nejvzdalenejsi
## vzdalenost: (euclidean, manhattan, maximum)
shluk_hier <- function(data, k, vazba, vzdalenost)
{
  vzdal = as.matrix(dist(data,method=vzdalenost,diag=T,upper=T))
  diag(vzdal) = Inf; N = nrow(vzdal); shluky = 1:N
  vazba = switch(vazba, single = rowMins, complete = rowMaxs)

  for(i in 1:(N-k))
  {
    # Nejblizsi shluky
    nejbliz = which(vzdal == min(vzdal), arr.ind = TRUE)[1, ]
    sousedi = cbind(vzdal[, nejbliz[1]], vzdal[, nejbliz[2]])

    # Slouceni shluku
    nazvy = as.numeric(row.names(vzdal))
    shluky[which(shluky == nazvy[nejbliz[2]])] = nazvy[nejbliz[1]]

    # Vzdalenosti po slouceni
    nove_vzdal = vazba(sousedi, value = T)
    nove_vzdal[nejbliz[1]] = nove_vzdal[nejbliz[2]] = Inf

    # Nahrazeni vzdalenosti
    vzdal[nejbliz[1], ] = nove_vzdal
```

```
    vzdal[, nejbliz[1]] = nove_vzdal
    vzdal = vzdal[-nejbliz[2], -nejbliz[2]]
  }
  return (prejmenuj(shluky))
}
## výstup: vektor přiřazení objektů do shluků.  např. (1, 3, 2, 3 ...)
## výsledné shluky jsou přejmenovány vzestupně. např. (1, 2, 3, 2 ...)
```

Pro dataset Iris je výsledek implementované metody hierarchického shlukování shodný s výsledkem shlukování funkce *hclust* z balíčku *stats*, což si můžeme ověřit v příloze. Nadále budeme pracovat s metodou *hclust*, důvodem je větší množství voleb, vyšší rychlost a výstup umožňující sestavit odpovídající dendrogram.

Označme $\hat{C}$ jako výsledný rozklad a C jako rozklad referenční. Referenční rozklad $\hat{C}$ máme v tomto případě k dispozici, jelikož jedním z atributů datasetu *Iris* je druh kosatce. Je vhodné si uvědomit, že referenční rozklad nemáme v praxi téměř nikdy a proto následující analýza slouží pouze pro ilustrativní účely.

Výsledný rozklad C je nutno spárovat s rozkladem referenčním $\hat{C}$. Spárování provedeme pomocí kontigenční tabulky zobrazující počty objektů shluků v C s počty objektů shluků v $\hat{C}$. Sloupce tabulky (nalezené shluky) seřadíme tak, aby shluku odpovídající druh ležel v odpovídajícím řádku. Porovnání a spárování nalezených shluků (Euklidovská vzdálenost, průměrná vazba) s druhy kosatců vidíme v tabulce 3.3.

Tabulka 3.3: Spárování rozkladu (Euklidovská vzdálenost, průměrná vazba) s druhy kosatců (v závorce uvedeny řádkové relativní četnosti odpovídajícího druhu)

Druh	Shluk			Celkem
	C_1	C_2	C_3	
Setosa	50 (1,00)	0	0	50
Versicolor	0	50 (1,00)	0	50
Virginica	0	14 (0,38)	36 (0,72)	50
Celkem	50	64	36	150

V tabulce 3.3 můžeme pozorovat správné rozlišení kosatců Setosa a Versicolor, jimž odpovídají shluky C_1 a C_3. Také si můžeme všimnout špatného přiřazení 14 květů Virginica do shluku C_2 odpovídajícímu kosatcům Versicolor. Důvodem je již zmiňovaná podobnost květu Versicolor a Virginica, kterou jsme zkoumali v kapitole 2 pomocí explorační analýzy.

Nyní využijeme spárovaných shluků v tabulce 3.3 k výpočtu kvality shlukování. Výslednou kvalitu shlukování získáme jako podíl počtu správně přiřazených objektů k celkovému počtu objektů. V našem případě $^{136}/_{150} = 0{,}907$. Obdobně byla vyhodnocena kvalita shlukování dalšími studovanými metodami s Euklidovskou, maximální a Manhattenskou vzdáleností. Porovnání metod hierarchického shlukování na základě výše uvedeného způsobu vidíme v tabulce 3.4.

Tabulka 3.4: Kvalita hierarchického shlukování v závislosti na metodě shlukování a zvolené vzdálenosti

Vzdálenost	Metoda shlukování				
	Nejbližší. s.	Nejvzdálenější. s.	Průměrné v.	Centroidova	Wardova
Euklidovská	0,680	0,840	0,907	0,907	0,893
Maximální	0,680	0,820	0,733	0,740	0,907
Manhattanská	0,673	0,893	0,900	0,693	0,887

Z tabulky 3.4 můžeme pozorovat, že kvalita hiearchického shlukování s Wardovou metodou a metodou nejbližšího souseda je v tomto případě téměř nezávislá na volbě vzdálenosti.

Dalším z pozorování je neuspokojivá kvalita shlukování metodou nejbližšího souseda oproti zbylým volbám. Za kvalitní shlukování v našem případě považujeme shlukování, jehož kvalita je $\geq 0{,}9$ (hierarchické shlukování s Euklidovskou vzdáleností a průměrnou, centroidovou vazbou; maximální vzdáleností a Wardovou vazbou; Manhattenskou vzdáleností a průměrnou vazbou).

Nadále budeme pracovat s hierarchickým shlukováním s Euklidovskou vzdáleností. Důvodem je vysoká kvalita, kterou dosahují vazby s touto vzdáleností a intuitivní porozumnění. Tato volba je ovšem subjektivní.

3.3.3 Dendrogram

Hierarchické shlukování můžeme vizualizovat pomocí stromové struktury, nazývané dendrogramem. S dendrogramem jsme se již setkali při demonstraci metody nejbližšího souseda v kapitole 3.3.1. Vzdálenosti, ve kterých se sloučily shluky při hierarchickém shlukování, jsou uzly dendrogramu.

Pro každou z výše srovnávaných metod shlukování s Euklidovskou vzdáleností (vyjma centroidové metody, kvůli podobnosti s metodou průměrné vazby, viz tabulka 3.4) sestavíme spárování nalezených shluků s druhy kosatců. Výsledné shluky označíme C_1, C_2, C_3.

Jednotlivé, námi označené shluky, jsou shluky s maximální shodou v počtu primárního přiřazení druhů kosatců (C_1 obsahuje primárně kosatce Setosa, C_2 Versicolor a C_3 Virginica). Výsledné spárování můžeme vidět v tabulce 3.5.

Tabulka 3.5: Posouzení kvality shlukování (pomocí spárovaných shluků) v závislosti na metodě shlukování s Euklidovskou vzdáleností

Nejbližší. s.

Druh	Shluk			Celkem
	C_1	C_2	C_3	
Setosa	50	0	0	50
Versicolor	0	50	0	50
Virginica	0	48	2	50
Celkem	50	98	2	150

Nejvzdálenější. s.

Druh	Shluk			Celkem
	C_1	C_2	C_3	
Setosa	50	0	0	50
Versicolor	0	27	23	50
Virginica	0	1	49	50
Celkem	50	28	72	150

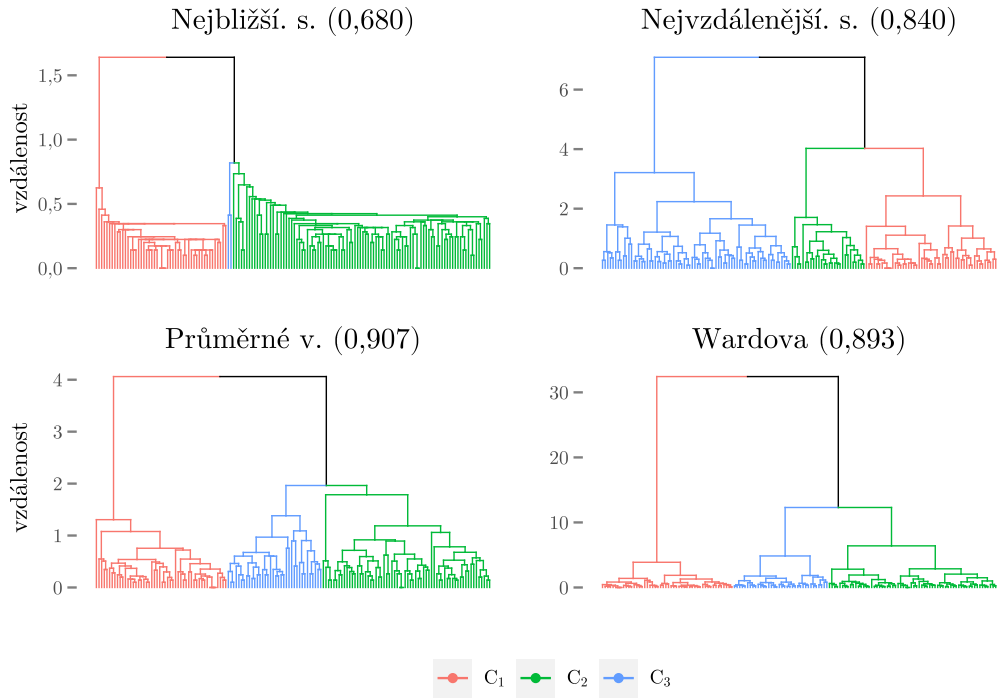
Průměrné v.

Druh	Shluk			Celkem
	C_1	C_2	C_3	
Setosa	50	0	0	50
Versicolor	0	50	0	50
Virginica	0	14	36	50
Celkem	50	64	36	150

Wardova

Druh	Shluk			Celkem
	C_1	C_2	C_3	
Setosa	50	0	0	50
Versicolor	0	49	1	50
Virginica	0	15	35	50
Celkem	50	64	36	150

Odpovídající dendrogramy výše srovnaných metod shlukování s výslednou kvalitou shlukování, získanou s využitím tabulky 3.5, lze vidět na obrázku 3.11.



Obrázek 3.11: Dendrogramy reprezentující hiearchické shlukování s Euklidovskou vzdáleností a různými vazbami (v závorce je uvedena kvalita shlukování).

Využijeme tabulky 3.5 k posouzení kvality nalezených rozkladů shlukování daných metod. Například, u shlukování metodou nejbližšího souseda nedošlo k rozlišení kosatců Versicolor a Virginica, kde převážná většina kosatců Virginica byla přiřazena do shluku C_2. Dále u shlukování metodou nejvzdálenějšího souseda došlo k přiřazení téměř poloviny kosatců Versicolor do shluku C_3, tvořeného primárně kosatci Virginica.

Zbylé dvě metody dosáhly téměř stejného přiřazení, kde část kosatců Virginica byla přiřazena do shluku C_2, tvořeného převážně kosatcem druhu Versicolor. I přes částečně chybné přiřazení dosáhly tyto metody vysoké kvality, viz obrázek 3.11 a tabulka 3.4.

