UNIVERZA V MARIBORU

FAKULTETA ZA NARAVOSLOVJE IN MATEMATIKO

Oddelek za matematiko in računalništvo

DIPLOMSKO DELO

Eva Ferk

Maribor, 2009

UNIVERZA V MARIBORU

FAKULTETA ZA NARAVOSLOVJE IN MATEMATIKO

Oddelek za matematiko in računalništvo

Diplomsko delo

DELNE UREJENOSTI IN HIERARHIČNO GRUČENJE

Mentorja:

doc. dr. Drago Bokal

doc. dr. Krista Rizman Žalik

Kandidatka:

Eva Ferk

Maribor, 2009

ZAHVALA

V življenju so poti, ki jih je treba prehoditi,
so ljudje, katere je vredno spoštovati
in so dlani, katerim želimo v zahvalo stisniti.

Iz srca se zahvaljujem mentorjema, doc. dr. Dragu Bokalu in doc. dr. Kristi Rizman Žalik, za strokovno vodenje, motivacijo in pomoč pri nastajanju diplomske naloge.

Posebna hvala vsem sodelavcem za razumevanje, še posebej Andreji in Robertu za vso podporo, ki sem je bila deležna. Hvala tudi Nejcu za vzpodbudne besede, pomoč pri oblikovanju in potrpežljivost ob neštetih vprašanjih.

Velika hvala pa tudi staršem, bratu ter vsem, ki so mi v tem napornem, a zelo lepem in pomembnem obdobju življenja stali ob strani, me vzpodbujali ter verjeli vame in s tem pripomogli k temu, da sem postala, kar sem si želela.

Vsem iskreno hvala.

UNIVERZA V MARIBORU

FAKULTETA ZA NARAVOSLOVJE IN MATEMATIKO

IZJAVA

Podpisana Eva Ferk, rojena 10. avgusta 1985, študentka Fakultete za naravoslovje in matematiko Univerze v Mariboru, študijskega programa matematika in računalništvo, izjavljam, da je diplomsko delo z naslovom

DELNE UREJENOSTI IN HIERARHIČNO GRUČENJE

pri mentorjih doc. dr. Drago Bokal in doc. dr. Krista Rizman Žalik, avtorsko delo. V diplomskem delu so uporabljeni viri in literatura korektno navedeni; teksti niso uporabljeni brez navedbe avtorjev.

Maribor, 17. marec 2009

Delne urejenosti in hierarhično gručenje

program skupnega diplomskega dela

Delitev podatkov v skupine ali gručenje predstavlja enega osnovnih pristopov k organiziranju velikih količin podatkov, pri katerih je znana le podobnost med posameznimi podatki. Na podlagi podobnosti se z algoritmi za gručenje podatki razdelijo v gruče, od katerih vsaka vsebuje le med sabo podobne podatke.

Matematično gledano je taka razdelitev particija množice podatkov oz. ekvivalenčna relacija, v kateri sta dva podatka ekvivalentna, če sta si podobna. Ker pa je podobnost vedno nekoliko subjektiven pojem, je mogoče dano množico podatkov v gruče razdeliti na več načinov, dobimo torej več različnih ekvivalenčnih relacij. Množico takih ekvivalenčnih relacij nad dano množico podatkov imenujemo podatkovna hierarhija. Podatkovna hierarhija je naravno delno urejena z relacijo 'biti finejši od'. V diplomskem delu bomo proučevali odnos med to relacijo in algoritmi za gručenje.

V matematičnem delu diplomskega dela predstavite pojem relacije in lastnosti relacij, pojem ekvivalenčne relacije in pojem delne urejenosti. Predstavite pojem grafa delne urejenosti ter grafa pokritij delne urejenosti. V nadaljevanju definirajte podatkovno hierarhijo in pokažite, da je podatkovna hierarhija delno urejena z relacijo 'biti finejši od'. Zastavi se vprašanje, katere delne urejenosti se lahko pojavijo kot podatkovne hierarhije. Zato za nekaj družin grafov, ki nastopajo kot grafi pokritij delnih urejenosti, pokažite, da jih je mogoče predstaviti kot podatkovne hierarhije.

V računalniškem delu diplomskega dela predstavite hierarhični pristop k gručenju, ki ga lahko izvajamo z združevanjem ali z delitvijo gruč. Obrazložite predstavitev hierarhij gruč z dendogramom in ugnezdenim gručnim diagramom. Podrobneje predstavite združevalne hierarhične metode gručenja. Implementirajte algoritem, ki bo izvedel hierarhično gručenje po maksimalni, minimalni in povprečni metodi. Prikažite rezultate na izbranih testnih primerih in metode primerjajte. Opišite prostorsko in časovno zahtevnost algoritma.

V zadnjem poglavju pa predstavite povezavo med matematičnim in računalniškim delom diplomske naloge. Pri tem obrazložite podatkovne hierarhije, ki izhajajo iz predstavljenih algoritmov za gručenje in pokažite, da so podatkovne hierarhije, ki ustrezajo dendrogramom, linearno urejene.

Priporočljiva literatura:

1. J. Han, M. Kambler: Data mininig, concepts and techniques, second edition, Morgan Kaufmann publishers, San Francisco, 2006.

2. A. Brandstädt, V. B. Le, J. P. Spinrad, Graph classes, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philiadelphia, 1999.

3. W. T. Trotter, Combinatorics and partially ordered sets: dimension theory. The Johns Hopkins University Press, Baltimore, MD, 1992.

4. G. K. Gupta, A. Strehl, J. Ghosh, Distance based clustering of association rules, v Intelligent Engineering Systems Through Artificial Neural Networks, (Proceedings 1999), 759–764, ASME Press, 1999.

doc. dr. Drago Bokal,
doc. dr. Krista Rizman Žalik

FERK, E.: Delne urejenosti in hierarhično gručenje.
Diplomsko delo, Univerza v Mariboru, Fakulteta za naravoslovje in matematiko, Oddelek za matematiko in računalništvo, 2009.

IZVLEČEK

Gručenje podatkov velja za eno najpomembnejših metod podatkovnega rudarjenja, ki se kot nova informacijska tehnologija dnevno razvija. Razvrščanja objektov v gruče so se tekom let raziskovalci lotevali na več načinov, kar s seboj prinese obilico različnih metod in postopkov. V diplomski nalogi se podrobneje seznanimo z merili za podobnost objektov znotraj posamezne gruče. Predstavljenih je več metod, od tega so tri hierarhične metode implementirane, predstavljene pa so tudi razlike med njimi.

Vsaka razvrstitev objektov v gruče je matematično gledano ekvivalenčna relacija. Dva podatka sta ekvivalentna, če sta v isti gruči.

V prvem delu je razvito matematično orodje, s katerim kasneje raziskujemo lastnosti podatkovne hierarhije, ki nastane med izvajanjem algoritmov gručenja.

Končna ugotovitev kaže na to, da je reducirani graf podatkovne hierarhije, ki ga dobimo tekom razvrščanja hierarhičnih algoritmov gručenja, enak poti, za nehiearhično metodo K-voditeljev pa je to graf brez povezav.

Ključne besede: podatkovna hierarhija, gručenje podatkov, delne urejenosti, dendrogram, ugnezdeni gručni diagram, ekvivalenčna relacija, minimalna metoda, maksimalna metoda, povprečna metoda.

FERK, E.: Partial orders and hierarchical clustering.
Graduation Thesis, University of Maribor, Faculty of Natural Scicences and Mathematics, Department of Mathematics and Computer Science, 2009.

ABSTRACT

Data clustering is the task of organizing a set of object into groups (clusters) according to some similarity measure of the objects. As such, data clustering represents one of the most important methods of data mining. Over the years, the researchers have classified objects into clusters in several ways, which brings up plenty of different methods and procedures. In this graduation thesis, we implemented three hierarchical clustering methods and studied differences between them.

Each classification of objects is from mathematical point of view an equivalence relation. Two data objects are equivalent if they are in the same cluster.

In the first part, a mathematical framework of data hierarchies is developed. It enables us to study characteristics of data hierarchy, which we obtain during the execution of clustering algorithms. We show that reduced graph of data hierarchy, which results from execution of a hierarchical clustering algorithm, is a path. During nonhierarchical algorithm of K-means we obtain a graph without edges.

Keywords: Data hierarchy, clustering, partial order, dendrogram, nested cluster diagram, equivalence relation, single link, complete link, group-average agglomerative clustering.

Math. Subj. Class. (2000): 06A06 Partial order, general,
06A07 Combinatorics of partially ordered sets,
68R10 Discrete mathematics in relation to computer science, graph theory.

Kazalo

Seznam uporabljenih kratic in simbolov

R	relacija
D	množica podatkov
V	množica ekvivalenčnih relacij
H	podatkovna hierarhija
V(G)	množica vozlišč grafa G
E(G)	množica povezav grafa G
P_n	pot na n vozliščih
C_n	cikel na n vozliščih
$R[x]$	ekvivalenčni razred
$\preceq$	relacija 'biti finejši'
$\sim$	ekvivalentnost
$d(X, Y)$	mera podobnosti
C	množica vseh gruč, razvrstitev
C_i	i-ta gruča
K	število vseh gruč
T_i	težišče gruče C_i
n_i	število objektov v gruči C_i
x_{ki}	i-ti objekt v gruči C_k
m_k	novi voditelj (težišče) k-te gruče

Uvod

Razvrščanje v skupine je pojem, s katerim se srečujemo v vsakdanjem življenju, čeprav se ga včasih niti ne zavedamo. Naši možgani so usmerjeni k razvrščanju objektov po velikosti, barvi, itd. Prav tako zavedno ali ne razvrstimo ljudi okoli sebe na razne načine. Že sama delitev na moške in ženske, na sorodnike in sodelavce, na starejše in mlajše, je razvrščanje, čeprav je za nas to povsem običajen proces. Vsako stvar lahko glede na njene lastnosti uvrstimo v skupine. Vendar pa se v svetu obilice podatkov pojavljajo problemi, ki jih glede na veliko količino in kopico lastnosti sami preprosto ne moremo ustrezno razvrstiti. Eden izmed takšnih problemov je razvrstitev celotne vsebine svetovnega spleta po tematikah. Tekom svojega življenja posameznik ne reši tega problema, takšnih problemov pa je vedno več. Zato vse več časa posvečamo razvijanju novih informacijskih tehnologij in s tem tudi razvoju metod in postopkov, ki čim bolj učinkovito in hitro razvrstijo podatke s podobnimi lastnostmi v iste skupine. Pri vsakodnevni obilici monotonega razvrščanja bi si tako že zelo želeli, da bi obstajal stroj, ki bi zamudno delo opravil namesto nas.

V strokovni literaturi pa je razvrščanje v skupine ali gručenje podatkov v gruče (angl. data clustering) proces, ki razvrsti objekte v gruče na podlagi podobnih lastnosti. Objekti znotraj posamezne gruče so si čim bolj podobni, prav tako pa se tem bolj razlikujejo od objektov drugih gruč.

Razvrščanje v skupine je uporabno na zelo širokem področju. Vedno pogoststeje ga srečujemo v biologiji in medicini, npr. pri diagnozi bolezni in analizi genskega materiala, v marketingu, kjer se išče potrošnike s podobnim vedenjskim vzorcem, uporablja pa se tudi v svetovnem spletu pri reševanju zgornjega problema.

Če na razvrščanje objektov pogledamo iz matematičnega vidika, opazimo, da razvrstitev v gruče predstavlja ekvivalenčno relacijo. Dva objekta sta ekvivalentna natanko takrat, kadar sta v isti gruči. Kot v vsakdanjem življenju tudi v matematiki vpeljemo pojem “finosti”. Kot so finejše zmleti orehi boljši, saj so razdrobljeni na manjše koščke, tako je tudi finejša

relacija razdeljena na več ekvivalenčnih razredov. Z množico ekvivalenčnih relacij nad podatki vpeljemo tudi pojem podatkovne hierarhije. S tem pa se pojavi vprašanje, kakšne podatkovne hierarhije pridobimo po izvedbi algoritmov gručenja.

Delo je organizirano v tri dele. V prvem razvijamo matematično orodje za primeravo rezultatov gručenja - podatkovno hierarhijo. Najprej predstavimo matematične osnove, nato pa definiramo podatkovno hierarhijo in predstavimo nekaj primerov. V drugem delu se seznanimo z gručenjem in njegovimi metodami, nato se podrobneje seznanimo z združevalno hierarhično metodo gručenja. Implementiramo tri hierarhične metode in jih med seboj primerjamo. V tretjem, zadnjem delu okaraktiziramo grafe podatkovnih hierarhij, ki nastanejo med najpogosteje uporabljenimi algoritmi gručenja.

Del I

Delne urejenosti

Poglavje 1

Delne urejenosti in njihovi grafi pokritij

1.1 Delne urejenosti

Pravi predmet moderne matematike ne predstavljajo reči, temveč lastnosti teh reči in odnosi, ki veljajo med njimi (Prijatelj, [17]). V Slovarju tujk Cankarjeve založbe je beseda relacija, ki izhaja iz latinščine, definirana kot odnos, razmerje ali vračanje, ponavljanje. Sama beseda ima več pomenov, eden izmed njih je povezava med dvema množicama števil v matematiki. Tako kot v vsakdanjem življenju (oče in sin, mož in žena,...) lahko različne odnose opazimo tudi med matematičnimi pojmi:

Število a je VEČJE od števila b.
Premica p je PRAVOKOTNA na premico q.

V matematiki relacija pomeni odnos, zvezo ali razmerje med dvema objektoma.

Definicija 1.1 *Množica R je (dvomestna)* relacija*, če je vsak njen element urejen par oziroma velja*

$$R \text{ je relacija} \Leftrightarrow \forall x \in R, \exists\, u, v : x = (u, v)$$

Zgled. Poglejmo si primera relacije:
$A = \{1, 2, 3, 4\}$, pri čemer je $R = \{(1, 2), (2, 3), (3, 1)\}$.
$B = \mathbb{N}$, pri čemer je $R = \{(u, v) | u, v \in \mathbb{N} \wedge u \leq v\}$.

Ker pogosto srečujemo določene relacije, imajo le-te upravičeno posebna imena. Zaradi kasnejše uporabe definirajmo naslednje:

Definicija 1.2 *Relacija R je* refleksivna *ali* povratna*, če je vsak element v relaciji sam s seboj. Torej velja*

$$\forall v \in V : (v, v) \in R. \tag{1.1}$$

Definicija 1.3 *Relacija R je* simetrična *ali* vzajemna*, če lahko elementa relacije zamenjamo: če je element u v relaciji R z elementom v, je element v v relaciji R z u. Torej velja*

$$\forall u, v \in V : (u, v) \in R \Rightarrow (v, u) \in R. \tag{1.2}$$

Definicija 1.4 *Relacija R je* antisimetrična*, če za poljubna dva elementa iz množice V velja, da iz $(u, v) \in R$ in $(v, u) \in R$ sledi, da $u = v$. Torej velja*

$$\forall u, v \in V : (u, v) \in R \wedge (v, u) \in R \Rightarrow u = v. \tag{1.3}$$

Definicija 1.5 *Relacija R je* tranzitivna *ali* prehodna*, če za poljubne tri elemente iz množice V velja, da iz $(u, v) \in R$ in $(v, w) \in R$ sledi, da $(u, w) \in R$. Torej velja*

$$\forall u, v, w \in V : (u, v) \in R \wedge (v, w) \in R \Rightarrow (u, w) \in R. \tag{1.4}$$

Za določene relacije lahko velja več navednih lastnosti. V takem primeru več lastnosti skupaj predstavlja nov tip relacije.

Definicija 1.6 *Naj bo V množica točk. Relacijo $R \subseteq V \times V$ imenujemo* delna urejenost*, če je relacija R*

(i) refleksivna: $\forall v \in V, (v, v) \in R$,

(ii) antisimetrična: $\forall u, v \in V$ velja: če $(u, v) \in R$ in $(v, u) \in R$, potem $u = v$,

(iii) tranzitivna: $\forall u, v, w \in V$ velja: če $(u, v) \in R$ in $(v, w) \in R$, potem $(u, w) \in R$.

Namesto oznake $(u, v) \in R$ lahko pišemo tudi uRv. V primeru delne urejenosti pa namesto znaka za relacijo pogosto R uporabljamo znak $\leq$. Na primer $x \leq y$ (beremo x *je pod* y), če je xRy in $x < y$, če $x \leq y$ in $x \neq y$.

Zgled. Med primere delnih urejenosti spada relacija "manjše ali enako" na poljubni podmnožici realnih števil.

Delno urejena množica je *linearno urejena*, če veljajo vse lastnosti delne urejenosti (definicija 1.6), ter za vsak $u, v \in V$ velja tudi $u \leq v$ ali $v \leq u$. Zadnjo lastnost imenujemo *sovisnost*. Linearna urejenost je torej sovisna delna urejenost.

Zgled. Relacija "manjše ali enako" linearno ureja poljubno podmnožico $\mathbb{R}$.

Definicija 1.7 *Naj bo $P = (V, \leq)$ delno urejena množica. Delno urejena množica P je* končna*, če je množica V končna.*

V primeru končne in tudi neskončne delno urejene množice v delnih urejenostih nastopajo posebni elementi.

Definicija 1.8 *Naj bo V delno urejena z relacijo $\leq$. Pravimo, da je*

(i) m minimalen *v V, če za vsak element v iz V velja: $v \leq m \Rightarrow v = m$.*

(ii) M maksimalen *v V, če za vsak element v iz V velja: $M \leq v \Rightarrow v = M$.*

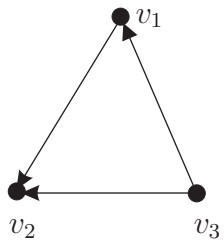
(iii) a prvi *v V, če za vsak element v iz V velja: $a \leq v$.*

(iv) z zadnji *v V, če za vsak element v iz V velja: $v \leq z$.*

1.2 Grafi delnih urejenosti

Definicija 1.9 *Par $G = (V, E)$ je* graf *na množici vozlišč $V(G)$ in z množico povezav $E(G)$, pri čemer je V končna neprazna množica, E pa poljubna družina dvoelementarnih podmnožic množice V.*

Vsak graf je torej določen takoj, ko poznamo njegova vozlišča in vemo, kateri pari vozlišč so med seboj povezani. V primeru, da imata obe krajišči povezave isto vozlišče, dobimo *zanko*.
Včasih je pomembno, kako je posamezna povezava v grafu usmerjena. V tem primeru povezavo predstavljamo kot puščico, graf z usmejenimi povezavami pa imenujemo *usmerjeni graf* ali *digraf*.



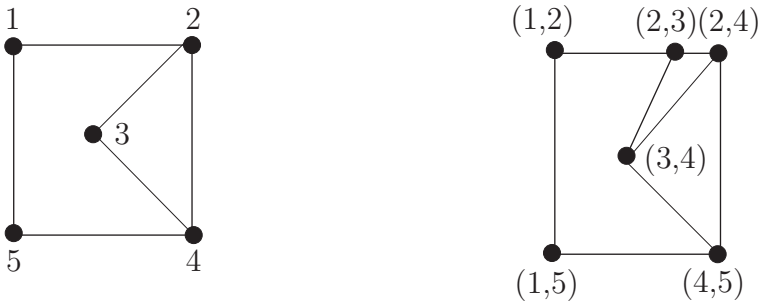
Slika 1.1: Primer usmerjenega grafa.

Definicija 1.10 Usmerjeni graf D *(slika 1.1) je sestavljen iz množice vozlišč* $V(D)$ *in iz seznama urejenih parov vozlišč oziroma iz usmerjenih povezav (lokov)* $A(D)$*. Če sta vozlišči* v_1 *in* v_2 *v usmerjenem grafu* D*, potem je povezava* v_1v_2 *usmerjena od* v_1 *do* v_2 *oziroma gre iz* v_1 *v* v_2*.*

Na sliki 1.1 sta vozlišči v_1 in v_2, povezava pa je usmerjena od v_1 do v_2 oziroma gre iz v_1 v v_2. V primeru, da nas usmerjenost v grafu ne bi zanimala, bi dobili *neusmerjeni graf*.

Vozlišči, ki sta povezani s povezavo, imenujemo *sosednji*. S pomočjo povezav, ki povezujejo vozlišča v grafu, lahko dobimo *graf povezav*, ki predstavlja vse povezave v posameznem grafu.

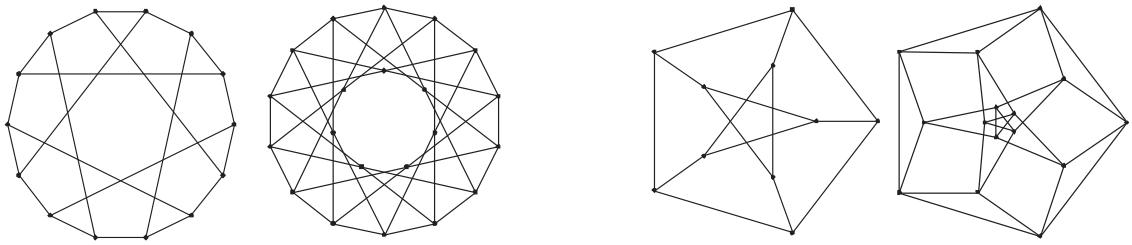
Definicija 1.11 Graf povezav $L(G)$ *(slika 1.2) ima za množico vozlišč množico povezav grafa* G*. Dve vozlišč med seboj povežemo, če imata povezavi v* G *skupno krajišče.*



Slika 1.2: Graf G (levo) in njegov graf povezav L(G) (desno)

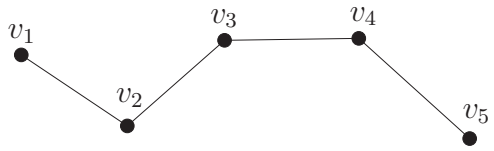
Vsako vozlišče v grafu povezav $L(G)$ predstavlja posamezno povezavo njemu ustreznega grafa G. Dva znana grafa in njuna grafa povezav sta *Heawoodov graf* in njegov štirivalenten graf povezav, ter *Petersenov graf* in njegov graf povezav.