Všimněme si rovněž, že Wardova metoda vytvořila očekávaně rovnoměrně rozložené shluky. Tyto shluky jsou podobné shlukům vytvořeným shlukováním s průměrnou vazbou.

Dendrogramy jsou užitečnou pomůckou při zkoumání postupu hierarchického shlukování. Poskytují nám přehlednou vizualizovanou představu o postupu zvoleného algoritmu.

3.3.4 Shlukování bez reference

Po celou dobu jsme pracovali s referenčním rozkladem $\hat{C}$, díky kterému jsme mohli posuzovat kvalitu shlukování a získat tak představu o chování metod hierarchického shlukování.

Představme si nyní, že jsme biologem a našim úkolem je popsat druhy rostlin. Hledáme tedy shluky a díky jejich typickým hodnotám atributů můžeme následně definovat nové druhy rostlin a jejich specifikaci. V tomto případě žádný referenční rozklad k dispozici nemáme, jak tedy postupovat?

V praxi se typicky provede několik shlukování s různými metodami a díky externí informaci, kterou nám poskytne profesionál v dané problematice, se zvolí vhodná metoda. My profesionála k dispozici nemáme a musíme se uchýlit k určitým předpokladům na výsledný rozklad.

Předpokládejme tedy, že člověk, který zaznamenával údaje o kosatcích, rozeznal jednotlivé druhy, díky čemuž je počet vzorků jednotlivých druhů téměř stejný. Díky této informace můžeme zvolit konkrétní metodu a vzdálenost, kupříkladu Wardovu s Euklidovskou vzdáleností.

Statistická odlišnost shluků

Vraťme se nyní k námi zvoleným parametrům shlukování, tj. Wardova metoda, Euklidovská vzdálenost a hledejme tři shluky, které opět označíme C_1, C_2, C_3 (biolog má pocit, že jde o tři druhy kosatců).

Jednou z možností jak porovnat shluky, je srovnáním statistických odlišností jednotlivých atributů dle nalezených shluků s využitím testu ANOVA, resp. Kruskalova-Wallisova testu.

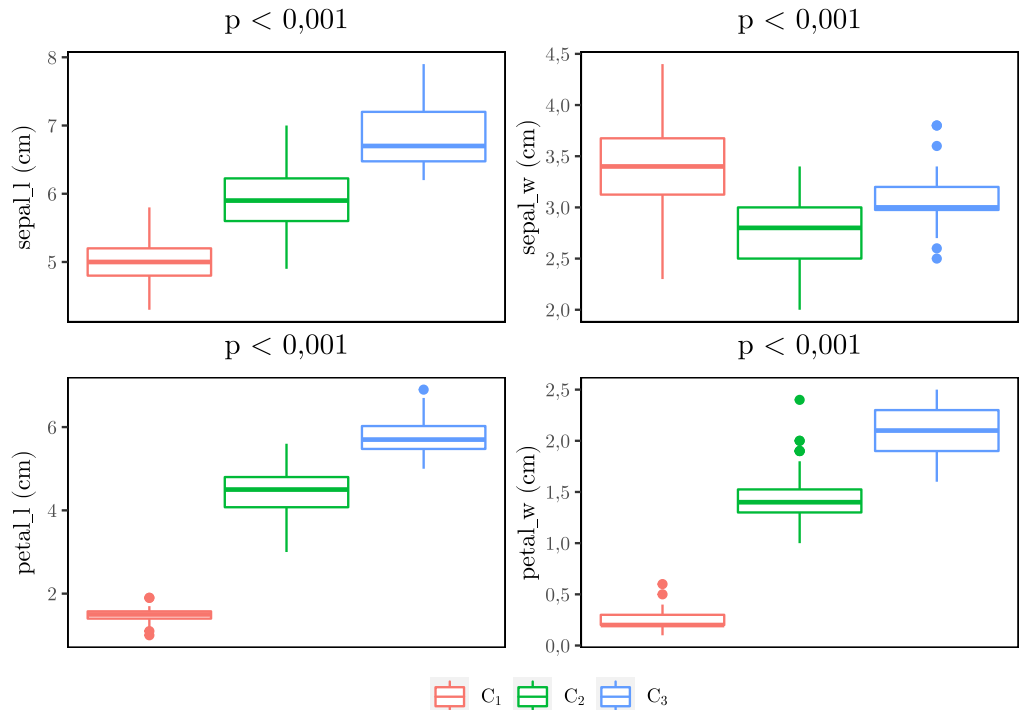
Statistický test je rozhodovacím procesem v němž se přikláníme k nulové (H_0), nebo alternativní (H_A) hypotéze. O nezamítnuti H_0, nebo zamítnutí H_0 ve prospěch H_A se rozhoduje na základě p-hodnoty (p) a zvolené hladiny významnosti α (standardně $\alpha = 0,05$), kde p-hodnota je nejnižší hladinou významnosti, na níž se zamítá nulová hypotéza [11].

Analýza rozptylů (ANOVA) je testem shody středních hodnot. Nulová hypotéza H_0 předpokládá shodu středních hodnot atributů jednotlivých shluků. H_A je logickou negací H_0, tj. lze předpokládat, že existuje alespoň jedna dvojice shluků, pro kterou nejsou střední hodnoty atributu stejné. ANOVA předpokládá nezávislost, normalitu každého z výběru a homoskedasticitu (shodu rozptylů).

Kruskalův-Wallisův (K-W) test je neparametrickou alternativou testu ANOVA, využívá se pro nezávislá data při nesplnění předpokladů ANOVA testu. Nulová hypotéza H_0 předpokládá shodu mediánů atributů v jednotlivých shlucích. Alternativní hypotéza H_A předpokládá existenci alespoň jedné dvojice shluků, pro kterou nejsou mediány atributů stejné. Podrobnější informace o testech a testování hypotéz lze nalézt v [11].

Nejdříve zjistíme, zda lze rozdělení atributů kosatců jednotlivých shluků považovat za normální, což můžeme provést například pomocí Shapirova-Wilkova testu normality. K ověření homoskedasticity můžeme využít Bartlettův test, příp. Leveneho test.

Bylo zjištěno (testy na 5% hladině významnosti), že požadavky ANOVA testu splňuje pouze atribut šířky kališního lístku. Protože převážná většina atributů nesplnila požadavky na provedení ANOVA testu, budeme pro ucelenost postupu pracovat u všech atributů s K-W testem ($\alpha = 0{,}05$). Krabicové grafy jednotlivých atributů (s odpovídající p-hodnotou K-W testu) pro srovnání shluků C_1, C_2, C_3 jsou vidět na obrázku 3.12.



Obrázek 3.12: Krabicové grafy atributů shluků C_1, C_2, C_3 společně s odpovídající p hodnotou K-W testu

Z obrázku 3.12 (a výsledků K-W testů) lze pozorovat statisticky významnou odlišnost jednotlivých atributů mezi shluky C_1, C_2, C_3.

Nyní potřebujeme zjistit, mezi kterými shluky jsou v jednotlivých atributech statisticky významné rozdíly. K tomuto využijeme post-hoc analýzu, konkrétně Dunnové test (vícenásobné porovnání) opět na 5% hladině významnosti.

Testy bylo zjištěno, že se všechny shluky mezi sebou z hlediska rozdělení jednotlivých atributů statisticky významně liší, tj. nebyly nalezeny homogenní podskupiny shluků. Jako hypotetičtí biologové můžeme tedy konstatovat, že se jedná o odlišné druhy. Tyto druhy můžeme nazvat třeba Setosa (C_1), Versicolor (C_2) a Virginica (C_3). Charakteristiky atributů nalezených shluků můžeme vidět v tabulce 3.6.

Tabulka 3.6: Výběrové charakteristiky jednotlivých atributů nalezených shluků, chceme-li druhů kosatců, Setosa (C_1), Versicolor (C_2) a Virginica (C_3)

sepal_l (cm)

	Setosa	Versicolor	Virginica
min	4,3	4,9	6,2
Q_1	4,80	5,60	6,48
průměr	5,01	5,92	6,87
medián	5,00	5,90	6,70
Q_3	5,20	6,23	7,20
max	5,8	7,0	7,9
sm. odch.	0,36	0,48	0,50
var. koef.	0,07	0,08	0,07
šikmost	0,12	−0,01	0,62
špicatost	−0,35	−0,16	−0,83

sepal_w (cm)

	Setosa	Versicolor	Virginica
min	2,3	2,0	2,5
Q_1	3,13	2,50	2,98
průměr	3,42	2,75	3,09
medián	3,40	2,80	3,00
Q_3	3,68	3,00	3,20
max	4,4	3,4	3,8
sm. odch.	0,39	0,30	0,29
var. koef.	0,11	0,11	0,09
šikmost	0,10	−0,28	0,58
špicatost	0,69	−0,31	0,65

petal_l (cm)

	Setosa	Versicolor	Virginica
min	1,0	3,0	5,0
Q_1	1,40	4,08	5,48
průměr	1,46	4,42	5,77
medián	1,50	4,50	5,70
Q_3	1,58	4,80	6,03
max	1,9	5,6	6,9
sm. odch.	0,18	0,53	0,48
var. koef.	0,12	0,12	0,08
šikmost	0,07	−0,47	0,61
špicatost	0,81	−0,09	−0,24

petal_w (cm)

	Setosa	Versicolor	Virginica
min	0,1	1,0	1,6
Q_1	0,20	1,30	1,90
průměr	0,24	1,43	2,11
medián	0,20	1,40	2,10
Q_3	0,30	1,53	2,30
max	0,6	2,4	2,5
sm. odch.	0,11	0,30	0,25
var. koef.	0,44	0,20	0,12
šikmost	1,16	0,71	−0,19
špicatost	1,30	0,59	−0,93

K vyhodnocení nalezených druhů kosatců můžeme využít obrázku 3.12 a tabulky 3.6. Vidíme, že statisticky významně nejmenší okvětní lístky (šířka i délka) mají kosatce Setosa, nevětší pak kosatce Virginica. Průměrná šířka kališních lístků kosatce Setosa je nejvyšší, kdežto u kosatců Versicolor nejnižší.

Nalezení statisticky odlišných druhů kosatců nám jako hypotetickým biologům umožnilo

zavést nové druhy kosatců. Každý z těchto druhů má své typické charakteristiky (viz tabulka 3.6). Na základě těchto charakteristik můžeme rozpoznat tyto druhy a vytvářet si tak další představu o jejich vlastnostech.

3.4 Nehierarchické shlukování

V této kapitole se (alespoň stručně) budeme věnovat nehierarchickému shlukování, které je tvořeno skupinou metod pracujících s nějakým vstupním parametrem a kritériem kvality. Celé nehierarchické shlukování můžeme chápat jako iterativní optimalizaci počátečního rozkladu C.

Vstupním parametrem může být například počet shluků, jako při hierarchickém shlukování. Počet shluků samozřejmě není jedinou možností, některé metody pracují s parametry, které chytrým způsobem počet shluků nahradí. U metody DBSCAN například pomocí poloměru shluku a počtu objektů v tomto shluku. Můžeme říci, že parametry vytváří restrikci na rozklad C.