Vpeljali bomo tri pomembne vrste grafov, ki jih bomo uporabljali v nadaljevanju. V teoriji grafov se običajno srečujemo s tremi pomembnimi oblikami, kot so *pot*, *cikel* in *drevo*.



(1a) (1b) (2a) (2b)

Slika 1.3: Heawoodov graf (1a) in njegov graf povezav (1b) ter Petersenov graf (2a) in njegov graf povezav (2b) (vir: http://www.pef.upr.si/MARA/2/RaMa/predavanja.html)



Slika 1.4: Pot P_5.

Definicija 1.12 Pot *(slika 1.4) na n vozliščih, ki jo označimo s* P_n*, je graf, ki ima vozlišča* $v_1, v_2, ..., v_n$ *in povezave* $v_1v_2, v_2v_3, ..., v_{n-1}v_n$*. Je preprost graf, ki ima n vozlišč in* $n-1$ *povezav. Stopnjo* 2 *ima* $n-2$ *vozlišč, preostali dve vozlišči, ki predstavljata krajišči poti, pa imata stopnjo* 1*.*

$$V(P_n) = \{v_1, v_2, ..., v_n\} \tag{1.5}$$

$$E(P_n) = \{v_iv_{i+1} | 1 \leq i < n\} \tag{1.6}$$

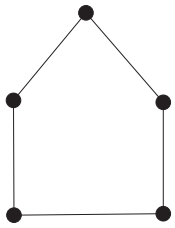
Definicija 1.13 Cikel *(slika 1.5) na n vozliščih, ki ga označimo s* C_n*, dobimo iz poti* P_n*,* $n \geq 3$*, tako da dodamo povezavo* v_1v_n*. Cikel ima n povezav.*

$$V(C_n) = \{v_1, v_2, ..., v_n\} \tag{1.7}$$

$$E(C_n) = \{v_iv_{i+1} | 1 \leq i < n\} \cup \{v_i, v_n\} \tag{1.8}$$

Definicija 1.14 *Graf je* povezan*, če obstaja pot med poljubnim parom vozlišč.*

Definicija 1.15 Drevo *je povezan graf brez ciklov.*



Slika 1.5: Cikel C_5.

Poglejmo si nekaj grafov, ki jih lahko definiramo na delno urejeni množici.

Definicija 1.16 *Naj bo $P = (V, \leq)$ končna delno urejena množica. Potem je $G_P = (V, E_P)$ z $xy \in E_P$, če $x < y$ ali $y < x$,* graf primerljivosti *delno urejene množice P. $G = (V, E)$ je graf primerljivosti, če obstaja delno urejena množica P, tako da je $G \sim G_P$.*

Obstaja tudi povezava med usmerjenim in neusmerjenim grafom:

Definicija 1.17 *Naj bo $G = (V, E)$ neusmerjeni graf, pri čemer je V množica vozlišč grafa G in E množica neusmerjenih povezav. Potem je usmerjeni graf $G' = (V, E')$* orientacija *grafa G, če za vse povezave $xy \in E$ velja ena od usmerjenosti ali $(x, y) \in E'$ ali $(y, x) \in E'$ ter za vse pare $(x, y) \in E'$ velja $xy \in E$.*
Usmerjeni graf G' je tranzitivna orientacija neusmerjenega grafa G, če je E' tranzitivna dvočlena relacija nad množico točk V.
Usmerjeni graf G' je aciklična orientacija neusmerjenega grafa G, če v usmerjenem grafu G' ni usmerjenih ciklov.

Če je $G' = (V, E')$ usmerjeni graf, potem je $G = (V, E)$ z $xy \in E$ če in samo če $(x, y) \in E'$ ali $(y, x) \in E'$ osnoven neusmerjeni graf G.

Opomba 1.18 *Po definiciji je neusmerjeni graf G* graf primerljivosti*, če in samo če ima graf G aciklično tranzitivno orientacijo.*

Vsaka relacija delne urejenosti nad V enolično določa relacijo *pokritja*.

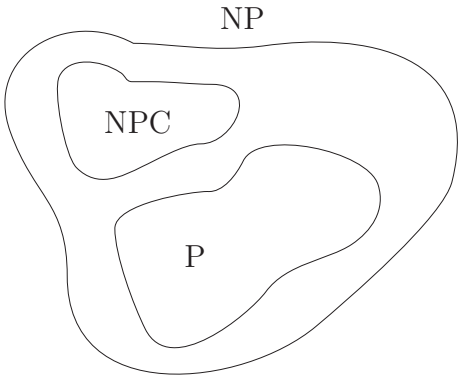
Definicija 1.19 *Naj bo $P = (V, <)$. Element $x \in V$ je* pokrit *z $y \in V$, če je $x < y$ in ne obstaja tak $z \in V$, da velja $x < z < y$.* Graf pokritij *delno urejene množice $(P, \leq)$ je neusmerjeni graf $G = (V, E)$, tako da $xy \in E$, če x pokriva y ali y pokriva x.*

Slika 1.6: Primer grafa pokritij.

Zgled. Za vozlišče v_2 (slika 1.6) je prvi pogoj izpolnjen za vsa tri vozlišča ($v_2 < v_3$, $v_2 < v_4$ in $v_2 < v_5$), ki ležijo v ravnini nad vozliščem v_2. Vendar le za vozlišče v_3 velja tudi, da ne obstaja takšno vozlišče, ki leži med v_2 in v_3 oziroma ne obstaja $z \in V$: $v_2 < z < v_3$. Vozlišče v_3 torej pokriva vozlišče v_2.

Trditev 1.20 (Brightwell, Brandstädt in Le ter Spinrad, [2]) *Problem, ali je graf G graf pokritij za kakšno delno urejeno množico, je NP-poln problem.*

Posamezen problem, kjer znotraj množice vseh možnih podatkov iščemo tiste z neko določeno lastnostjo, imenujemo *odločitveni problem*. Odločitveni problem je v *razredu P*, če za njegovo reševanje obstaja polinomski algoritem. V *razredu NP* pa so odločitveni problemi, ki jih je mogoče rešiti v polinomskem času z *nedeterminističnim algoritmom*. Obstaja torej nek polinomski algoritem, ki v primeru pozitivnega odgovora preveri pravilnost odgovora. Očitno je torej, da je P $\subseteq$ NP, vendar velja domneva P $\neq$ NP (Žerovnik, [31]).



Slika 1.7: Razredi odločitvenih problemov

V razredu NP pomembno podmnožico predstavljajo *NP-polni problemi*, ki jih označujemo z NPC zaradi angl. NP complete. Za NP probleme velja naslednje: če bi obstajal polinomski algoritem, s katerim bi lahko problem rešili, potem bi lahko v polinomskem času rešili vse probleme iz NP, torej bi bilo P = NP. Velja pa tudi obratno: če bi dokazali, da za enega od NP-polnih problemov ni mogoče zapisati polinomskega algoritma, potem bi to veljalo za vse probleme iz NPC. V razredu NP-polnih problemov so mnogi težki problemi, zato je zelo verjetno, da velja P $\neq$ NP (Žerovnik, [31]).

Delno urejenost lahko predstavimo s *Hassejevim diagramom*: elemente množice predstavimo s točkami v ravnini tako, da za poljubna elementa a in b velja

(i) če $a \leq b$, potem b leži nad a,

(ii) a in b povežemo, če b pokriva a.

Definicija 1.21 Hassejev diagram *D delno urejene množice $P = (V, <)$ je predstavitev grafa pokritij P v ravnini tako, da če je $x < y$, potem x v ravnini leži nižje od y.*

Pri zgledu na sliki 1.6 tako opazimo, da vozlišče v_2 leži v ravnini nižje kot vozlišče v_3. Podobno velja za ostale primerljive pare. Slika 1.6 torej predstavlja Hassejev diagram.

Ena izmed lastnosti delno urejene množice je tudi *N-prostost*:

Definicija 1.22 *Delno urejena množica P je* N-prosta*, če diagram delno urejene množice P ne vsebuje štirih vozlišč a, b, c, d tako da je množica povezav v diagramu med a, b, c, d natanko $(a, b), (c, b), (c, d)$.*

Zgled. N je delno urejena množica (slika 1.8), ki vsebuje vozlišča a, b, c, d, tako da $a < c$, $b < c$, $b < d$, b neprimerljiv z a, a neprimerljiv z d in d neprimerljiv s c.

Slika 1.8: Primer delno urejene množice, ki vsebuje N (levo) in primer N-proste delno urejene množice P (desno).

S postopkom, imenovanim *subdivizija*, lahko vedno ustvarimo N-prosto delno urejeno množico. Subdivizijo dobimo, če na povezan graf G dodamo točke stopnje 2.

Delna urejenost je torej pomemben odnos med rečmi v moderni matematiki. Pomembna in uporabna lastnost relacij pa je tudi *ekvivalentnost*, ki si jo bomo pogledali v naslednjem poglavju.

Poglavje 2

Podatkovna hierarhija in njen graf pokritij

V tem poglavju si bomo najprej pogledali definicijo ekvivalentnosti ter primere ekvivalenčne relacije. Nato bomo definirali relacijo finejši, seznanili se bomo s pojmom podatkovna hierarhija in definirali njen graf pokritij. Ob koncu poglavja se bomo posvetili še grafom, ki lahko predstavljajo grafe podatkovnih hierarhij.

Definicija 2.1 *Relacija $R \subseteq V \times V$ je* ekvivalenčna relacija*, če je*

(i) refleksivna: $\forall v \in V$ velja: $(v, v) \in R$,

(ii) simetrična: $\forall u, v \in V$ velja: če $(u, v) \in R$, potem $(v, u) \in R$,

(iii) tranzitivna: $\forall u, v, w \in V$ velja: če $(u, v) \in R$ in $(v, w) \in R$, potem $(u, w) \in R$.

Je dvočlena relacija in v matematiki običajno označena s simbolom $\sim$. Z uporabo tega znaka lahko zgornje lastnosti zapišemo kot:

(i) za vsak a velja $a \sim a$,

(ii) iz $a \sim b$ sledi $b \sim a$,

(iii) iz $a \sim b$ in $b \sim c$ sledi $a \sim c$.

Zgled. Preprost primer ekvivalenčne relacije je *vzporednost* premic. Dober primer predstavlja tudi relacija *kongruence* po modulu k v množici celih števil, kjer je k poljubno celo število, različno od 0.

$$a \equiv b \pmod{k} \Leftrightarrow a - b \text{ deljiv z } k.$$

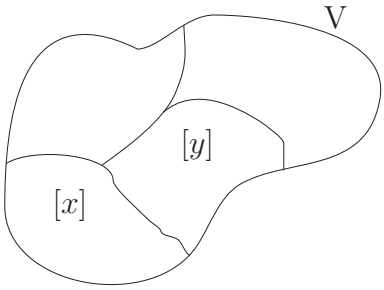
Najpomembnejša lastnost ekvivalenčne relacije je, da množico V razdeli na *neprazne in paroma tuje podmnožice, katerih unija je osnovna množica* V. Tem podmnožicam rečemo *ekvivalenčni razredi*:

Definicija 2.2 *Naj bo R ekvivalenčna relacija na V. Za $x \in V$ z $R[x]$ označimo množico elementov, ki so z x v relaciji R, torej $R[x] = \{y \in V \mid xRy\}$. To podmnožico imenujemo* ekvivalenčni razred *elementa x glede na ekvivalenčno relacijo R.*

Ekvivalenčni razred torej dobimo , da iz množice V izberemo poljubni element x in mu priredimo tisto podmnožico množice V, v kateri so natanko kotisti elementi, ki so v relaciji R z x. Velja pa tudi naslednje:

Trditev 2.3 *Če sta x in $y \in V$, potem velja natanko ena od možnosti: $R[x] = R[y]$ ali $R[x] \cap R[y] = \emptyset$.*

Dokaz. Naj bosta $x, y \in V$. Velja natanko ena od možnosti: $y \in [x]$ ali $y \notin [x]$.



Slika 2.1: Ekvivalenčni razredi

(i) Če $y \in [x]$ potem $[x] = [y]$.
Po definiciji 2.2 velja: $y \in [x] \Rightarrow xRy$. Naj bo z poljuben element iz $[x]$. Potem sledi xRz in zaradi simetričnosti ekvivalenčne relacije sledi zRx. Če velja zRx in xRy potem zaradi tranzitivnosti ekvivalenčne relacije velja tudi zRy. Če pa je z v relaciji R z y, potem je zaradi simetričnosti tudi yRz, iz česar sledi $z \in [y]$. Tako torej velja $[x] \subseteq [y]$. Podobno dokažemo tudi $[y] \subseteq [x]$. Če velja $[x] \subseteq [y]$ in $[y] \subseteq [x]$, potem je $[x] = [y]$.

(ii) Če $y \notin [x]$ potem $[x] \cap [y] = \emptyset$.
Predpostavimo nasprotno. Naj obstaja tak $z \in [x] \cap [y]$. Če je z element preseka ekvivalenčnih razredov, potem leži v vsakem ekvivalenčnem razredu in je torej $z \in [x]$ in $z \in [y]$. Potem je po definiciji 2.2 xRz in yRz. Zaradi simetričnosti iz yRz sledi zRy. Po tranzitivnosti pa zaradi xRz in zRy sledi xRy oziroma sledi $y \in [x]$.
$y \in [x]$ je v protislovju z domnevo $y \notin [x]$.

Iz (i) in (ii) sledi trditev. □

Posledica 2.4 *Množica vseh ekvivalenčnih razredov predstavlja razbitje množice V.*

Zgled. Nadaljujmo s prejšnjim zgledom.
Ekvivalenčni razredi so v primeru relacije kongruentnosti *razredi ostankov* po modulu k. V vsakem ekvivalenčnem razredu so torej vsa tista cela števila, ki dajo pri deljenju s k isti ostanek. Očitno je, da je takših razredov natanko k.
Poglejmo si primer kongruentnosti po modulu 2. Poljubni števili m in n sta v relaciji, če pri deljenju s številom 2 dobimo isti ostanek. Tako množica naravnih števil razpade na množico sodih $\{2, 4, 6, ...\}$ in množico lihih števil $\{1, 3, 5, ...\}$.

Vpeljimo še pojem *finosti* dveh relacij:

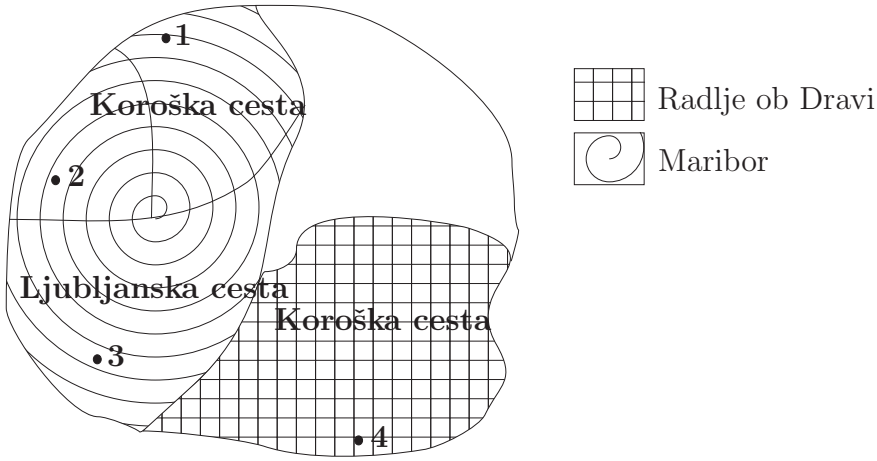
Definicija 2.5 *Ekvivalenčna relacija R_1 je* finejša *od ekvivalenčne relacije R_2 (oziroma s simbolom $R_1 \preceq R_2$), če za vsak ekvivalenčni razred $R_1[x]$ velja, da je vsebovan v ekvivalenčnem razredu $R_2[x]$, $R_1[x] \subseteq R_2[x]$.*

Zgled. Imamo naslove:

1. Koroška cesta 158, 2000 Maribor, Slovenija
2. Koroška cesta 160, 2000 Maribor, Slovenija
3. Ljubljanska cesta 13, 2000 Maribor, Slovenija
4. Koroška cesta 17, 2360 Radlje ob Dravi, Slovenija

Na podlagi zgleda definiramo ekvivalenčno relacijo: $\sim_U$ pomeni, da sta naslova v isti ulici, $\sim_M$, da sta naslova v istem mestu in $\sim_D$ v isti državi.

Potem za naš zgled velja:



Slika 2.2: Množica naslovov.

$$\begin{array}{lll} 1 \sim_U 2 & 1 \not\sim_U 3 & 1 \not\sim_U 4 \\ 1 \sim_M 2 & 1 \sim_M 3 & 1 \not\sim_M 4 \\ 1 \sim_D 2 & 1 \sim_D 3 & 1 \sim_D 4 \end{array}$$

Opazimo, da če sta naslova v relaciji ulice, sta tudi v relaciji mesta in države oziroma če sta v relaciji mesta, sta tudi v relaciji države. Vendar pa ni nujno, da če sta naslova v relaciji države, da sta tudi v relaciji ulice oziroma mesta. Na primer naslov 1 in naslov 4 imata sicer isto ime ulice, vendar to ni ista ulica, saj je v različnih mestih. Tako sta le v relaciji države in ne tudi v relaciji mesta oziroma ulice.

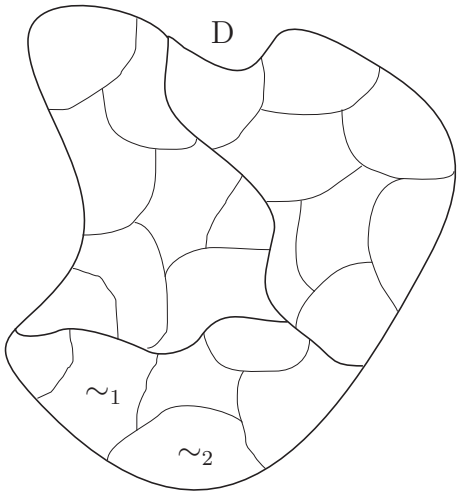
Slika 2.3: Primer ekvivalentnosti naslovov.

Po definiciji 2.5 potem velja $\sim_U \preceq \sim_M \preceq \sim_D$.

Trditev 2.6 *Naj bo D množica podatkov in V množica ekvivalenčni relacij nad D. Relacija $\preceq$ delno ureja množico ekvivalenčnih relacij V.*

Dokaz. Naj bo D množica podatkov (slika 2.4).

Da bo $\preceq$ delna urejenost, je potrebno dokazati lastnosti iz definicije 1.6:



Slika 2.4: Relacija $\preceq$ nad množico podatkov D.

(i) Refleksivnost: $R_1[x] \subseteq R_1[x]$.
Vsak ekvivalenčni razred je vsebovan sam v sebi, torej po definiciji 2.5 velja $R_1 \preceq R_1$.

(ii) Antisimetričnost: $R_1 \preceq R_2$ in $R_2 \preceq R_1 \Rightarrow R_1 = R_2$
$\Leftrightarrow R_1 \preceq R_2$ sledi $\forall x : R_1[x] \subseteq R_2[x]$.
Če je ekvivalenčni razred $R_1[x]$ vsebovan v ekvivalenčnem razredu $R_2[x]$, potem je vsak element ekvivalenčnega razreda $R_1[x]$ v ekvivalenčnem razredu $R_2[x]$. In če je vsak element $R_2[x]$ v $R_1[x]$, potem sta ekvivalenčna razreda lahko le enaka, oziroma je $R_1[x] = R_2[x]$. Ker so ekvivalenčni razredi za vsak element enaki, sta po definiciji 2.5 tudi relaciji enaki.

(iii) Tranzitivnost: $R_1[x] \subseteq R_2[x]$ in $R_2[x] \subseteq R_3[x] \Rightarrow R_1[x] \subseteq R_3[x]$.
Če je vsak element ekvivalenčnega razreda $R_1[x]$ v ekvivalenčnem razredu $R_2[x]$ in je vsak element ekvivalenčnega razreda $R_2[x]$ v ekvivalenčnem razredu $R_3[x]$, potem je vsak element iz $R_1[x]$ v $R_3[x]$. Po definiciji 2.5 sledi $R_1 \preceq R_3$.

Iz (i), (ii) in (iii) sledi zgornja trditev. □

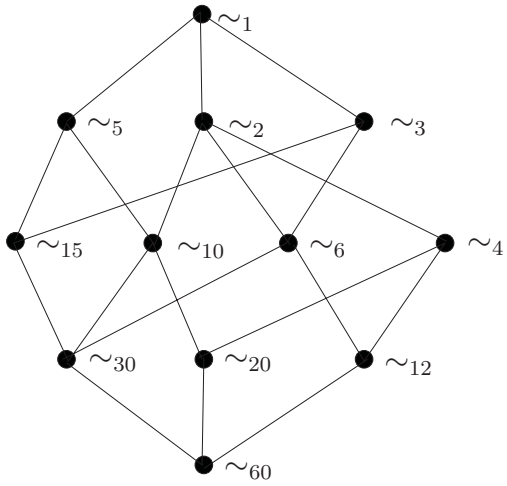
Definicija 2.7 *Naj bo D množica podatkov in V množica ekvivalenčnih relacij nad D. Par (D, V) imenujemo* podatkovna hierarhija.