Optimalizace rozkladu C probíhá typicky na základě maximalizace kvality shlukování, chceme-li minimalizace odlišnosti mezi objekty v rámci jednotlivých shluků. Kritéria kvality nejsou pevně definována a proto existuje mnoho alogritmů, které mají odlišný pohled na „správný“ rozklad C.

Metod nehierarchického shlukování je mnoho, například: K-means, Mean-shift, DBSCAN, Expectation-maximization, Affinity propagation, aj. My se opět podíváme na základní z nich: K-means a DBSCAN. Pěkné shrnutí metod poskytuje např. [7], [12].

3.4.1 K-means

Jedním z nejpopulárnějších a nejvíce využívaných algoritmů shlukování je metoda k-means (nebo *k-průměrů*, pro obecnou znalost a používání i v českých textech se nadále budeme držet označení metody k-means). Důvodem popularity této metody je její snadná implementace a vysoká rychlost. Metoda k-means se považuje za základní metodu nehierarchického shlukování. Jejím popisem sa zabývá např. [9], [13], [14].

Jediným vstupním parametrem této metody je očekávaný počet shluků k. Cíl metody je intuitivní, hledá takových k shluků, jejichž vnitroskupinová podobnost je maximální [5], tj. součet podobností objektů uvnitř shluku vzhledem k jeho centroidu je maximální, čehož dosahuje minimalizací vnitroshlukového součtů čtverců rozkladu. Analýzou součtu čtverců rozkladu se detailněji zabýváme v kapitole 3.5.1.

Formulace

Mějme rozklad C o k shlucích $C_1, ..., C_k$. Zaveďme tzv. vnitroshlukový součet čtverců SS_W (viz kapitola 3.5.1), tj. součet čtverců vzdálenosti objektů jednotlivých shluků od jejich centroidů.

$$SS_W(C) = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d^2(\mathbf{x}, \bar{C_i})$$

Metoda k-means minimalizuje vnitroshlukový součet čtverců SS_W využívající Euklidovskou vzdálenost d_e. Uvažujme tedy, že máme rozklad C. Chceme upravit přiřazení objektů do shluků

v rozkladu tak, abychom v dalším kroku dosáhli nižšího vnitroshlukového součtu čtverců SS_W.

$$\min_{C_1,\ldots,C_k} SS_W(C)$$

Minimalizací daného výrazu dostáváme centroidy jednotlivých shluků [14].

$$\bar{C_i} = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$

Pro minimalizaci SS_W musíme tedy přepočítat centroidy shluků na jejichž základě se provede úprava rozkladu. Celý proces můžeme inicializovat dvěma způsoby. Buď můžeme vycházet z počátečního rozkladu (náhodného, nebo např. z jiné metody shlukování) a následně získat centroidy shluků tohoto rozkladu, nebo můžeme přímo zvolit k počátečních centroidů (typicky náhodných objektů). Princip metody můžeme poté shrnout ve třech základních krocích:

(1) Výpočet vzdálenosti objektů k centroidům $\bar{C_i}$
(2) Přeuspořaní objektů do shluků C_i
(3) Určení centroidů $\bar{C_i}$ nově vytvořených shluků C_i

Výpočet vzdáleností objektů k centroidům (1) využívá Euklidovské vzdálenosti. Přeuspořádání objektů do shluků (2) spočívá v přiřazení objektu do shluku, k jehož centroidu má nejblíže. Na základě nově získaného rozkladu přepočítáme centroidy (3) a celý proces opakujeme. Tímto způsobem pokračujeme dokud dochází ke změnám v rozkladu (resp. v centroidech).

Problém metody k-means spočívá v lokální konvergenci, kdy je výsledný rozklad a tedy i výsledná hodnota vnitroshlukového součtu čtverců SS_W determinována počáteční volbou centroidů. Náhodnou volbou počátečních reprezentantů (centroidů) můžeme tedy způsobit, že výsledné rozklady budou dosahovat nežádoucích (vysokých) hodnot SS_W. Částečným řešením je provést výše uvedený proces shlukování několikrát a zvolit nejlepší rozklad (s nejnižší hodnotou SS_W). Další možnost spočívá v rafinovanější volbě počátečních reprezentantů, kterou využívají komplexnější verze metody k-means jako např. k-means++.

Dalším nedostatkem metody k-means je její citlivost na odlehlá pozorování a rozpětí atributů, která je částečně způsobená užíváním Euklidovské vzdálenosti. Opět existují komplexnější verze metody k-means, jako např. metoda k-medoids, řešící problém odlehlých pozorování volbou medoidů jako reprezentantů shluků namísto centroidů. Poznamenejme ještě, že metoda k-means klade předpoklady na výsledný rozklad, jehož shluky by měly být tvořeny podobným počtem objektů.

Příklad

Ukážeme si princip výpočtu metody k-means na datech z tabulky 3.1. Hledejme dva shluky, $k = 2$. Zvolme tedy počáteční centroidy shluků, například: $\bar{C}_e = \mathbf{e}, \bar{C}_f = \mathbf{f}$. Uvažujme matici, která je tvořena Euklidovskými vzdálenostmi objektů ke všem centroidům, červeně označme nižší ze vzdáleností.

	a	**b**	**c**	**d**	**e**	**f**	**g**
C_e	3,064718	3,148412	2,720294	3,508917	0	0,640312	0,728011
C_f	2,427447	2,510478	2,080865	2,869233	0,640312	0	0,424264

Objekt přiřadíme do toho shluku, k jehož centroidu má objekt nižší vzdálenost. Tyto shluky označme C_{abcdfg} a C_e. Nyní přepočteme centroidy. Dostáváme $\bar{C}_{abcdfg} = [1{,}77, 1{,}53]$, $\bar{C}_e = [3{,}50, 3{,}00]$. Následně přepočteme vzdálenosti objektů od nově definovaných centroidů.

	a	**b**	**c**	**d**	**e**	**f**	**g**
C_{abcdfg}	0,851646	0,887299	0,487647	1,239072	2,270198	1,630276	1,712834
C_e	3,064718	3,148412	2,720294	3,508917	0	0,640312	0,728011

Označme nové shluky C_{abcd} a C_{efg}. Opět přepočteme centroidy. Získáváme $\bar{C}_{abcd} = [1{,}08, 1{,}06]$, $\bar{C}_{efg} = [3{,}27, 2{,}63]$ a přepočteme vzdálenosti.

	a	**b**	**c**	**d**	**e**	**f**	**g**
C_{abcd}	0,272947	0,241868	0,404969	0,427200	3,101613	2,461301	2,542833
C_{efg}	2,674192	2,730073	2,322456	3,095368	0,435660	0,271662	0,331361

Vidíme, že rozklad se nemění, tudíž nedochází ke změnám centroidů, metoda tedy končí. Zde bychom mohli vypočítat vnitroshlukový součet čtverců a celý proces opakovat znovu.

My ovšem takto postupovat nebudeme, jelikož díky tabulky 3.1 vidíme, že metoda k-means vytvořila správný rozklad i přes počáteční reprezentanty ležící ve stejném shluku.

Implementace v software R

I v tomto případě, pro demonstraci řešení v software R, implementujeme metodu k-means, kterou využijeme k nalezení rozkladu datasetu Iris. Počáteční volba centroidů je řešena náhodnou volbou k objektů vstupního souboru.

Výsledný rozklad hledáme iterativním způsobem, kdy na základě nově vypočtených centroidů nalezneme nový rozklad. Konec nastává, pokud nedochází ke změnám v rozkladu (resp. centroidech). V praxi je metoda navíc omezena, např. maximálním počtem iterací. Celý proces několikrát (p-krát) provedeme a výsledný rozklad volíme na základě nejmenšího vnitroshlukového součtu čtverců SS_W, který je implementován a popsán v kapitole 3.5.1.

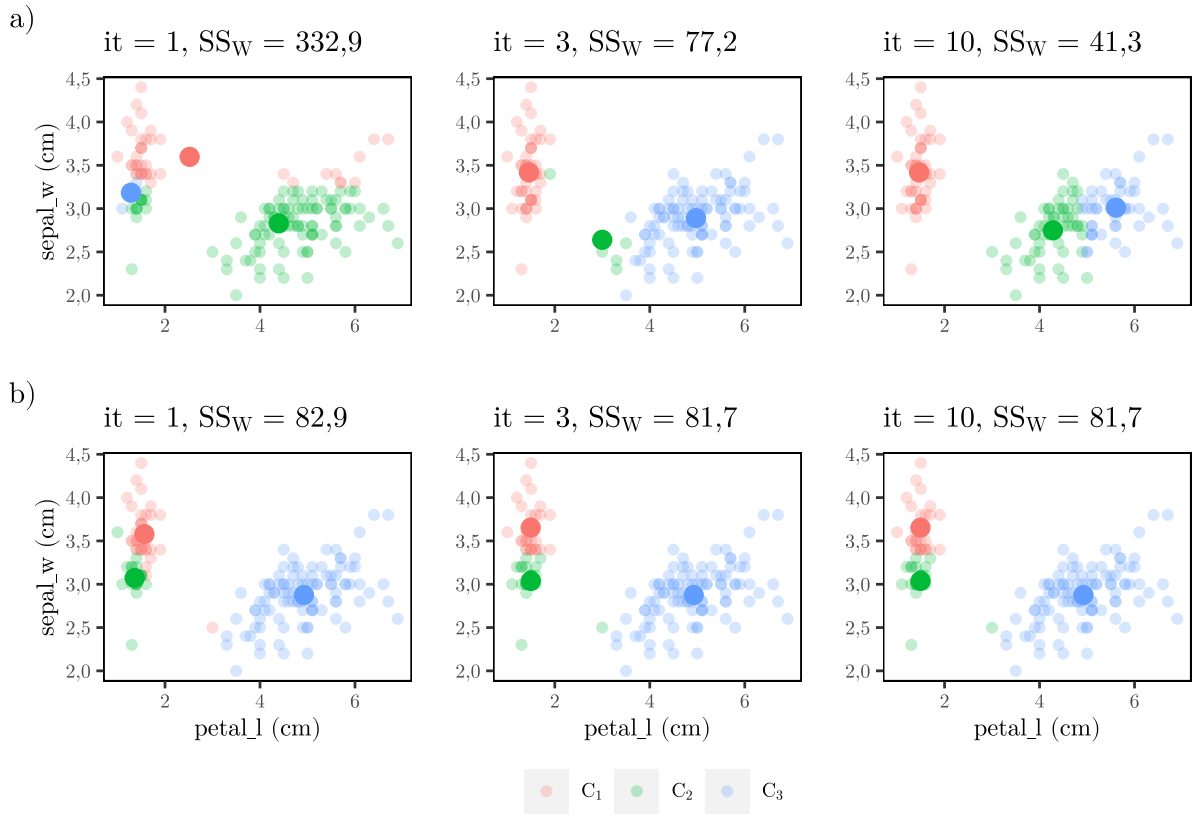
Vstupem algoritmu je datová matice (data), požadovaný počet shluků k a počet iterací p. Výstupem je vektor, jehož složky odpovídají přiřazení původních objektů do nalezených shluků.

```
## data: vstupní matice dat,  k: počet hledaných shluků, p: počet iterací
shluk_kmeans <- function(data, k, p)
{
  N = nrow(data); M = ncol(data); nej_rozptyl = Inf
  shluky = nej_shluky = rep(0, N)
  for(iter in 1:p)
  {
    # Náhodné centroidy
    centroidy = centroidy_prev = data[sample(N, k), ]; pohly_se = T

    while(pohly_se)
    {
      # Přepočet vzdáleností
      vzdal = apply(centroidy, 1, function(x)
        sqrt(rowSums(sweep(data, 2, x)^2)))
      # Přeuspořádání objektů
      shluky = rowMins(vzdal)
      # Přepočet centroidů
      centroidy = t(sapply(1:k, function(x)
        colMeans(data[which(shluky == x), ])))
      pohly_se = sum(centroidy != centroidy_prev) > 0;
      centroidy_prev = centroidy
    }

    # Výběr nejlepšího rozkladu na základě SS_W
    rozptyl = kvalita_vnitrni(data, shluky)
    if(rozptyl < nej_rozptyl) { nej_rozptyl = rozptyl; nej_shluky = shluky }
  }
```

```
  return(nej_shluky)
}
## výstup: vektor přiřazení objektů do shluků.  např. (1, 3, 2, 3 ...)
```

Pro dataset Iris je výsledek implementované metody k-means shodný s výsledkem shlukování funkcí *kmeans* z balíčku *stats*, což si můžeme ověřit v příloze. Nadále budeme pracovat s funkcí *kmeans*. Hlavním důvodem je v tomto případě vyšší rychlost algoritmu, flexibilita a stabilita (vůči možnostem jako je například prázdný shluk).

Algoritmus využijeme k zobrazení postupu tvorby rozkladu atributů délky okvětního a šířky kališního lístku datasetu Iris v závislosti na počtu iterací (it) s odlišnou volbou počátečních centroidů. Zmíněný postup můžeme vidět na obrázku 3.13.



Obrázek 3.13: Ilustrace problému počáteční volby centroidů metody k-means ($k = 3$) pro atributy délky okvětního a šířky kališního lístku datasetu Iris

Využijeme obrázku 3.13 k analýze vlivu počáteční volby centroidů na nalezený rozklad. Zaměřme se na hodnoty SS_W. V případě (a) vidíme s rostoucím počtem iterací pokles celkového vnitroshlukového součtu čtverců SS_W, v případě (b) můžeme naopak vidět hodnotu SS_W téměř neměnnou.

Navíc, po porovnání výsledných hodnot SS_W vidíme, že druhá volba počátečních centroidů vedla k horšímu výsledku. Můžeme tedy konstatovat, že počáteční volba centroidů v druhém

případě vedla k lokálnímu minimu, které není optimální.

Metodu k-means využijeme k určení druhů kosatců, výslednou kvalitu shlukování získáme spárováním s druhy kosatců stejně jako u hierarchických metod. Výsledná kvalita je $0,893$ což je na úrovni hierarchických metod shlukování dosahujících nejvyšší kvality shlukování.

Jeden z dalších nedostatků metody k-means nastává při hledání rozkladu souboru tvořeného shluky s rozdílnou velikostí a hustotou, kdy se kolem každého z centroidů vytváří vlivem použití Euklidovské vzdálenosti hyperkoule, ve které leží všechny objekty daného shluku. Určité tvary shluků nemusí jít dobře zapouzdřit do této hyperkoule, např. shluk ve tvaru elipsoidu lze obalit koulí, ale poté bude výsledný shluk tvořen i objekty, které v elipsoidu nejsou.

Existuje mnoho algoritmů (např. k-means++, k-medoids), které se snaží eliminovat nedostatky metody k-means. Typicky se ovšem tyto metody nepoužívají, pokud očekáváme, že shluky jsou různorodé (jiná velikost a hustota), v opačném případě je metoda k-means a její alternativy vhodnou volbou pro rychlé nalezení rozkladu.

3.4.2 DBSCAN

Další z řady populárních nehierarchických metod shlukování je metoda Density-based spatial clustering of application with noise (DBSCAN), která je dobře popsána např. v [7], [14], [15]. DBSCAN jako jediná z metod shlukování dokáže oddělit shluky od tzv. šumu (analogie odlehlých pozorování), což můžeme vnímat jako její největší výhodu.

Metoda DBSCAN je jednou z mnoha metod využívajících tzv. hustotu objektů (počet objektů v určitém okolí daného objektu), díky níž eliminuje dvě z nevýhod metody k-means, a to již zmíněnou citlivost na odlehlá pozorování a nezávislost na počtu objektů v jednotlivých shlucích. Díky odlišnému přístupu ke shlukování je tato metoda vhodná pro ilustraci několika základních, pro nás nových myšlenek tvorby rozkladu.

Formulace

Mějme datovou matici X, minimální počet objektů h tvořících shluk a poloměr shluku ϵ.

Zaveďme funkci S_ϵ přiřazující objektu $\mathbf{x}$ množinu objektů ležící v jeho ϵ-okolí, tvořenou objekty jejichž vzdálenost je menší, nebo rovna poloměru ϵ.

$$S_\epsilon(\mathbf{x}) = \{\mathbf{y} \in X \,|\, d(\mathbf{x}, \mathbf{y}) \leq \epsilon\}$$

Na základě výše uvedené funkce můžeme objekt $\mathbf{x} \in X$ klasifikovat dle tabulky 3.7.

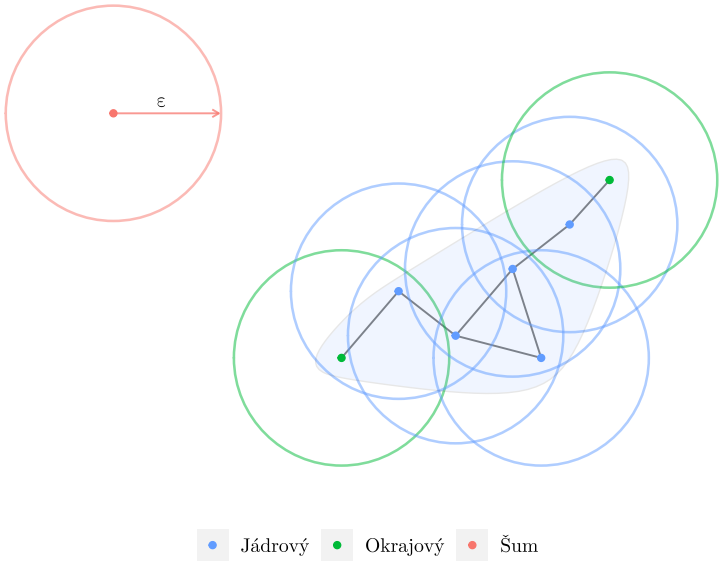
Tabulka 3.7: Klasifikace objektu na základě objektů ležících v jeho ϵ-okolí

Typ objektu	Vlastnosti objektu
Jádrový	Počet objektů v ϵ-okolí $S_\epsilon(\mathbf{x})$ objektu $\mathbf{x}$ je větší nebo roven h
Okrajový	Objekt není jádrový, ale leží v ϵ-okolí alespoň jednoho z jádrových objektů
Šum	Objekt není jádrový ani okrajový

Objekty přímo dosažitelné jádrovým objektem $\mathbf{x}$ nazýváme všechny objekty ležící v ϵ-okolí jádrového objektu $\mathbf{x}$. Za objekty nepřímo dosažitelné jádrovým objektem $\mathbf{x}$ pak označujeme objekty, pro něž existuje posloupnost přímo dosažitelných objektů z jádrového objektu $\mathbf{x}$. Shluk je pak tvořen jádrovým objektem a jím přímo a nepřímo dosažitelnými objekty. Každý shluk tedy obsahuje alespoň jeden jádrový objekt.

Dále definujme tzv. propojenost, kdy objekty $\mathbf{x}, \mathbf{y}$ označíme jako propojené, pokud existuje jádrový objekt $\mathbf{z}$, pro který jsou objekty $\mathbf{x}, \mathbf{y}$ nepřímo dosažitelné. Poznamenejme, že propojenost tvoří symetrickou relaci na množině objektů [15].

Algoritmicky můžeme hledání všech propojených objektů ve shluku chápat jako sofistikovanou obdobu prohledávání grafu do hloubky na základě dosažitelnosti jednotlivých objektů. Ilustrativní klasifikaci objektů pomocí výše uvedených definic vidíme na obrázku 3.14.



Obrázek 3.14: Ilustrace klasifikace objektů metodou DBSCAN s využitím Euklidovské vzdálenosti a parametry $\epsilon = 0{,}5$, $h = 2$

Na obrázku 3.14 vidíme všechny tři možnosti klasifikace objektů. Princip metody spočívá v iterativní klasifikaci všech objektů, na jejichž základě vytváříme shluky, kde pro nenavštívený objekt $\mathbf{x}$ můžeme tento princip shrnout ve třech základních krocích:

(1) Nalezení objektů v ϵ-okolí $S_\epsilon(\mathbf{x})$

(2) Pokud je $\mathbf{x}$ jádrový objekt, vytvořit shluk

(3) Pokud není $\mathbf{x}$ jádrový objekt, označit jako šum

Tvorba shluku (2) se řeší postupným hledáním přímo dosažitelných objektů, na jejichž základě nalezneme všechny nepřímo dosažitelné objekty, čímž získáme všechny propojené objekty v daném shluku a tím shluk definujeme. Hledání propojených objektů můžeme pozorovat na obrázku 3.14, kdy můžeme začít libovolným jádrovým objektem a díky postupnému prohledávání přímo dosažitelných objektů nalezneme všechny propojené objekty shluku.

Příklad

Ukážeme si princip výpočtu metody DBSCAN s Euklidovskou vzdáleností na datech z tabulky 3.1. Zvolme například $\epsilon = 1$ a $h = 3$. Začneme procházet, chceme-li navštěvovat, jednotlivé objekty, kdy prvním objektem je $\mathbf{a}$. Množinu navštívených objektů označme N. Spočteme vzdálenost objektu $\mathbf{a}$ ke všem objektům a červeně označíme vzdálenosti menší, nebo rovny ϵ.

	a	**b**	**c**	**d**	**e**	**f**	**g**
a	0	0,514782	0,364006	0,608276	3,064719	2,427447	2,553919

Množina objektů v ϵ-okolí objektu $\mathbf{a}$ je $S_\epsilon(\mathbf{a}) = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$. Množina navštívených objektů

je $N = \{\mathbf{a}\}$. Protože je počet objektů $|S_\epsilon(\mathbf{a})| = 4 \geq 3 = h$, nazveme objekt **a** jádrovým a budeme vytvářet nový shluk, označme jej $C_{abcd} = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$.

Musíme nalézt všechny objekty nacházející se v tomto shluku, tj. všechny propojené objekty tohoto shluku. Pro nalezení všech propojených objektů shluku C_{abcd} můžeme využít ještě nenavštívených objektů ve shluku C_{abcd} a nalézt objekty v jejich ϵ-okolí.

	a	**b**	**c**	**d**	**e**	**f**	**g**
b	0,514782	0	0,559017	0,380789	3,148412	2,510478	2,551960
c	0,364006	0,559017	0	0,832166	2,720294	2,080865	2,193171
d	0,608276	0,380789	0,832166	0	3,508917	2,869233	2,926175

Množina navštívených objektů je nyní $N = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$. Každý z objektů (**b**, **c**, **d**) je jádrový, ovšem přímo dosažitelné objekty každého z nich jsou již navšívené, shluk C_{abcd} tedy obsahuje všechny propojené objekty. Postupujeme na další, nenavštívený objekt, tj. **e** a určíme jeho vzdálenost ke všem objektům.

	a	**b**	**c**	**d**	**e**	**f**	**g**
e	3,064719	3,148412	2,720294	3,508917	0	0,640312	0,728011

Množina objektů v ϵ-okolí objektu **e** je $S_\epsilon(\mathbf{e}) = \{\mathbf{e}, \mathbf{f}, \mathbf{g}\}$. Množina navštívených objektů je nyní $N = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}\}$. Protože je počet objektů $|S_\epsilon(\mathbf{e})| = 3 \geq 3 = h$, nazveme objekt **e** jádrovým a budeme vytvářet nový shluk, označme jej $C_{efg} = \{\mathbf{e}, \mathbf{f}, \mathbf{g}\}$.

Musíme nalézt všechny objekty nacházející se v tomto shluku. Pro nalezení všech propojených objektů shluku C_{efg} můžeme využít ještě nenavštívených objektů ve shluku C_{efg} a nalézt objekty v jejich ϵ-okolí.

	a	**b**	**c**	**d**	**e**	**f**	**g**
f	2,427447	2,510478	2,080865	2,869233	0,640312	0	0,424264
g	2,553919	2,551960	2,193171	2,926175	0,728011	0,424264	0

Množina navštívených objektů je $N = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}, \mathbf{g}\}$. Každý z objektů (**f**, **g**) je jádrový, ovšem přímo dosažitelné objekty každého z nich jsou již navšívené, shluk C_{efg} tedy obsahuje všechny propojené objekty. Všechny objekty jsou již navštíveny, výsledný rozklad je tvořen shluky C_{abcd} a C_{efg}.

Za povšimnutí stojí, že žádný objekt nebyl okrajový. Důvodem je velká vzdálenost objektů patřících mezi tyto shluky a volba velkého ϵ. Zde je vhodné poznamenat, že metoda DBSCAN je velmi citlivá na volbu ϵ a h, kde i malou změnou ϵ můžeme získat naprosto odlišné rozklady.

Implementace v software R

Pro demonstraci řešení v software R implementujeme metodu DBSCAN s Euklidovskou vzdáleností, kterou využijeme k nalezení rozkladu datasetu Iris. Nalezení všech objektů ve shluku řešíme pomocí dynamického pole jádrových objektů, na základě kterých se hledají další propojené objekty.

Upozorněme na situaci, kdy je objekt $\mathbf{x}$ považován za šum, ale nový shluk C_y jej bere jako okrajový objekt. V takovém případě je samozřejmě nutno zpětně označit $\mathbf{x}$ jako objekt patřící do shluku C_y.

Další krajní situace nastává při klasifikaci okrajového objektu v jehož okolí se nachází více než jeden jádrový objekt. V tomto případě je okrajový objekt zařazen do shluku s jádrovým objektem, který byl navštíven první.

Vytvoříme si pomocnou funkci, jejímž výstupem je množina indexů tzv. sousedů v ϵ-okolí objektu $\mathbf{x}$, tj. $S_\epsilon(\mathbf{x}) \setminus \{\mathbf{x}\}$, na základě Euklidovské vzdálenosti d_e. Vstupem je datová matice (data), index objektu i a poloměr ϵ. Výstupem je vektor indexů sousedů.

```
# data: vstupni matice dat,  epsilon: velikost poloměru
# i: index objektu pro který hledáme sousedy v epsilon-okolí
shluk_dbscan_sousedi <- function(data, i, epsilon)
{
    sousedi = vector(mode = "integer"); N = nrow(data)
    for(j in (1:N)[-i])
      if(sqrt(sum((data[j, ]-data[i, ])^2)) <= epsilon)
        sousedi = append(sousedi, j)
    return (sousedi)
}
# výstup: vektor indexů sousedů v epsilon-okolí ítého objektu
```

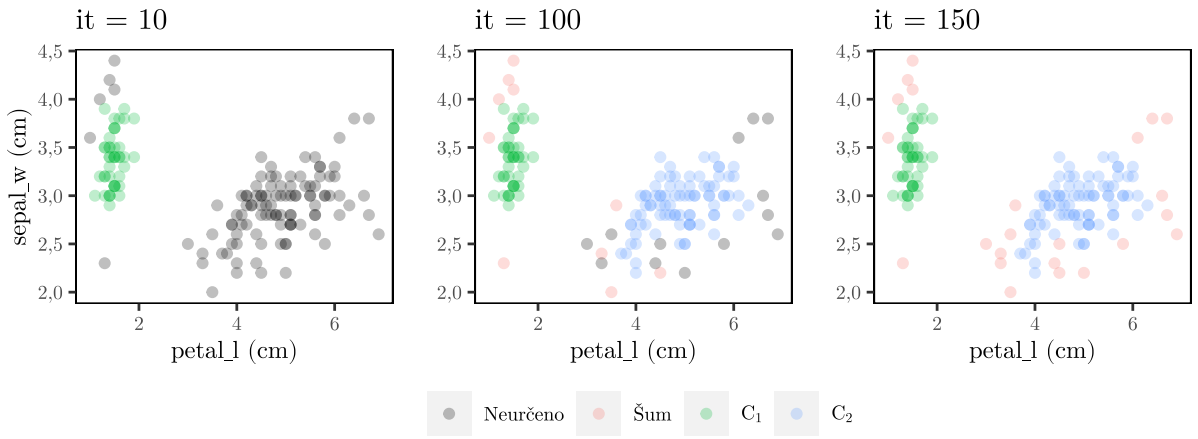
Pomocnou funkci využijeme pro implementaci metody DBSCAN s Euklidovskou vzdáleností. Vstupem algoritmu je datová matice (data), poloměr ϵ a hranice h. Výstupem je vektor, jehož složky odpovídají přiřazení původních objektů do nalezených shluků.

```
# data: vstupni matice dat,  epsilon: velikost poloměru
# h: minimální počet objektů pro shluk
shluk_dbscan <- function(data, epsilon, h)
{
  N = nrow(data); M = ncol(data)
  nenavstiven = -1; je_sum = 0; shluk = 0
  shluky = rep(nenavstiven, N)
  for(i in 1:N)
  {
    if(shluky[i] != nenavstiven) next
```

```
    # Nalezení sousedů v epsilon-okolí
    sousedi = shluk_dbscan_sousedi(data, i, epsilon)

    # Objekt není jádrový, označit jako šum
    if(length(sousedi)+1 < h)
      shluky[i] = je_sum
    else
    {
      shluk = shluk+1; shluky[i] = shluk; iter = 0
      # Nalezení všech sousedů
      while(iter < length(sousedi))
      {
        iter = iter + 1; id = sousedi[iter]
        if(shluky[id] == je_sum) shluky[id]=shluk
        if(shluky[id] != nenavstiven) next
        shluky[id] = shluk