Iz trditve 2.6 sklepamo, da je vsaka podatkovna hierarhija delno urejena z relacijo $\preceq$. Zato lahko definiramo:

Definicija 2.8 Graf pokritij *delne urejenosti* $(V, \preceq)$ *imenujemo reducirani graf podatkovne hierarhije.*

Zgled. Naj bo D množica naslovov in V množica ekvivalenčnih relacij $\sim_U, \sim_M$ in $\sim_D$ nad naslovi. Par $H = (D, V)$ predstavlja podatkovno hierarhijo nad množico naslovov.

Zgled. Naj par $H_{60} = (\mathbb{N}, V_{60})$ predstavlja podatkovno hierarhijo vseh deliteljev števila 60. Podatkovno hierarhijo nam generirajo kongruentnosti po modulih 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60 (slika 2.5).



Slika 2.5: Podatkovna hierarhija $H_{60} = (\mathbb{N}, V_{60})$.

Vprašanje 2.9 *Vprašamo se, kateri grafi lahko nastopajo kot grafi podatkovnih hierarhij.*

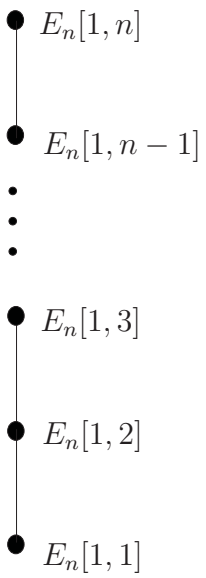
Zaradi kasnejše uporabe vpeljimo nove oznake:

$$D_n = \{1, 2, ..., n\}. \tag{2.1}$$

$$E_n[i, j] = \{[i, j] \cap \mathbb{N}\} \cup \{\{k\} \,|k \notin [i, j]\}. \tag{2.2}$$

$$V_n^p = \{E_n[1, k] \,|k = 1, 2, ..., n\}. \tag{2.3}$$

Trditev 2.10 *Za $n \geq 1$ je graf podatkovne hierarhije $H_n = (D_n, V_n^p)$* pot na n vozliščih P_n.



Slika 2.6: Za poljuben n.

Dokaz. Pot P_n ima naslednjo obliko (slika 2.6):

Pri tem je po vpeljavi oznak 2.1, 2.2 in 2.3
$E_n[1,1] = \{\{1\},\{2\},\{3\},...,\{n-1\},\{n\}\}$,
$E_n[1,2] = \{\{1,2\},\{3\},...,\{n-1\},\{n\}\}$,
$E_n[1,3] = \{\{1,2,3\},\{4\},...,\{n-1\},\{n\}\}$,
$E_n[1,n-1] = \{\{1,2,3,...,n-1\},\{n\}\}$ in
$E_n[1,n] = \{\{1,2,3,...,n-1,n\}\}$.

Dokazati je torej potrebno, da $E_n[1,l+1]$ pokriva $E_n[1,l]$ oziroma je

$$E_n[1,l] \preceq E_n[1,l+1]$$

za poljuben $l \in \{1,2,3,...,n\}$.

Vidimo, da obstajata dve možnosti:

1. če je $l \in \{[i,j] \cap \mathbb{N}\}$ in $l \leq j$, potem iz tega sledi, da je l znotraj množice $\{1,...,j\}$ in velja tudi, da je $l \in \{1,...,j,j+1\}$, torej velja $l \in \{[i,j+1] \cap \mathbb{N}\}$. Če je torej l znotraj ene množice z več elementi, bo tudi na višjem nivoju hierarhije znotraj te iste množice z enim elementom več;

2. v primeru, da je l na j-tem nivoju podatkovne hierarhije edini element v svoji množici, torej velja $\{\{1,2\},\{3\},...,\{l\},...,\{n-1\},\{n\}\}$. V tem primeru za naslednji nivo obstajata ponovno dve možnosti:

2.1. če je $l = j + 1$ oziroma velja $\{\{1, 2, ..., j\}, \{l\}, ..., \{n-1\}, \{n\}\}$, potem bo na naslednjem koraku l v množici z večimi elementi in ne bo več le edini predstavnik svoje množice. Torej bo iz $l \in \{\{k\} | k \notin [i, j]\}$ in $l = j + 1$ sledilo $l \in \{[i, j + 1] \cap \mathbb{N}\}$;

2.2. v drugem primeru pa bo $l > j+1$, torej je že na j-tem nivoju l edini predstavnik svoje množice in velja $\{\{1, 2, ..., j\}, \{j + 1\}, ..., \{l\}, ..., \{n-1\}, \{n\}\}$ Tudi na $(j + 1)$-koraku bo l še vedno edini predstavnik svoje množice in bo veljalo $\{\{1, 2, ..., j, j + 1\}, ..., \{l\}, ..., \{n-1\}, \{n\}\}$. V tem primeru je torej $l \in \{\{k\} | k \notin [i, j]\}$ in $l > j + 1$. Iz tega sledi $l \in \{\{k\} | k \notin [i, j + 1]\}$;

Po tranzitivnosti je $E_n[1, k][l] \subseteq E_n[1, j][l]$ za $\forall j \geq k$, $\forall l$, torej $E_n[1, k] \succeq E_n[1, j]$.

Za pot P_n torej velja zgornja trditev.

□

Trditev 2.11 *Za $n \geq 4$ je graf podatkovne hierarhije $H_n = (D_n, V_n^p)$* cikel na n vozliščih C_n.

Dokaz. Predpostavimo, da je n sodo število.

Pri ciklu C_n bomo na Trditev 2.11 dokazali na podoben način kot pri poti P_n. Dokazati moramo, da $E_n[1, j] \preceq E_n[1, j + 1]$ za $j \in \left\{1, 2, ..., \frac{n}{2}\right\}$ za desno stran cikla in $E_n[l, \frac{n}{2}] \preceq E_n[l-1, \frac{n}{2}]$ za $l \in \left\{2, 3, ..., \frac{n}{2}\right\}$ za levo stran cikla. Nato bomo le še dokazali, da je $E_n[\frac{n}{2}, \frac{n}{2}] = E_n[1, 1]$ in zgornja trditev bo dokazana.

Vemo, da velja $E_n[1, j] \preceq E_n[1, j + 1]$, saj smo to že dokazali pri poti. Torej na enak način za

$$E_n[1, \frac{n}{2}] = \left\{\left\{1, 2, 3, ..., \frac{n}{2}\right\}, \left\{\frac{n}{2} + 1\right\}, ..., \{n\}\right\}$$

$$E_n[2, \frac{n}{2}] = \left\{\{1\}, \left\{2, 3, ..., \frac{n}{2}\right\}, \left\{\frac{n}{2} + 1\right\}, ..., \{n\}\right\}$$

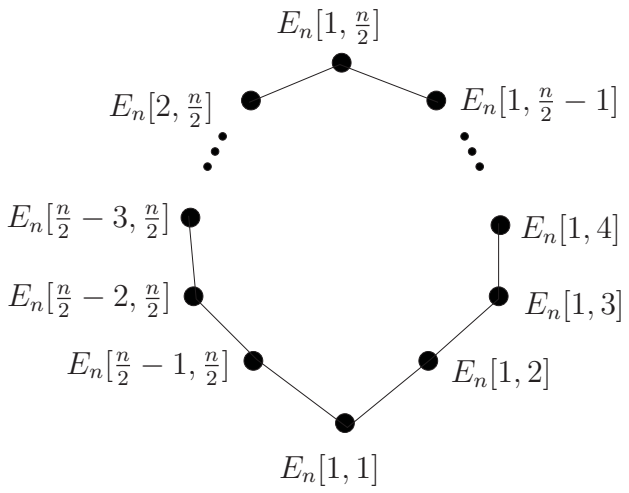
$$E_n[\frac{n}{2} - 3, \frac{n}{2}] = \left\{\{1\}, ..., \left\{\frac{n}{2} - 3, \frac{n}{2} - 2, \frac{n}{2} - 1, \frac{n}{2}\right\}, \left\{\frac{n}{2} + 1\right\}, ..., \{n\}\right\}$$

$$E_n[\frac{n}{2} - 2, \frac{n}{2}] = \left\{\{1\}, ..., \left\{\frac{n}{2} - 2, \frac{n}{2} - 1, \frac{n}{2}\right\}, \left\{\frac{n}{2} + 1\right\}, ..., \{n\}\right\}$$

$$E_n[\frac{n}{2} - 1, \frac{n}{2}] = \left\{\{1\}, ..., \left\{\frac{n}{2} - 1, \frac{n}{2}\right\}, \left\{\frac{n}{2} + 1\right\}, ..., \{n\}\right\}$$

$$E_n[\frac{n}{2}, \frac{n}{2}] = \left\{\{1\}, \{2\}, ..., \left\{\frac{n}{2}\right\}, \left\{\frac{n}{2} + 1\right\}, ..., \{n\}\right\}$$

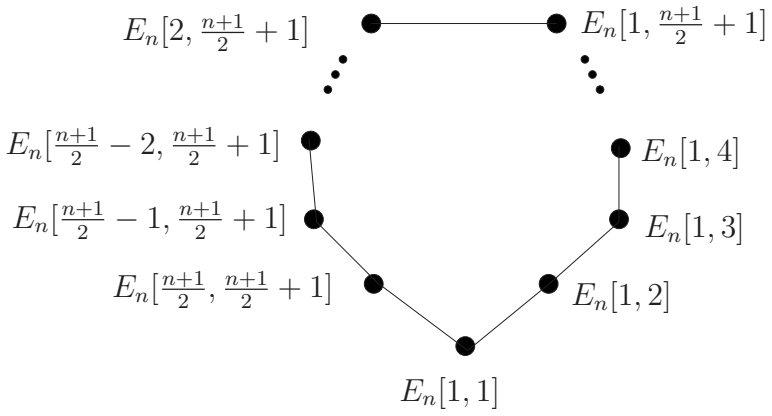
vemo, da je
$E_n[\frac{n}{2}, \frac{n}{2}] \preceq E_n[\frac{n}{2} - 1, \frac{n}{2}] \preceq E_n[\frac{n}{2} - 2, \frac{n}{2}] \preceq E_n[\frac{n}{2} - 3, \frac{n}{2}] \preceq ... \preceq E_n[2, \frac{n}{2}] \preceq E_n[1, \frac{n}{2}]$.



Slika 2.7: Cikel na n vozliščih, kjer je n sodo število.

Ker je $E_n[\frac{n}{2}, \frac{n}{2}] = E_n[1, 1]$, saj je vsak element predstavnik svoje množice, potem velja tudi $E_n[1, 1] \preceq E_n[\frac{n}{2} - 1, \frac{n}{2}] \preceq E_n[\frac{n}{2} - 2, \frac{n}{2}] \preceq E_n[\frac{n}{2} - 3, \frac{n}{2}] \preceq ... \preceq E_n[2, \frac{n}{2}] \preceq E_n[1, \frac{n}{2}]$.