        # Nalezeni sousedů v epsilon-okolí
        sousedi_novi = shluk_dbscan_sousedi(data, id, epsilon)
        # Pokud je jádrový objekt, přidat k prohledání
        if(length(sousedi_novi)+1 >= h)
          sousedi = append(sousedi, sousedi_novi)
      }
    }
  }
  return (shluky)
}
# výstup: vektor přiřazení objektů do shluků.  např. (1, 3, 2, 3 ...)
```

Pro dataset Iris je výsledek implementované metody DBSCAN shodný s výsledkem shlukování funkcí *dbscan* z balíčku *dbscan*, což si můžeme ověřit v příloze. Nadále budeme opět pracovat s funkcí *dbscan*. Hlavním důvodem je v tomto případě vyšší rychlost algoritmu.

Algoritmus využijeme k zobrazení postupu tvorby rozkladu atributů délky okvětního a šířky kališního lístku datasetu Iris v závislosti na počtu iterací (it). Zmíněný postup můžeme vidět na obrázku 3.15.



Obrázek 3.15: Ilustrace klasifikace objektů metodou DBSCAN ($\epsilon = 0{,}25$, $h = 5$) s využitím Euklidovské vzdálenosti v závislosti na počtu iterací pro atributy délky okvětního a šířky kališního lístku datasetu Iris

Na obrázku 3.15 vidíme, že metoda DBSCAN při volbě $\epsilon = 0{,}25$, $h = 5$ rozlišila dva shluky, C_1 a C_2, zbytek kosatců byl označen za šum. Výsledné rozklady si můžeme intuitivně odůvodnit, ale my víme, že počet shluků má být $k = 3$ bez šumu.

Můžeme očekávat, že shluk C_1 bude odpovídat kosatci Setosa a shluk C_2 kosatcům Virginica a Versicolor, které metoda DBSCAN s danými parametry nedokázala rozlišit. Zde si můžeme také všimnout první z vlastností této metody. Nemůžeme lehce určit počet shluků, museli bychom několikrát použít metodu s odlišnými kombinacemi ϵ a h.

Již zmíněný nedostatek metody DBSCAN spočívá v citlivosti na vstupní parametry ϵ a h. Další nedostatek této metody souvisí s hledáním shluků o různých hustotách. Typickým příkladem jsou blízké shluky s velmi odlišnými hustotami.

Výhody této metody spočívají hlavně v nalezení šumu a definovaní shluků libovolných tvarů. Náhrada požadavků na počet shluků požadavkem na jejich hustotu definovanou poloměrem ϵ a minimálním počtem objektů h ve shluku může být v některých případech preferováno.

Typicky se DBSCAN používá u souborů s malým počtem záznamů, u kterých lze očekávat, že budou obsahovat odlehlá pozorování, která nechceme zařadit do výsledných shluků, což není případ datasetu Iris. Aplikaci metody DBSCAN na tento dataset je proto nutno brát pouze jako ilustrativní.

3.5 Kvalita shlukování

Jaký je optimální počet shluků, jakou metodu zvolit a s jakými parametry? Na tyto otázky se bez externí informace nedá jednoznačně odpovědět. My jsme již na základě referenčního rozkladu definovali kvalitu shlukování, dle které jsme porovnávali jednotlivé metody.

Bez referenčního rozkladu se ovšem musíme uchýlit k jiným metodám měření kvality shlukování. Intuitivně považujeme za kvalitní rozklad takový, který je tvořen od sebe dobře separovatelnými a kompaktními shluky.

Na měření kompaktnosti a separovatelnosti shluků se můžeme dívat z mnoha úhlů pohledu, a proto, stejně jako u metod shlukování, i zde existuje více metod měření kvality shlukování, některými z nichž se zabývá např. [16].

My se podíváme např. na součet čtverců, siluetovou funkci, Calinského-Harabaszův, Dunnův a Davidův-Bouldinův index. Metody využijeme k nalezení optimálního počtu shluků v intervalu $\langle 2, 10\rangle$ při shlukování metodou průměrné vazby. Pro lepší pochopení jsou jednotlivé metody implementovány a v příloze porovnány s výstupem funkce *intCriteria* z balíčku *clusterCrit*.

3.5.1 Součet čtverců rozkladu

K měření kompaktnosti a vzdálenosti jednotlivých shluků C_i v rozkladu C o k shlucích můžeme využít tzv. součtu čtverců odchylek rozkladu. Celkový součet čtverců SS_T je definován jako součet čtverců vzdáleností jednotlivých objektů od centroidu všech objektů ($\bar{C}$).

$$SS_T(C) = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d^2(\mathbf{x}, \bar{C})$$

Celkový součet čtverců SS_T můžeme dále rozdělit na tzv. vnitroshlukový součet čtverců SS_W popisující kompaktnost shluků rozkladu a mezishlukový součet čtverců SS_B popisující separovatelnost shluků rozkladu, viz [14].

$$SS_T(C) = SS_W(C) + SS_B(C)$$

$$SS_W(C) = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d^2(\mathbf{x}, \bar{C}_i)$$

$$SS_B(C) = \sum_{i=1}^{k} |C_i|\, d^2(\bar{C}_i, \bar{C})$$

Pojďme se nyní zamyslet nad chováním vnitroshlukového a mezishlukového součtu čtverců. Začneme s vnitroshlukovým součtem čtverců SS_W. Uvažujme n objektů a počet shluků $k = n$, tedy každý objekt tvoří shluk. Jaká bude hodnota SS_W? Nulová, jelikož každý objekt bude centroidem shluku. Co nastane pokud snížíme počet shluků o jeden, tj. $k = n - 1$? Hodnota SS_W se zvýší, jelikož jeden ze shluků bude nyní tvořen dvěma objekty, můžeme očekávat, že centroid shluku nebude jedním z těchto objektů a bude tedy vzdálenější objektům ze shluku.

Obecněji, hodnota SS_W roste s klesajícím počtem shluků a je maximální, pokud jsou všechny objekty sloučeny do jednoho shluku, tedy $k = 1$. Proto hledáme velký pokles hodnoty SS_W na úkor přidání jednoho shluku, takový bod označme jako zlom (směrem dolů).

Implementujeme funkci, která spočte vnitroshlukový součet čtverců Euklidovské vzdálenosti d_e pro daný rozklad. Vstupem je datová matice a vektor přiřazení objektů do shluků.

```
# data: vstupni matice dat,  shluky: vektor přiřazení objektů do shluků
kvalita_vnitrni <- function(data, shluky)
{
  k = length(unique(shluky))
  SS_Wi = rep(0, k)
  for(i in 1:k)
  {
    shluk = data[which(shluky == i), , drop=F]
    SS_Wi[i] = sum(rowSums(sweep(shluk, 2, colMeans(shluk))^2))
  }
  return(sum(SS_Wi))
}
# výstup: vnitroshlukový součet čtverců
```

Zaměřme se nyní na mezishlukový součet čtverců SS_B. Uvažujme n objektů a $k = 1$, tedy všechny objekty jsou v jednom shluku. Jaká bude hodnota SS_B? Nulová, jelikož centroid shluku bude zároveň centroidem všech objektů $\bar{C}$. Co nastane pokud zvýšíme počet shluků o jeden, tj. $k = 2$? Hodnota SS_B se zvýší, jelikož centroidy C_1, C_2 již nebudou shodné s $\bar{C}$. Obecněji, hodnota SS_B roste s rostoucím počtem shluků a je maximální, pokud každý z objektů tvoří samostatný shluk, tj. při $k = n$. Opět hledáme zlom (směrem nahoru), tj. velký nárust hodnoty SS_B na úkor přidání jednoho shluku.

Implementujeme funkci, která spočte mezishlukový součet čtverců Euklidovské vzdálenosti d_e pro daný rozklad. Vstupem je datová matice a vektor přiřazení objektů do shluků.

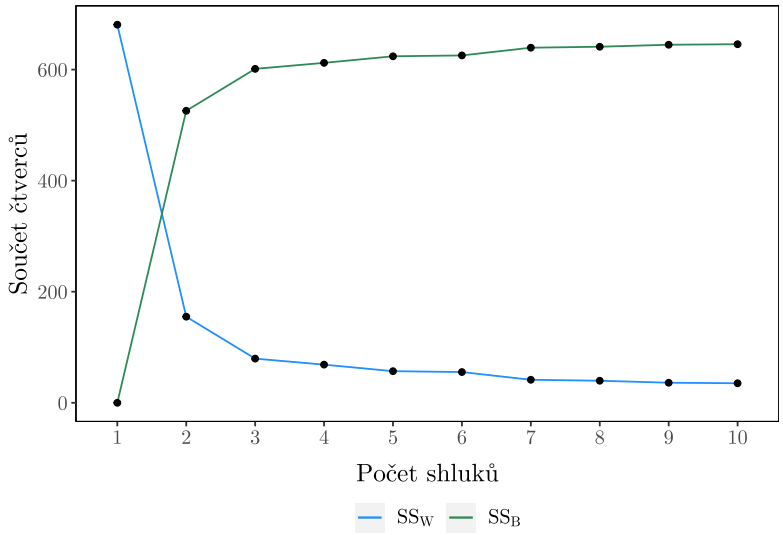
```
# data: vstupni matice dat,  shluky: vektor přiřazení objektů do shluků
kvalita_mezi <- function(data, shluky)
{
  k = length(unique(shluky))
  if(k == 1) return(0)
  SS_Bi = rep(0, k)
  for(i in 1:k)
  {
    shluk = data[which(shluky == i), , drop=F]; n = nrow(shluk)
    SS_Bi[i] = n * sum((colMeans(shluk)-colMeans(data))^2)
  }
  return(sum(SS_Bi))
```