Torej za n je sodo število velja zgornja trditev. Pa si poglejmo še, ali velja tudi za liho število n.

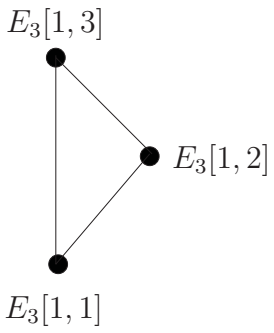


Slika 2.8: Cikel na n vozliščih, kjer je n liho število.

Opazimo, da zgornja trditev prav tako drži za n je liho število, vendar zgornja elementa v ciklu, $E_n[1, \frac{n+1}{2} + 1]$ in $E_n[2, \frac{n+1}{2} + 1]$, nista primerljiva, kot v ciklu s sodim številom vozlišč n.

□

Opomba 2.12 *Za tricikel oziroma cikel na* 3 *vozliščih trditev 2.11 ne drži, saj bi dobili tranzitivne povezave, ki pa jih v grafu pokritij ni.*



Slika 2.9: Tricikel

Tranzitivne povezave so povezave, ki jih graf pokritij ne vsebuje.

Del II

Hierarhično gručenje podatkov

Poglavje 3

Odkrivanje znanja v podatkih in razvrščanje v skupine

V času množične uporabe računalnikov so se razširile tudi računalniške baze podatkov. S tem se je pojavila potreba po orodjih, ki bi iz množice podatkov pridobila uporabno znanje. Razvila se je nova informacijska tehnologija - odkrivanje znanja in podatkovno rudarjenje - ki nam omogoča hranjenje, obdelavo in analizo podatkov. V teh podatkih tako odkrivamo nove zakonitosti in jih poskušamo uporabiti na več področjih, med drugim tudi za napovedovanje prihodnjih dogodkov.

V tem poglavju si bomo pogledali, kaj sploh je odkrivanje znanja v podatkih ter iz katerih faz je postopek oziroma proces odkrivanja sestavljen. Bolj podrobno si bomo pogledali eno izmed faz odkrivanja znanja v podatkih, t.j. podatkovno rudarjenje.

Odkrivanje znanja v podatkih (angl. Knowledge Discovery in Databases, KDD) je netrivialen proces odkrivanja implicitnega, do sedaj neznanega in potencialno uporabnega znanja iz podatkov (Han in Kamber, [8]). To pa je področje, ki temelji na metodah in algoritmih umetne inteligence ter njenega ožjega področja *strojnega učenja* (Kononenko, [13]). Strojno učenje je ena izmed bolj učinkovitih možnosti za obdelavo velike količine podatkov. Je skupek metod in postopkov, s katerimi lahko iz velike količine podatkov pridobimo oziroma izluščimo znanje in informacije (Štajdohar in Standeker, [23]).

Proces odkrivanja uporabnih vzorcev v podatkih ima več izrazov - podatkovno rudarjenje (angl. data mining), izkopavanje znanja (angl. knowledge extraction), žetev informacij (angl. information harvesting), odkrivanje informacij (angl. information discovery), izkopavanje znanja (angl. knowledge mining), obdelava podatkovnih vzorcev (angl. data pattern processing) itd. Kot najprimernejši izraz je bil leta 1989 na prvi delavnici o odkrivanju

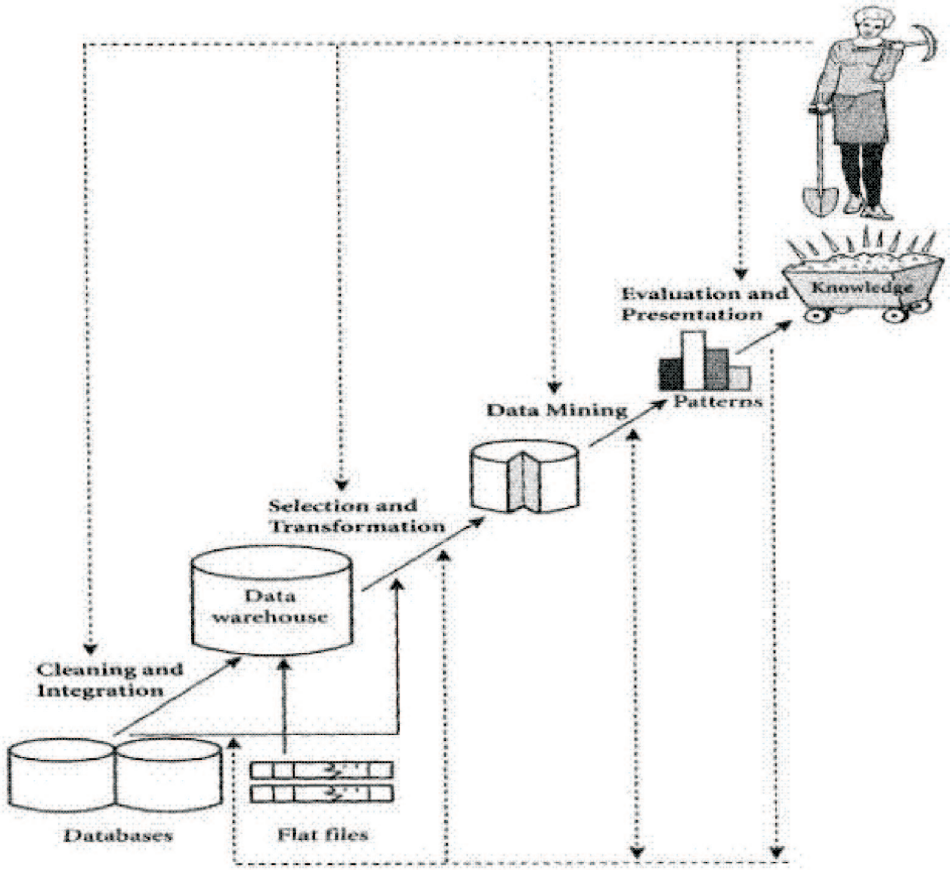
Slika 3.1: Rudarjenje podatkov, vir: Han in Kamber, [8]

znanja v Detroitu, ZDA predlagan izraz *odkrivanje znanja v zbirkah podatkov* oziroma odkrivanje znanja v podatkih (Kožuh, [15]).

Odkrivanje znanja v podatkih je proces, ki vsebuje 7 faz (Han in Kamber, [8]):

1. Čiščenje podatkov
2. Integracija podatkov (oblikovanje podatkovnega skladišča)
3. Selekcija podatkov
4. Transformacija podatkov
5. Podatkovno rudarjenje
6. Vrednotenje podatkov
7. Predstavitev podatkov

Prve štiri faze (1., 2., 3. in 4. faza) bi lahko poimenovali *priprava podatkov* oziroma *predobdelava pridobljenih podatkov*, kateri sledijo metode rudarjenja in iskanja zakonitosti v



Slika 3.2: Podatkovno rudajenje kot korak v procesu odkrivanja znanja, vir: Han in Kamber, [8]

podatkih (5. faza). Po uporabi tehnik podatkovnega rudarjenja sledi še *poobdelava podatkov* (6. in 7. faza). Rezultat odkrivanja znanja v podatkih je neko novo znanje oziroma informacija, ki smo jo pridobili iz pridobljenih podatkov (Podpečan, [16]). S pomočjo podatkovnega rudarjenja smo tako izbrali najboljše in jih primerno predstavili. Sedaj je potrebno pridobljeno znanje le še uporabiti.

Namen procesa KDD je izkoriščanje podatkov kot vir znanja za boljše odločanje in delovanje (Bohanec, [1]). S KDD želimo v podatkih odkriti zakonitosti, pravila oziroma vzorce, s katerimi bi analitiku omogočili napovedati pridhodnje primere. Omogoča nam, da na podlagi preteklih podatkov napovemo, kaj bo v prihodnje (npr. analiza nakupovalne košarice).

Podatkovno rudarjenje (angl. Data Mining, DM) je iskanje zakonitosti, vzorcev ali gibanj v veliki količini podatkov, zbirki ali skladišču podatkov, z uporabo namenskih orodij (Slovar infromatike, [21]). Nanaša se na izkopavanje oziroma rudarjenje znanja iz velike količine podatkov (Han in Kamber, [8]). Je ključen, najpomembnejši korak v procesu odkrivanja zakonitosti v podatkih. V tej fazi KDD dejansko pride do odkrivanja znanja.

Slika 3.3: Faze KDD, vir: Bohanec, [1]

Je tehnika pregledovanja podatkov z namenom odkrivanja pomembnih vzorcev v podatkih. Tehnika je zelo uporabljana v podjetjih, ki segmentirajo stranke in predvidevajo njihovo vedenje.

Podatkovno rudarjenje predstavlja izvajanje metod za odkrivanje zakonitosti v podatkih, medtem ko je KDD celoten proces analize podatkov, vseh sedem faz. DM torej predstavlja le del večjega procesa, imenovanega KDD.

Za podatkovno rudarjenje je značilno, da uporablja številne različne metode (Bohanec, [1]). Najpomembnejše metode so (Han in Kamber, [8]):

1. statistične metode (osnovne, korelacijske, diskriminantne, regresijske analize)
2. metode vizualizacije podatkov
3. metode strojnega učenja (odločitvena drevesa)
4. metode gradnje asociacijskih (povezovalnih) pravil
5. razvrščanje v skupine

Kot se različni strokovnjaki ne strinjajo z definicijo podatkovnega rudarjenja, se razhajanja najdejo tudi med tehnikami. Vendar menim, da bi se večina strinjala, da so vsi seznami

nepopolni, saj se nove tehnike odkrivajo in nastajajo dnevno, prav tako pa imajo različne tehnike mnogo različic.

V primerjavi s statistiko so rezultati podatkovnega rudarjenja natančnejši, saj zajamejo oziroma se prilegajo (vsem) podatkom, medtem ko statistične metode ne obravnavajo podatkov izven pričakovanega vzorca.

3.1 Usmerjeno in neusmerjeno podatkovno rudarjenje

Podatkovno rudarjenje bi lahko razdelili na *usmerjeno* in *neusmerjeno* podatkovno rudarjenje. Usmerjen pristop poteka od vrha navzdol in točno vemo, kaj iščemo. Ciljna spremenljivka je vnaprej določena. Pri neusmerjenem podatkovnem rudarjenju pa cilj ni v naprej določen. V primeru odkritja zakonitosti, vzorcev v podatkih, mi odločimo o njihovi pomembnosti oziroma nepomembnosti. Med usmerjeno podatkovno rudarjenje spada klasifikacija, ocenjevanje in napredovanje. Aktivnosti neusmerjenega pa so opisovanje in vizualizacija, asociacije ter razvrščanje.