```
}
# výstup: mezishlukový součet čtverců
```

Našim cílem je nalézt optimální počet shluků, tj. vysoce kompaktní (nízká hodnota SS_W) a dobře separovatelné (vysoká hodnota SS_B) shluky na úkor přidání jednoho shluku. Hledáme tedy počet shluků, při jehož volbě dojde k maximálnímu poklesu SS_W a nárustu SS_B oproti předchozímu počtu shluků.

Implementované funkce pro výpočet SS_W, SS_B při použití Euklidovské vzdálenosti využijeme k nalezení optimálního počtu shluků k metody průměrné vazby aplikované na dataset Iris. Hodnoty SS_W a SS_B pro jednotlivé hodnoty k metody průměrné vazby vidíme na obrázku 3.16.

Poznamenejme ještě, že hledání zlomu v $k = 2$ je problematické bez hodnot SS_W a SS_B v předchozím počtu shluků ($k = 1$), proto jsou v obrázku 3.16 přidány hodnoty SS_W a SS_B pro $k = 1$.



Obrázek 3.16: Hodnoty vnitroshlukového součtu čtverců SS_W a mezishlukového součtu čtverců SS_B pro rozklad datasetu Iris nalezený metodou průměrné vazby s využitím Euklidovské vzdálenosti v závislosti na počtu shluků k

Z obrázku 3.16 lze pozorovat již zmíněné chování vnitroshlukového a mezishlukového součtu čtverců. Zaměřme se nyní na hledání optimálního počtu shluků pomocí zlomů.

Největší zlom v našem případě nastává v $k = 2$, další, mnohem menší, v $k = 3$. Dle vnitroshlukového a mezishlukového součtu čtverců je tedy optimální počet shluků $k = 2$, případně $k = 3$.

Variabilita rozkladu

Na základě součtů čtverců můžeme definovat variability rozkladu C obdobně jako například v [11]. Uvažujme tzv. stupně volnosti, odpovídající počtu nezávislých veličin ovlivňujících výslednou hodnotu daného součtu čtverců.

Celkový součet čtverců SS_T vzdálenosti objektů od centroidu všech objektů $\bar{C}$ je ovlivněn n objekty, čemuž odpovídá $n-1$ stupňů volnosti. Definujme nyní celkovou variabilitu rozkladu MS_T jako podíl celkového součtu čtverců SS_T a příslušných stupňů volnosti.

$$MS_T(C) = \frac{SS_T(C)}{n-1}$$

Vnitroshlukový součet čtverců SS_W je založen na součtu vzdáleností objektů od centroidu shluku $\bar{C}_i$ do nějž je objekt přidělen, čemuž odpovídá $n-k$ stupňů volnosti (shluk C_i má $|C_i|-1$ stupňů volnosti). Definujme nyní vnitroshlukovou variabilitu rozkladu MS_W jako podíl vnitroshlukového součtu čtverců SS_W a příslušných stupňů volnosti.

$$MS_W(C) = \frac{SS_W(C)}{n-k}$$

Mezishlukový součet čtverců SS_B je založen na součtu vzdáleností k centroidů nalezených shluků $\bar{C}_i$ od centroidu všech objektů $\bar{C}$, čemuž odpovídá $k-1$ stupňů volnosti. Definujme nyní mezishlukovou variabilitu rozkladu MS_B jako podíl mezishlukového součtu čtverců SS_B a příslušných stupňů volnosti.

$$MS_B(C) = \frac{SS_B(C)}{k-1}$$

Je vhodné poznamenat, že myšlenka využití celkové variability rozkladu společně s vnitroshlukovou a mezishlukovou variabilitou není nová, využívá se například při analýze rozptylu a regresní analýze.

3.5.2 Siluetová funkce (Silhouette)

Jedna z nejpoužívanějších metod pro měření kvality shlukování je siluetová funkce [17]. Princip této metody spočívá v jednoduchých krocích, které společně tvoří kvalitní a v praxi osvědčenou metodu.

Mějme rozklad C o k shlucích, dále uvažujme objekt $\mathbf{x} \in C_i$, kde $|C_i| > 1$. Uvažujme funkci a přiřazující objektu $\mathbf{x}$ hodnotu popisující průměrnou vzdálenost objektu $\mathbf{x}$ od zbylých objektů uvnitř shluku C_i.

$$a(\mathbf{x}) = \frac{1}{|C_i|-1} \sum_{\mathbf{y} \in C_i, \mathbf{y} \neq \mathbf{x}} d(\mathbf{x}, \mathbf{y})$$

Dále uvažujme funkci c, přiřazující objektu $\mathbf{x}$ hodnotu popisující průměrnou vzdálenost objektu $\mathbf{x} \in C_i$ od objektů ve shluku C_j.

$$c(\mathbf{x}, C_j) = \frac{1}{|C_j|} \sum_{\mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$

Na základě funkce c definujme funkci b přiřazující objektu $\mathbf{x}$ hodnotu popisující průměrnou

vzdálenost objektu $\mathbf{x} \in C_i$ od objektů v nejbližším shluku.

$$b(\mathbf{x}) = \min_{C_j, C_j \neq C_i} c(\mathbf{x}, C_j)$$

A konečně definujme tzv. siluetovou funkci s. Funkce s je pro objekty tvořící samostatný shluk ($|C_i| = 1$) nulová. V opačném případě, tj. pro $|C_i| > 1$, přiřadí funkce s objektu $\mathbf{x}$ hodnotu, popisující kvalitu zařazení objektu $\mathbf{x}$ do shluku C_i na základě normalizovaného rozdílu funkčních hodnot $b(\mathbf{x})$ a $a(\mathbf{x})$.

$$s(\mathbf{x}) = \begin{cases} \dfrac{b(\mathbf{x}) - a(\mathbf{x})}{\max(b(\mathbf{x}), a(\mathbf{x}))} & \text{pro } |C_i| > 1 \\ 0 & \text{pro } |C_i| = 1 \end{cases}$$

Náš požadavek je opět stejný, hledáme vysoce kompaktní (vysoká hodnota $b(\mathbf{x})$) a dobře separovatelné (nízká hodnota $a(\mathbf{x})$) shluky. Z požadavků na $a(\mathbf{x})$ a $b(\mathbf{x})$ plyne, že kvalitní rozklad je charakterizován nízkým poměrem ${}^{a(\mathbf{x})}/_{b(\mathbf{x})}$, případně vysokým poměrem ${}^{b(\mathbf{x})}/_{a(\mathbf{x})}$.

Uvažujme dvě možnosti, tj. $a(\mathbf{x}) < b(\mathbf{x})$, nebo $a(\mathbf{x}) > b(\mathbf{x})$. Pro $a(\mathbf{x}) < b(\mathbf{x})$ můžeme rovnici přepsat do tvaru $s(\mathbf{x}) = 1 - {}^{a(\mathbf{x})}/_{b(\mathbf{x})}$.

Pro $a(\mathbf{x}) > b(\mathbf{x})$ můžeme rovnici přepsat do tvaru $s(\mathbf{x}) = -1 + {}^{b(\mathbf{x})}/_{a(\mathbf{x})}$. Po těchto úpravých vidíme, že obor hodnot siluetové funkce je $\langle -1, 1 \rangle$, kde 1 odpovídá správnému a hodnota -1 špatnému přiřazení objektu do shluku. Nejvyšších hodnot bude $s(\mathbf{x})$ nabývat pro objekty blízké danému centroidu shluku a naopak nejnižších hodnot bude $s(\mathbf{x})$ nabývat pro objekty tvořící hranici shluku.

Výsledné hodnoty siluetové funkce (zaokrouhleny na dvě desetinná místa) můžeme interpretovat detailněji dle doporučení v [8], které můžeme vidět v tabulce 3.8.

Tabulka 3.8: Interpretace hodnot siluetové funkce dle [8].

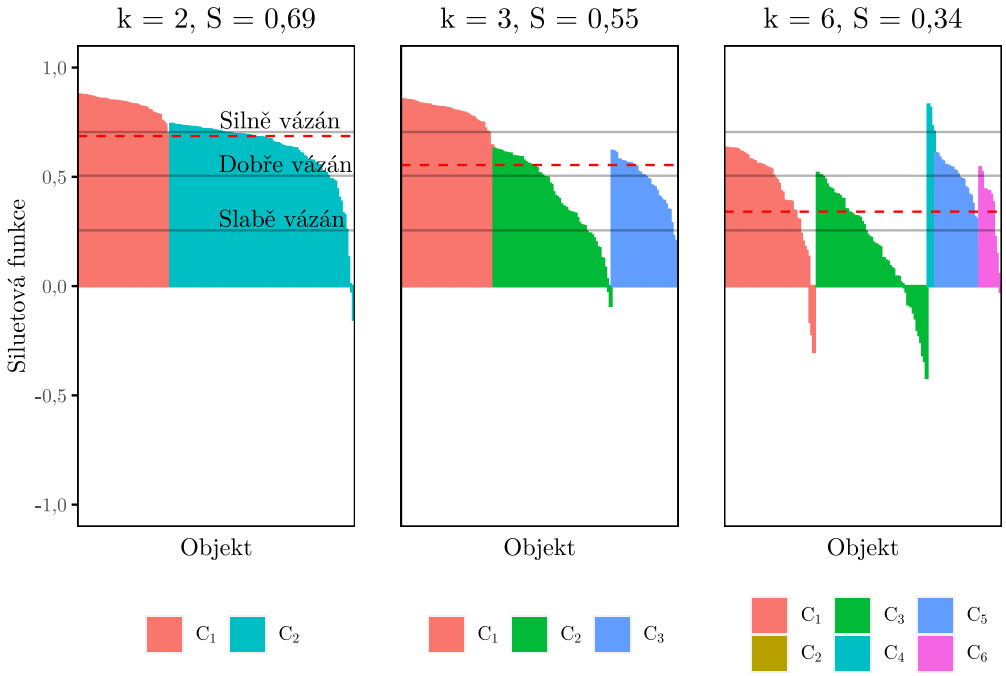
s($\mathbf{x}$)	Interpretace zařazení objektu $\mathbf{x} \in C_i$
$\langle 0,71, 1,00 \rangle$	Objekt je silně vázán ke shluku.
$\langle 0,51, 0,70 \rangle$	Objekt je dobře zařazen do shluku.
$\langle 0,26, 0,50 \rangle$	Objekt je slabě vázán ke shluku.
$\langle -1,00, 0,25 \rangle$	Objekt je pravděpodobně špatně zařazen do shluku.

Na závěr definujme tzv. průměrnou šířku rozkladu S, která rozkladu C přiřadí hodnotu, popisující průměrnou kvalitu přiřazení objektů do shluku.

$$S(C) = \frac{1}{n} \sum_{i=1}^{n} s(\mathbf{x_i})$$

Využijeme siluetové funkce k porovnání kvality rozkladu nalezeného metodou průměrné vazby s odlišným počtem shluků k aplikované na dataset Iris. Hodnoty siluetové funkce pro

rozklady nalezené metodou průměrné vazby společně s průměrnou šířkou rozkladu můžeme vidět na obrázku 3.17.



Obrázek 3.17: Hodnoty siluetové funkce s společně s průměrnou šířkou rozkladu S (červeně čerchovaná čára) pro rozklad datasetu Iris nalezený metodou průměrné vazby s využitím Euklidovské vzdálenosti v závislosti na počtu shluků k. Černě vyznačena doporučení dle [8].

Z obrázku 3.17 vidíme, že rozklad nalezený metodou průměrné vazby s počtem shluků $k = 2$ má nejvyšší průměrnou šířku rozkladu, v tomto případě je tedy nejlepší volbou. Obecně můžeme pozorovat, že s rostoucím počtem shluků klesá průměrná šířka rozkladu.

Pro detailnější interpretaci kvality přiřazení jednotlivých objektů můžeme využít doporučení z tabulky 3.2. Zaměřme se na kvalitu rozkladu pro $k = 3$. Můžeme si všimnout, že převážná většina objektů ze shluku C_1 je k němu silně vázána, kdežto u shluků C_2, C_3 je část objektů zařazena dobře, většina však pouze slabě, nebo špatně.

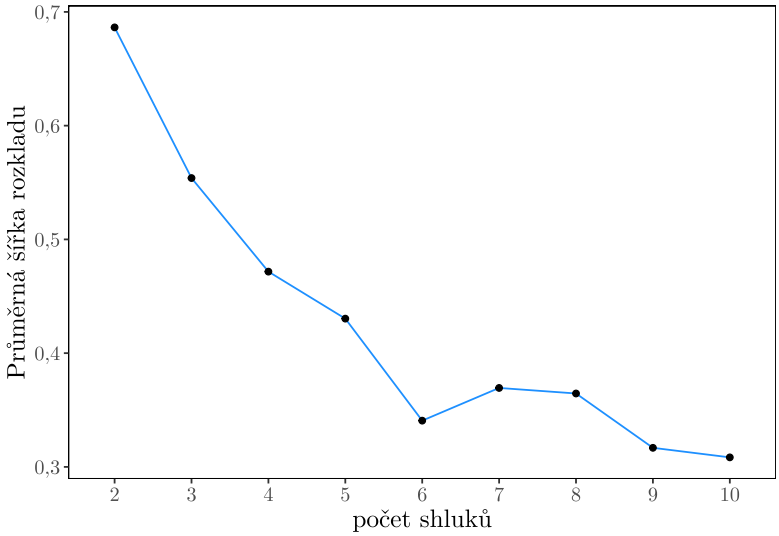
Můžeme předpokládat, že rozdílné kvality přiřazení objektů jsou v tomto případě způsobeny větší podobností shluků C_2, C_3, které odpovídají kosatcům Versicolor a Virginica. Tento jev jsme již viděli při analýze dat, díky které můžeme pochopit problematické rozlišení zmíněných shluků. Analogicky můžeme analyzovat i zbylé rozklady.

Siluetová funkce nám umožňuje porovnat kvalitu nejenom výsledného rozkladu, ale i jednotlivých objektů, díky čemuž získáme detailnější informace o jednotlivých shlucích.

My budeme nadále pracovat pouze s průměrnou šířkou rozkladu. Implementujeme funkci, která spočte průměrnou šířku rozkladu s využitím Euklidovské vzdálenosti d_e. Vstupem je datová matice a vektor přiřazení objektů do shluků.

```r
# data: vstupni matice dat,  shluky: vektor přiřazení objektů do shluků
kvalita_sil_index <- function(data, shluky)
{
  vzdalenosti = as.matrix(dist(data, upper = T, diag = T))
  nazvy = levels(shluky); s = rep(0, nrow(data))
  if(length(nazvy) == 1) return(0)
  shluky = split(1:nrow(data), shluky)
  for(i in 1:length(nazvy))
  {
    shluk = shluky[[i]]

    for(j in 1:length(shluk))
    {
      # Pokud objekt tvoří samostatný shluk
      if(length(shluk) == 1)
        {s[shluk[j]] = 0; next}

      vzdalenost = vzdalenosti[shluk[j], ]
      a = sum(vzdalenost[shluk])/(length(shluk)-1)
      b = min(sapply(shluky[-i], function(x) mean(vzdalenost[x])))
      s[shluk[j]] = (b - a) / max(b, a)
    }
  }
  return(mean(s))
}
# výstup: průměrná šířka rozkladu
```

Implementovanou funkci využijeme k nalezení optimálního počtu shluků k při využití metody průměrné vazby aplikované na dataset Iris. Průměrné šířky rozkladu S pro rozklady metody průměrné vazby vidíme na obrázku 3.18.



Obrázek 3.18: Průměrné šířky rozkladu S pro rozklad datasetu Iris nalezený metodou průměrné vazby s využitím Euklidovské vzdálenosti v závislosti na počtu shluků k.

Z obrázku 3.18 lze vidět, že optimální počet shluků dle průměrné šířky rozkladu je $k = 2$.

Je vhodné upozornit, že tato metoda bude kontrolována v příloze s výstupem funkce *silhouette* z balíčku *cluster* i přesto, že funkce *intCriteria* knihovny *clusterCrit* má volbu *silhouette*. Důvodem je odlišná definice výpočtu průměrné šířky rozkladu. Funkce *intCriteria* s volbou *silhouette* nejdříve počítá průměrné šířky shluků, průměrná šířka rozkladu je poté průměrem těchto hodnot. Problémem je, že výpočet průměru nebere v potaz velikosti jednotlivých shluků a proto je výsledná hodnota průměrné šířky rozkladu odlišná od námi očekávané.

3.5.3 Calinského-Harabaszův index

Jeden z indexů, který k hodnocení kompaktnosti a separovatelnosti využívá vnitroshlukové a mezishlukové variability rozkladu je Calinského-Harabaszův (ch) index [18].

Tento index je funkcí, která rozkladu C přiřadí hodnotu popisující podíl meziskupinové a vnitroskupinové variability.

$$ch(C) = \frac{MS_B(C)}{MS_W(C)} = \frac{\frac{SS_B}{k-1}}{\frac{SS_W}{n-k}} = \frac{SS_B}{k-1}\frac{n-k}{SS_W}$$

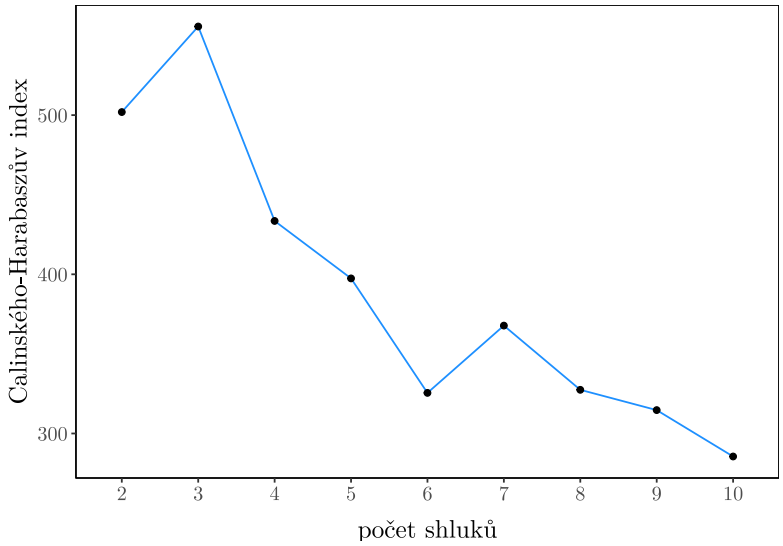
Jedná o analogií F-poměru používaného v analýze rozptylu, pojmenováného na počest Ronalda Fishera, s jehož datovým souborem pracujeme.

Jak již bylo zmíněno, hledáme vysoce kompaktní (nízká hodnota MS_W) a dobře separovatelné (vysoká hodnota MS_B) shluky. Vysoké hodnoty Calinského-Harabaszového indexu tedy indikují kvalitní shlukování.

Implementujeme funkci, která spočte Calinského-Harabaszův index. Vstupem je datová matice a vektor přiřazení objektů do shluků.

```
# data: vstupni matice dat,  shluky: vektor přiřazení objektů do shluků
kvalita_ch_index <- function(data, shluky)
{
  N = nrow(data); k = length(unique(shluky))
  if(k == 1 || k == N) return(NaN)
  vnitrni = kvalita_vnitrni(data, shluky)
  vnejsi = kvalita_mezi(data, shluky)
  return((vnejsi/vnitrni) * ((N-k)/(k-1)))
}
# výstup: Calinského-Harabaszův index
```

Využijeme implementované funkce k nalezení optimálního počtu shluků při využití metody průměrné vazby aplikované na dataset Iris. Hodnoty Calinského-Harabaszova indexu pro různé počty shluků k metody průměrné vazby vidíme na obrázku 3.19.



Obrázek 3.19: Hodnoty Calinského-Harabaszového *ch* indexu pro rozklad datasetu Iris nalezený metodou průměrné vazby s využitím Euklidovské vzdálenosti v závislosti na počtu shluků k.

Z obrázku 3.19 lze vidět, že optimální počet shluků dle Calinského-Harabaszového indexu je $k = 3$, případně $k = 2$.

3.5.4 Dunnův index

Další z metod pro měření kvality je Dunnův (du) index [19], který k hodnocení kompaktnosti využívá maximální vzdálenost uvnitř shluku a k hodnocení separovatelnosti nejmenší vzdálenost mezi shluky.

Uvažujme rozklad C a funkci pro výpočet vzdálenosti mezi shluky δ, která shlukům C_i, C_j

přiřadí vzdálenost odpovídající nejmenší vzdálenosti objektů $\mathbf{x} \in C_i, \mathbf{y} \in C_j$.

$$\delta(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$

Dále zaveďme funkci, označme *diam*, která shluku C_i přiřadí hodnotu popisující maximální vzdálenost objektů uvnitř tohoto shluku.

$$diam(C_i) = \max_{\mathbf{x}, \mathbf{y} \in C_i, \mathbf{x} \neq \mathbf{y}} d(\mathbf{x}, \mathbf{y})$$

Opět hledáme vysoce kompaktní (nízká hodnota *diam*) a dobře separovatelné (vysoká hodnota δ) shluky. Dále uvažujme poměr těchto hodnot $\delta(C_i, C_j)/diam(C_i)$, který pro kvalitní rozklad musí být, dle předpokladů pro *diam* a δ, vysoký.

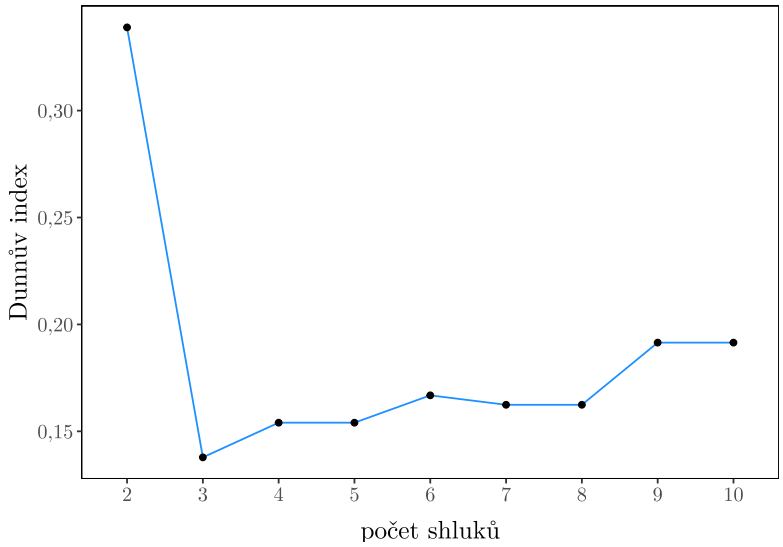
Dunnův index *du* je poté funkcí, která rozkladu C přiřadí hodnotu popisující nejnižší z těchto poměrů. Vysoké hodnoty Dunnova indexu tedy indikují kvalitní shlukování.

$$du(C) = \frac{\min\limits_{C_i, C_j \; C_i \neq C_j} \delta(C_i, C_j)}{\max\limits_{C_k} diam(C_k)}$$

Implementujeme funkci, která spočte Dunnův index s využitím Euklidovské vzdálenosti d_e. Vstupem je datová matice a vektor přiřazení objektů do shluků.

```
# data: vstupni matice dat,  shluky: vektor přiřazení objektů do shluků
kvalita_dunn_index <- function(data, shluky)
{
  vzdalenosti = as.matrix(dist(data, upper = T, diag = T))
  shluky = split(1:nrow(data), shluky)
  if(length(unique(shluky)) == 1) return(NaN)
  max_diam = -Inf
  for(i in 1:length(shluky))
  {
    if(max(vzdalenosti[shluky[[i]], shluky[[i]]]) > max_diam)
      max_diam = max(vzdalenosti[shluky[[i]], shluky[[i]]])
    vzdalenosti[shluky[[i]], shluky[[i]]] = Inf
  }
  return(min(vzdalenosti) / max_diam)
}
# výstup: Dunnův index
```

Využijeme implementované funkce k nalezení optimálního počtu shluků při využití metody průměrné vazby aplikované na dataset Iris. Hodnoty Dunnova indexu pro různé počty shluků k metody průměrné vazby vidíme na obrázku 3.20.



Obrázek 3.20: Hodnoty Dunnova *du* indexu pro rozklad datasetu Iris nalezený metodou průměrné vazby s využitím Euklidovské vzdálenosti v závislosti na počtu shluků k.

Z obrázku 3.20 lze vidět, že optimální počet shluků dle Dunnova indexu je $k = 2$.

3.5.5 Daviesův-Bouldinův index

Daviesův-Bouldinův (db) index [20] je posledním indexem, kterým se budeme zabývat. K hodnocení kompaktnosti využívá průměrné vzdálenosti objektů uvnitř shluků zatímco k hodnocení separovatelnosti používá vzdálenosti centroidů shluků.

Uvažujme rozklad C a funkci pro výpočet vzdálenosti mezi shluky δ, která shlukům C_i, C_j přiřadí vzdálenost odpovídající vzdálenosti jejich centroidů.

$$\delta(C_i, C_j) = d(\bar{C}_i, \bar{C}_j)$$

Dále zaveďme funkci $\bar{d}$, která shluku C_i přiřadí hodnotu popisující průměrnou vzdálenost objektů uvnitř shluku od jeho centroidu.

$$\bar{d}(C_i) = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \bar{C}_i)$$

Opět hledáme vysoce kompaktní (nízká hodnota $\bar{d}(C_i)$) a dobře separovatelné (vysoká hodnota $\delta(C_i, C_j)$). Dále uvažujme poměr $\bar{d}(C_i) + \bar{d}(C_j)$ k $\delta(C_i, C_j)$, který vzhledem k požadavkům na δ a $\bar{d}$ musí být pro kvalitní shluky nízký.

Zaveďme funkci m, která shluku C_i přiřadí hodnotu, popisující maximální hodnotu z výše uvedeného poměru.

$$m(C_i) = \max_{C_j, C_j \neq C_i} \left\{ \frac{\bar{d}(C_i) + \bar{d}(C_j)}{\delta(C_i, C_j)} \right\}$$

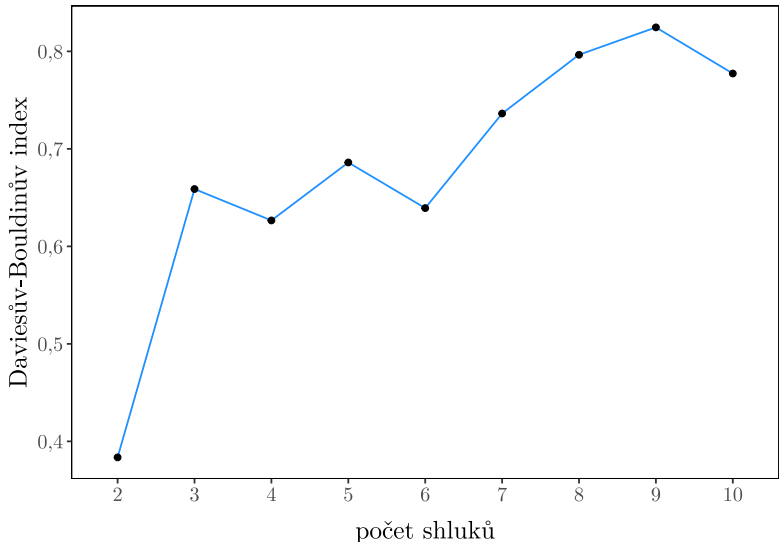
Daviesův-Bouldinův index db je funkcí, která rozkladu C přiřadí hodnotu, popisující průměrnou hodnotu m všech shluků. Nízké hodnoty Daviesova-Bouldinova indexu tedy indikují kvalitní shlukování.

$$db(C) = \frac{1}{k} \sum_{i=1}^{k} m(C_i)$$

Implementujeme funkci, která spočte Daviesův-Bouldinův index s využitím Euklidovské vzdálenosti d_e. Vstupem je datová matice a vektor přiřazení objektů do shluků.

```
# data: vstupni matice dat,  shluky: vektor přiřazení objektů do shluků
kvalita_db_index <- function(data, shluky)
{
  N = length(unique(shluky))
  shluky = split(data, shluky); if(N == 1) return(NaN)
  centroidy = matrix(unlist(lapply(shluky, colMeans)), nrow=N, byrow=T)
  vzdalenosti = as.matrix(dist(centroidy), diag=T, upper=T)
  diag(vzdalenosti) = Inf; m_i = rep(0, N)