Klasifikacija je postopek, s katerim enote, ki jih opazujemo, analiziramo ter razporedimo v vnaprej oblikovane razrede. Postopek, s katerim na podlagi vhodnih podatkov ocenimo vrednost določene spremenljivke, ki predstavlja posamezno lastnost opazovane enote, se imenuje *ocenjevanje*. *Napredovanje* je aktivnost, pri kateri se opazovane enote klasificirajno na podlagi vrednosti spremenljivk v prihodnosti.

Opisovanje in vizualizacija sta postopka, pri katerih gre za pojasnjevanje zakonitosti, ki so značilne za podatke v podatkovnih bazah. Vizualizacija predstavlja le grafično predstavitev opisov. Pojem *asociacije* nam pove, da gre za ugotavljanje zakonitosti, na podlagi katerih določene stvari spadajo skupaj. Med aktivnosti neusmerjenega podatkovnega rudajenja pa spada še *razvrščanje*, kar razumemo kot razvrščanje opazovanih metod v skupine glede na podobnost.

3.2 Gručenje podatkov v gruče

Ena najpomembnejših metod podatkovnega rudarjenja je razvršanje v skupine.

Razvrščanje v skupine ali *gručenje* podatkov (angl. clustering) je metoda razvrščanja podatkovnih objektov v gruče na podlagi podobnih, običajno lastnosti. Metoda objekte razvrsti po podobnosti, tako da so objekti znotraj posamezne gruče čim bolj pododobni

tistim znotraj te gruče in so čim bolj različni od objektov znotraj drugih gruč. Vsak objekt je lahko le v eni gruči.

Podrobneje si poglejmo, kaj pomeni razvrščanje objektov v gruče. Na začetku imamo množico objektov $P = \{x_1, x_2, ..., x_n\}$, kjer je n število vseh objektov. V primeru, da želimo na koncu dobiti K gruč, v katerih se bodo objekti nahajali, pri čemer je K celo število, iščemo preslikavo

$$f : P \to \{1, 2, ..., K\}\,, \tag{3.1}$$

kjer vsak objekt x_i lahko razvstimo v natanko eno gručo C_j, kjer velja $1 \leq j \leq K$. Rezultat razvrščanja v gruče je množica vseh gruč $C = \{C_1, C_2, ..., C_K\}$.

Razvrščanje v gruče poteka po določenem kriteriju, ki ga imenujemo *mera podobnosti* (angl. similarity measure). Obratna količina od podobnosti je *različnost* ali celo *razdalja*, ki se v literaturi pogosteje uporablja.

Razvrščanje poteka po naslednjih korakih (Košmelj in Breskvar Žaucer, [14]):

1. Izbiranje enot ter njihovih lastnosti: na začetku izberemo objekte, ki jih bomo opazovali oziroma merili. Te objekte lahko imenujemo tudi *vzorci*.

2. Standardizacija spremenljivk: iz vseh značilnosti izberemo tiste, ki so pomembne. Te značilnosti opišemo s spremenljivkami (atributi). Število spremenljivk, ki jih bomo opazovali, pa nam določi razsežnost prostora, v katerem se objekti nahajajo.

3. Izbira ustrezne razdalje med enotami: izberemo ustrezno mero podobnosti glede na objekte (številski ali nominalni tip) in pričakovano strukturo podatkov.

4. Izvajanje metod razvrščanja: izberemo ustrezen algoritem. Ni algoritma, ki bi bil najbolj ustrezen oziroma najboljši v vseh primerih, zato izberemo takšnega, ki bo prilagojen našemu problemu. Prav tako lahko uporabimo več postopkov in nato izberemo najboljšega.

5. Analiza rezultatov: ocenimo dobljene rešitve, saj poznamo zunanje in notranje kriterije za oceno kvalitete razvrščanja.

Metode razvrščana v skupine razdelimo na *hierarhične* (angl. hierarchical) in *nehierarhične* metode (Bohanec, [1]) ali *delitvene* (angl. partional). K razdelitvi, ki jo je podal Bohanec, pa bi lahko dodali še *geometrijske metode* ter vse ostale, ki jih ne moremo razvrstiti v eno izmed teh treh skupin, razporedimo v skupino preostalih metod (Ferligoj, [5]).

3.2.1 Nehierarhične metode

Nehierarhične metode lahko imenujemo tudi *optimizacijske metode* (Košmelj in Breskvar Žaucer, [14]), saj njihov rezultat predstavlja lokalno optimalne razvrstitve (Gasar in Bohanec, [6]). Njihova posebnost je, da mora uporabnik v naprej podati število skupin iskane razvrstitve. Pri optimizacijski metodi se na vsakem koraku izračuna določena kriterijska funkcija. Z drugačno razvrstitvijo elementov se poskuša doseči, da bi bila vrednost kriterijske funkcije čim nižja (Košmelj in Breskvar Žaucer, [14]). Torej nehierarhične metode razvrščajo elemente oziroma enote tako, da z izbrano optimizacijsko kriterijsko funkcijo izboljšujejo vnaprej podano začetno razvrstitev. Podanih je več kriterijskih funkcij, med najbolj znane pa spada *kriterij minimizacije vsote kvadratov razdalj posamezne enote do težišča v posamezni skupini* (Ferligoj, [5]).

Imenujemo pa jih lahko tudi *delitvene metode*, saj ne gradijo drevesa množic kot hierarhične, marveč eno samo množico gruč. Množico iterativno razdelijo na K gruč, pri čemer je K želeno število gruč oziroma vhodni podatek v algoritem. Cilj je optimizacija funkcije, ki odraža kvaliteto razvrščanja.

Rezultat teh metod so lokalno optimalne rešitve. Zaradi le-teh je priporočljivo, da razvščanje ponovimo z več različnimi začetnimi razvstitvami, ki so lahko dobljene z različnimi metodami. Začetno razvrstitev oziroma začetne voditelje lahko določimo:

(i) naključno,

(ii) čim bolj enakomerno razpršimo,

(iii) na osnovi analize podatkov (uporabimo rezultate katere druge metode razvrščanja).

Matrika različnosti med enotami (Ferligoj, [5]):

(i) je lahko podana,

(ii) lahko računamo sproti.

Glede na to tudi metode razvrščanja delimo v dve skupini. Najbolj znanja optimizacijska metoda razvščanja, ki temelji na že izračunanih različnostih med enotami, je *metoda prestavljanj* (angl. Relocation method). Metoda razvrščanja, ki temelji na sprotnem računanju matrike različnosti med enotami, pa je *metoda voditeljev* (znana tudi pod imenom K-means, metoda dinamičnih oblakov,...).

Metoda voditeljev

Metoda voditeljev (K-means) je najpogostejša nehierarhična oziroma *delitvena* metoda. Spada med *particijsko razvrščanje*. Računalnik izbere K točk, ki predstavljajo začetne voditelje. Vsak element, enoto doda najbližjemu voditelju, s čemer nastanejo nove skupine. Naslednji korak predstavlja izračun težišča novo nastale skupine, ki postane nov voditelj. Postopek se ponavlja, vse dokler se voditelji ne premikajo več. Postopek premešča objekte med gručami, vse dokler ne dosežemo lokalnega optimuma oziroma minimiziramo kriterijsko funkcijo.

a) *b)*

Slika 3.4: a) **Korak 1:** Točke so predstavljenje z modro, centroidi oziroma težišča so naključno določena z rdečo. b) **Korak 2:** Točke razdelimo v gruče.

a) *b)*

Slika 3.5: a) **Korak 3:** Poiščemo težišča nastalih skupin. b) **Korak 4:** Razdelimo točke v gruče z novimi težišči.

Vhodna podatka metode K-voditeljev sta število voditeljev K in število objektov n, ki

jih bomo razvrstili v K gruč. Na nekem koraku je potrebno izračunati nove voditelje po naslednji enačbi:

$$m_k = \frac{1}{N_{C_k}} \sum_{i=1}^{N_{C_k}} x_{ki}, \tag{3.2}$$

pri čemer je N_{C_k} število objektov v gruči C_k, x_{ki} pa i-ti objekt v gruči C_k.

Algoritem v grobem zapišemo po naslednjih korakih:
Korak 1: Določimo začetne voditelje (težišča) m_k, kjer je $k = 1, ..., K$.
Korak 2: Izračunamo razdalje med objekti in voditelji. Razvrstitev določimo tako, da vsako enoto priredimo njej najbližjemu voditelju.
Korak 3: Za vsako gručo C_k izračunamo njeno težišče m_k po zgornji enačbi.
Korak 4: Preverimo, ali so težišča spremenila svoj položaj glede na prejšnjo ponovitev. V primeru spremembe se vrnemo na korak 2, sicer postopek končamo. Postopek končamo tudi v primeru prekoračitve števila ponovitev.

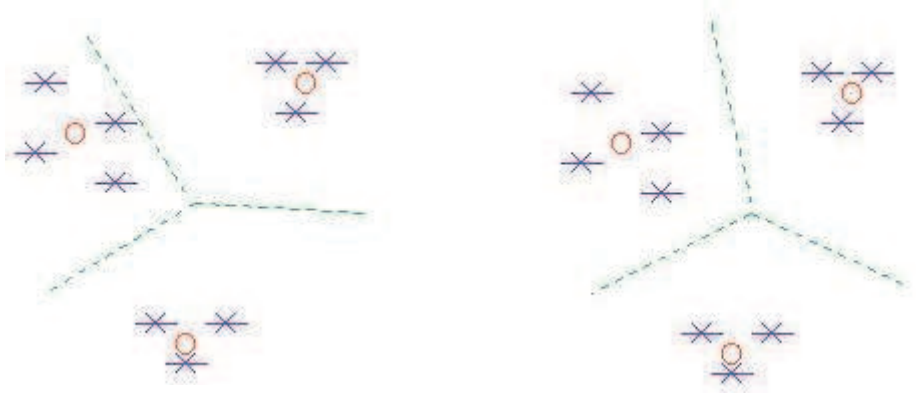
a) *b*)

Slika 3.6: a) **Korak 5:** Ponovno poiščemo težišča gruč. b) **Korak 6:** Točke razvrstimo v gruče z novimi težišči.

V vsakem primeru se zgornji postopek enkrat konča, saj imamo končno množico enot in s tem tudi končno število korakov. Postopek si podrobneje poglejmo na slikah 3.4, 3.5, 3.6 in 3.7.

3.2.2 Geometrijske metode

Razvrščanje v skupine je lahko *enodimenzionalno* razvrščanje, kjer objekte razvrščamo v gruče na osnovi ene same spremenljivke, lahko pa je tudi *večdimenzionalno*, kjer na objektih



a) *b*)

Slika 3.7: a) **Korak 7:** Ponovimo poiščemo težišča. b) **Korak 8:** Razvrstimo objekte v gruče. Razvrstitev v gručah je stabilna, zato je postopek končan.

merimo dve, tri ali več spremenljivk. V primeru dveh ali treh jih lahko predstavimo v dvo- ali trirazsežnem prostoru. Običajno pa na obravnavanih objektih merimo več spremenljivk, kar težko predstavimo v prostoru. V tem primeru nastopijo *geometrijske metode* (včasih tudi *ordinalne metode*), ki nam omogočajo preslikavo podatkov iz originalnega večrazsežnega prostora v manj razsežni, običajno dvorasežni prostor. Najznamenitejši geometrijski metodi sta *metoda glavnih komponent* in metoda *večrazsežnostno lestvičenje* (Sheppard 1962; Kruskal 1964; Ferligoj, [5]).