  for(i in 1:length(shluky))
  {
    m = -Inf
    d_i = mean(sqrt(rowSums(sweep(shluky[[i]], 2, centroidy[i, ])^2)))
    for(j in (1:length(shluky))[-i])
    {
      d_j = mean(sqrt(rowSums(sweep(shluky[[j]], 2, centroidy[j, ])^2)))
      d = d_i + d_j
      if((d / vzdalenosti[i, j]) > m)
        m = d / vzdalenosti[i, j]
    }
    m_i[i] = m
  }
  return(mean(m_i))
}
# výstup: Daviesův-Bouldinův index
```

Využijeme implementované funkce k nalezení optimálního počtu shluků při využití metody průměrné vazby aplikované na dataset Iris. Hodnoty Daviesova-Bouldinova indexu pro různé počty shluků k metody průměrné vazby vidíme na obrázku 3.21.



Obrázek 3.21: Hodnoty Daviesova-Bouldinova *db* indexu pro rozklad datasetu Iris nalezený metodou průměrné vazby s využitím Euklidovské vzdálenosti v závislosti na počtu shluků k.

Z obrázku 3.21 lze vidět, že optimální počet shluků dle Daviesova-Bouldinova indexu je $k = 2$.

Seznámili jsme se s několika metodami měření kvality shlukování. Díky odlišným hodnocením kompaktnosti a separovatelnosti rozkladu jsme mohli pozorovat, že náš úsudek o optimálním počtu shluků rozkladu se může lišit v závislosti na použité metodě měření kvality shlukování.

Proto se pro nalezení optimálních parametrů dané metody použije několik metod pro měření kvality shlukování, na základě kterých se zvolí optimální parametry. V našem případě jsme hledali počet shluků k, který je v našem případě $k = 2$, případně $k = 3$.

Pro rychlou orientaci napříč metodami měření kvality shlukování můžeme využít tabulky 3.9.

Tabulka 3.9: Přehled metod měření kvality shlukování

Metoda měření kvality shlukování	Hledáme
Vnitroshlukový součet čtverců SS_W	Zlom ↓
Mezishlukový součet čtverců SS_B	Zlom ↑
Siluetová funkce s a průměrná šířka rozkladu S	Maximum
Calinského-Harabaczův *ch* index	Maximum
Dunnův *du* index	Maximum
Daviesův-Bouldinův *db* index	Minimum

Srovnání hierarchických metod shlukování

Uvedené metody měření kvality shlukování využijeme k porovnání rozkladů ($k = 3$) vybraných hierarchických metod u kterých jsme díky referečnímu rozkladu $\hat{C}$ spočetli kvalitu, viz tabulka 3.5 a obrázek 3.11. Výsledky vybraných metod měření kvality shlukování aplikované na

rozklad datasetu Iris ($k = 3$) nalezený konkrétní hierarchickou metodou s využitím Euklidovské vzdálenosti vidíme v Tabulce 3.10.

Tabulka 3.10: Porovnání nalezených rozkladů vybraných metod hierarchického shlukování aplikovaných na dataset kosatců s využitím Euklidovské vzdálenosti na základě metod měření kvality shlukování

Metoda shlukování	Metoda měření kvality shlukování					
	SS_W	SS_B	S	ch	du	db
Nejbližšího s.	142,57	538,26	0,51	277,49	0,17	0,45
Nejvzdálenějšího s.	89,61	591,21	0,51	484,90	0,10	0,63
Průměrné v.	79,54	601,29	0,55	555,67	0,14	0,66
Wardova	79,39	601,44	0,55	556,84	0,11	0,66

Z tabulky 3.10 vidíme s použitím tabulky 3.9, že nalezení optimální metody není jednoznačné. Překvapivě metoda nejbližšího souseda dosahuje dle Dunnového a Davies-Bouldinova indexu nejlepší kvality. Naopak u vnitroshlukového i mezishlukového součtu čtverců společně s Calinského-Harabaszovým indexem a průměrnou šířkou rozkladu dosahuje kvality nejnižší.

Dále můžeme pozorovat, že metody dosahující nejlepší kvality na základě referenčního shlukování (viz obrázek 3.11, tabulka 3.5), tj. metoda průměrné vazby a Wardova metoda dosahují nejlepší kvality u vnitroshlukového i mezishlukového součtu čtverců společně s Calinského-Harabaszovým indexem a průměrnou šířkou rozkladu.

Na závěr je vhodné poznamenat, že volbou Euklidovské vzdálenosti pro metody shlukování a měření kvality shlukování jsme způsobili větší citlivost na odlišné rozpětí jednotlivých atributů. Tento problém se v praxi řeší standardizací jednotlivých atributů (viz podkapitola Euklidovské vzdálenosti kapitoly 3.1). Společně s malým datasetem jako je Iris to vede k zajímavému výsledku, kdy malá změna v nalezených rozkladech vlivem standardizace (např. odlišně přiřazeny tři objekty) vede k naprosto odlišným výsledkům měření kvality shlukování.

Kapitola 4

Závěr

V této práci jsme se zabývali shlukovou analýzou. Práce byla zhotovena v praktickém duchu, kdy postup a členění práce odpovídá typickému přístupu datového analytika. Teoretické části byly doprovázeny ilustracemi a následnými příklady. Většina z uvedených metod shlukování byla pro lepší pochopení implementována a aplikována na datasetu Iris tvořeným kvantitativními atributy.

Nejdříve jsme se s datasetem Iris seznámili při jeho analýze. Následně byl uveden teoretický popis a úvod do shlukové analýzy. Zabývali jsme se metodami hierarchického i nehierarchického shlukování. U metod hierarchického shlukování jsme využili referenčního rozkladu k vyhodnocení kvality jednotlivých metod. Viděli jsme diverzitu těchto metod, jejichž analýza nám umožnila zjistit jejich nedostatky a výhody.

Dále jsme se seznámili se základními metodami nehierarchického shlukování. Našim dalším úkolem bylo využít metod měření kvality shlukování k vyhodnocení a nalezení optimálního parametru (počtu shluků) metody průměrné vazby. Metody měření kvality shlukování jsme dále využili k vyhodnocení a porovnání hierarchických metod shlukování.

Nelze opomenout fakt, že problematika shlukování je mnohem širší. V práci jsme se nedotkli například vlivu standardizace dat na kvalitu shlukování, popř. shlukování objektů s kvalitativními atributy.

Hlavním tématem, které se opakovalo po celou dobu této práce, byla subjektivita shlukování. Můžeme říci, že vytvoření rozkladu je jednoduché, ovšem vytvoření kvalitního shlukování s požadovanými vlastnostmi je mnohem složitější. Volbu metod shlukování můžeme zúžit pouze díky předpokladům na výsledné shluky, nebo pomocí experta, který nám poskytl externí informaci o souboru.

Shluková analýza patří nyní k praktickým a velmi užitečným pomůckám při analýze dat. V úvodu byly zmíněny některé z aplikací shlukové analýzy, typicky ve financích a marketingu. S očekávaným technologickým pokrokem bude růst množství dat a tedy i možností aplikace shlukové analýzy, díky které jsme schopni nalézt strukturu v datech a získat tak užitečné informace, které můžeme následně využít k jejich porozumění.

Literatura

[1] U. von Luxburg, „Statistical Learning with Similarity and Dissimilarity Functions", PhD thesis, Technische Universität Berlin, Germany; Technische Universität Berlin, Germany, 2004.

[2] L. A. B. Muñoz, „Understanding (dis)similarity measures", *CoRR*, roč. abs/1212.2791, 2012 [Online]. Dostupné z: http://arxiv.org/abs/1212.2791

[3] P. Strejc, „Shluková analýza ve financích". Univerzita Karlova, Matematicko-fyzikální fakulta, Katedra pravděpodobnosti a matematické statistiky, Ke Karlovu 3, 121 16 Praha 2, 2009.

[4] V. Kumar, J. K. Chhabra, a D. Kumar, „Impact of Distance Measures on the Performance of Clustering Algorithms", in *Intelligent Computing, Networking, and Informatics*, 2014, s. 183--190.

[5] E. Janoušová, J. Holčík, D. Haruštiaková, S. Littnerová, a J. Jarkovský, „Vícerozměrné metody pro analýzu a klasifikaci dat". elektronická verze "online"; Masarykova univerzita, Brno, 2015 [Online]. Dostupné z: http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnocen i-biologickych-dat--vicerozmerne-metody-pro-analyzu-dat

[6] C. C. Aggarwal, A. Hinneburg, a D. A. Keim, „On the Surprising Behavior of Distance Metrics in High Dimensional Space", in *Database Theory — ICDT 2001*, 2001, s. 420--434.

[7] M. J. Zaki a W. M. Jr, *Data Mining and Analysis: Fundamental Concepts and Algorithms*. USA: Cambridge University Press, 2014.

[8] L. Kaufman a P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.

[9] R. A. Johnson, *Applied Multivariate Statistical Analysis (6th Edition)*. Pearson, 2007 [Online]. Dostupné z: https://www.xarg.org/ref/a/0131877151/

[10] D. Müllner, „Modern hierarchical, agglomerative clustering algorithms". 2011 [Online]. Dostupné z: http://arxiv.org/abs/1109.2378

[11] M. Litschmannová, „Úvod do statistiky." Ostrava: VŠB-TU Ostrava, 2020 [Online]. Dostupné z: http://mi21.vsb.cz/modul/uvod-do-statistiky

[12] J. Han, M. Kamber, a J. Pei, „10 - Cluster Analysis: Basic Concepts and Methods", 2012, s. 443--495.

[13] C. Reddy a B. Vinzamuri, „A Survey of Partitional and Hierarchical Clustering Algorithms", 2018, s. 87--110.

[14] P.-N. Tan, M. Steinbach, a V. Kumar, *Introduction to Data Mining, (First Edition)*. USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[15] M. Ester, H. Kriegel, J. Sander, a X. Xu, „A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", in *KDD*, 1996.

[16] Y. Liu, Z. Li, H. Xiong, X. Gao, a J. Wu, „Understanding of Internal Clustering Validation Measures", in *Proceedings of the 2010 IEEE International Conference on Data Mining*, 2010, s. 911--916, doi: 10.1109/ICDM.2010.35 [Online]. Dostupné z: https://doi.org/10.1109/ICDM.2010.35

[17] P. Rousseeuw, „Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis", *J. Comput. Appl. Math.*, roč. 20, č. 1, s. 53--65, lis. 1987, doi: 10.1016/0377--0427(87)90125-7. [Online]. Dostupné z: https://doi.org/10.1016/0377-0427(87)90125-7

[18] T. Caliński a J. Harabasz, „A dendrite method for cluster analysis", *Communications in Statistics*, roč. 3, č. 1, s. 1--27, 1974, doi: 10.1080/03610927408827101. [Online]. Dostupné z: https://www.tandfonline.com/doi/abs/10.1080/03610927408827101

[19] J. C. Dunn†, „Well-Separated Clusters and Optimal Fuzzy Partitions", *Journal of Cybernetics*, roč. 4, č. 1, s. 95--104, 1974, doi: 10.1080/01969727408546059. [Online]. Dostupné z: https://doi.org/10.1080/01969727408546059

[20] D. L. Davies a D. W. Bouldin, „A Cluster Separation Measure", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, roč. PAMI-1, č. 2, s. 224--227, 1979, doi: 10.1109/TPAMI.1979.4766909.

Obrázky

[O1] Alchetron, Free Social Encyclopedia for The World, [Iris Setosa]. *Alchetron* [Online]. [cit. 19.04.2021]. Dostupné z: https://alchetron.com/Iris-setosa#iris-setosa-0ab3145a-68f2-41ca-a529-c02fa2f5b02-resize-750.jpeg

[O2] Plant World Devon Ltd., [Iris Versicolor]. *Plant World Seeds* [Online]. [cit. 19.04.2021]. Dostupné z: https://www.plant-world-seeds.com/store/view_seed_item/3664

[O3] Dr. Jean Everett, College of Charleston, „Iris virginica var. virginica" [Iris Virginica]. *U.S. Forest Service* [Online]. [cit. 19.04.2021]. Dostupné z: https://www.fs.fed.us/wildflowers/beauty/iris/Blue_Flag/iris_virginica.shtml

Software

[S1] R Core Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2020 [Online]. Dostupné z: https://www.R-project.org/

[S2] M. Hahsler, M. Piekenbrock, a D. Doran, „dbscan: Fast Density-Based Clustering with R", *Journal of Statistical Software*, roč. 91, č. 1, s. 1--30, 2019, doi: 10.18637/jss.v091.i01.

[S3] B. Desgraupes, *clusterCrit: Clustering Indices*. 2018 [Online]. Dostupné z: https://CRAN.R-project.org/package=clusterCrit

[S4] A. Kassambara a F. Mundt, *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. 2020 [Online]. Dostupné z: https://CRAN.R-project.org/package=factoextra

[S5] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016 [Online]. Dostupné z: https://ggplot2.tidyverse.org

[S6] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, a K. Hornik, *cluster: Cluster Analysis Basics and Extensions*. 2019.

[S7] T. L. Pedersen, *patchwork: The Composer of Plots*. 2020 [Online]. Dostupné z: https://CRAN.R-project.org/package=patchwork

[S8] H. Wickham a D. Seidel, *scales: Scale Functions for Visualization*. 2020 [Online]. Dostupné z: https://CRAN.R-project.org/package=scales

[S9] L. Komsta a F. Novomestky, *moments: Moments, cumulants, skewness, kurtosis and related tests*. 2015 [Online]. Dostupné z: https://CRAN.R-project.org/package=moments

[S10] Y. Qiu a authors/contributors of the included software. See file AUTHORS for details., *showtext: Using Fonts More Easily in R Graphs*. 2020 [Online]. Dostupné z: https://CRAN.R-project.org/package=showtext

[S11] Y. Xie, *bookdown: Authoring Books and Technical Documents with R Markdown*. 2020 [Online]. Dostupné z: https://github.com/rstudio/bookdown

[S12] Y. Xie, *bookdown: Authoring Books and Technical Documents with R Markdown*. Boca Raton, Florida: Chapman; Hall/CRC, 2016 [Online]. Dostupné z: https://github.com/rstudio/bookdown

[S13] Y. Xie, „knitr: A Comprehensive Tool for Reproducible Research in R", in *Implementing*

Reproducible Computational Research, V. Stodden, F. Leisch, a R. D. Peng, Ed. Chapman; Hall/CRC, 2014 [Online]. Dostupné z: http://www.crcpress.com/product/isbn/9781466561595

[S14] Y. Xie, *Dynamic Documents with R and knitr*, 2nd vyd. Boca Raton, Florida: Chapman; Hall/CRC, 2015 [Online]. Dostupné z: https://yihui.org/knitr/

[S15] Y. Xie, *knitr: A General-Purpose Package for Dynamic Report Generation in R*. 2020 [Online]. Dostupné z: https://yihui.org/knitr/

[S16] M. Papadakis *et al.*, *Rfast: A Collection of Efficient and Extremely Fast R Functions*. 2020 [Online]. Dostupné z: https://CRAN.R-project.org/package=Rfast

[S17] A. Kassambara a F. Mundt, *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. 2020 [Online]. Dostupné z: https://CRAN.R-project.org/package=factoextra

[S18] S. Meschiari, *latex2exp: Use LaTeX Expressions in Plots*. 2015 [Online]. Dostupné z: https://CRAN.R-project.org/package=latex2exp

[S19] B. Rudis, B. Bolker, a J. Schulz, *ggalt: Extra Coordinate Systems, 'Geoms', Statistical Transformations, Scales and Fonts for 'ggplot2'*. 2017 [Online]. Dostupné z: https://CRAN.R-project.org/package=ggalt

Příloha A

Soubory

Příloha je tvořena interaktivním poznámkovým blokem v R (shluk_analyza.Rmd) jehož obsahem jsou jednotlivé implementace metod shlukování společně s metodami měření kvality shlukování a jejich následná kontrola.

Součástí je dataset Iris v souboru iris.csv.