3.2.3 Hierarhične metode

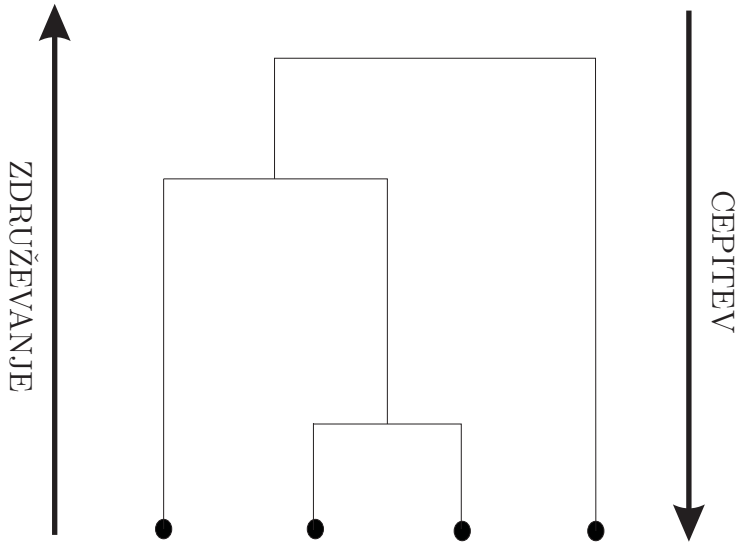
Hierarhične metode gručenja je prvi definiral S.C. Johnson v Shemah hierahičnega gručenja (angl. Hierarhichal Clustering Schemes) v Psychometrika 1967. So največkrat uporabljane metode in temeljijo na zaporednem združevanju dveh ali več gruč v novo gručo ali na razdruževanju gruč. Zgradijo celotno strukturo množic gruč; na najvišjem nivoju so vsi objekti v eni veliki gruči, na najnižjem nivoju pa je vsak objekt svoja gruča. Na vsakem nivoju vsak objekt po definiciji pripada natanko eni gruči. Rezultat nazorno prikažemo z drevesom združevanja, imenovanim *dendrogram*.

Postopek hierarhičnih metod poteka po naslednjih korakih (Košmelj in Breskvar Žaucer, [14]):

Korak 1: Vsak objekt predstavlja svojo gručo. Dobimo razvrstitev, v kateri je vsak objekt svoja gruča.

Korak 2: Izračunamo *matriko razdalj*.

Korak 3: Na vsakem koraku poiščemo med seboj najbližji gruči in ju združimo v novo.



Slika 3.8: Dendrogram

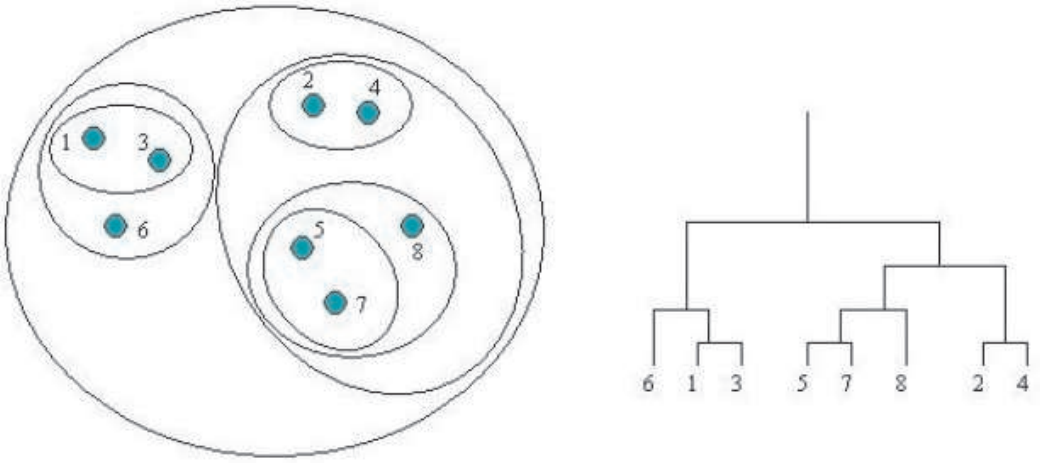
Korak 4: Staro matriko razdalj nadomesti nova matrika razdalj.
Tretji in četrti korak ponavljamo, dokler niso vsi objekti združeni v eno gručo. Številne metode hierarhičnega gručenja se razlikujejo v koraku (iii).

Algoritmi delujejo ali od spodaj navzgor (angl. bottom-up) ali od zgoraj navzdol (angl. top-down). Tako lahko tudi hierarhične metode razvrstimo na *aglomerativne* oziroma *metode združevanja* (angl. agglomerative) ter na *metode cepitve* (angl. divise). Pri združevalnih metodah, ki jih lahko imenujemo tudi *HAC* (angl. hierarchical agglomerative clustering), postopoma združujemo objekte v vedno večje gruče od spodaj navzgor - na vsakem nivoju združimo določene gruče skupaj, tako da na koncu dobimo le eno veliko gručo, v kateri so vsi objekti (slika 3.8). Pri cepitveni metodi pa je ravno nasprotno, saj le-ti začnejo na vrhu z gručo vseh objektov, in na vsakem nivoju določeno gručo razdelijo v dve, tako da dobimo vedno manjše gruče. Postopek je končan, kadar je vsak objekt v svoji gruči.

So zelo priljubljene, saj nam ni potrebno vnaprej določiti števila gruč oziroma skupin, kot je to potrebno pri nehierarhični metodi, rezultat pa lahko lepo nazorno grafično predstavimo z *dendrogramom* oziroma z *drevesom združevanja*. Hierarhično združevanje pa lahko grafično predstavimo tudi z *gnezdnim gručnim diagramom* (angl. nested cluster diagram) (slika 3.9).

Združevalne hierarhične metode so veliko bolj pogoste kot cepitvene, zato se bomo v tem diplomskem delu osredotočili na njih.

Algoritem bi lahko v grobem opisali kot (Ferligoj, [5]):
Korak 1: Vsaka enota predstavlja svojo gručo: $C_i = \{X_i\}$, $X_i \in E$, $i = 1, 2, ..., n$.
Korak 2: Ponavljaj, dokler ne ostane le ena gruča:



a) *b*)

Slika 3.9: Hierahično združevanje štirih objektov lahko predstavimo kot dendrogram (a)) in z ugnezdnim gručnim diagramom (b)), vir: Hou, [11]

Korak 2.1: določi najbližji gručo C_j in C_k,

Korak 2.2: združi najbližji gruči C_j in C_k v novo skupino $C_l = C_j \cup C_k$:

Korak 2.2.1: zamenjaj gruči C_j in C_k s C_l;

Korak 2.2.2: določi mere različnosti d med novo gručo C_l in preostalimi.

Poglavje 4

Združevalne hierarhične metode razvrščanja v gruče HAC

Rezultati *združevalnih hierarhičnih metod* oziroma krajše HAC (hierarchical agglomerative clustering) so tipično predstavljeni s hierarhijo gruč, ki jo običajno prikažemo v drevesu imenovanem *dendogram*. Je grafični prikaz združevanja s pomočjo drevesa. Omogoča nam vpogled v hierarhično stukturo in pomaga pri določanju smiselnega števila gruč. Smiselno število skupin dobimo tako, da dendrogram oziroma drevo prerežemo in s tem dobimo oziroma odčitamo dobljene skupine (Košmelj in Breskvar Žaucer, [14]).

Vsako združevanje v grafu predstavimo z vodoravno črto. Koordinata y vodoravne črte predstavlja podobnost med dvema gručama, ki sta bili združeni in sta sedaj videni kot ena gruča. S premikanjem od dna oziroma listov drevesa proti vrhnjemu vozlišču oziroma korenu, nam dendrogram omogoča rekonstrukcijo zgodovine združevanj posameznega primera [9]. Na sliki 3.9b vidimo, da sta bili nazadnje združeni gruči $\{6, 1, 3\}$ in $\{5, 7, 8, 2, 4\}$, pred tem pa $\{5, 7, 8\}$ ter $\{2, 4\}$.

Aglomerativna metoda oziroma metoda združevanja v gruče poteka po istem postopku, kot razvrščanje v gruče v prejšnjem poglavju.

Najpomembnejši del je določanje mer različnosti d med novo gručo in ostalimi. Določamo jo lahko na več načinov in ti načini določajo različne metode hierarhičnega združevanja v skupine. Poznamo več hierarhičnih metod združevanja (*minimalna*, *maksimalna* metoda, *povprečna*, *metoda težišč*, *Wardova metoda*,...), izhajajoč iz mer podobnosti, ki si jih bomo podrobneje pogledali.

4.1 Mere podobnosti oziroma različnosti

Razvrščanje v gruče združuje objekte, ki so si med seboj podobni. Vendar je podobnost zaradi svoje ohlapnosti potrebno natančneje definirati. Običajno je lažje iskati mero različnosti, kjer za vsak par (x, y), kjer je $x, y \in P$, veljajo naslednje lastnosti:

(i) $d(x, y) \geq 0$ (nenegativnost),

(ii) $d(x, y) = 0$,

(iii) $d(x, y) = d(y, x)$ (simetričnost).

Mero različnosti lahko imenujemo tudi *razdalja*, če zadošča še naslednjima dvema pogojema:

(i) $d(x, y) = 0 \Rightarrow x = y$ (razločljivost) in

(ii) za $\forall z \in P : d(x, y) + d(y, z) \geq d(x, z)$ (trikotniška neenakost).

V primeru, da zgornje lastnosti ne veljajo, bi lahko držale nemogoče trditve. V primeru, da bi ne veljala simetričnost, bi veljalo, da je 'Miha podoben Nejcu, vendar Nejc ni Mihu'. Pri razločljivosti bi to lahko pomenilo, da lahko obstajajo objekti, ki so različni, vendar jih ne moremo ločiti. Namesto pogoja trikotniške neenakosti bi držalo, da je 'Blaž podoben Nejcu in Blaž je podoben Mihu, vendar Nejc ni podoben Mihi'. Namesto lastnosti $d(x, y) = 0$ pa bi lahko držalo tudi, da je 'Miha bolj podoben Nejcu kot je Nejc podoben sebi'.

Mera različnosti nam pomaga pri izračunu razdalje med dvema točkama oziroma dvema objektoma. S tem podatkom pa lahko ocenimo mero podobnosti. Vidimo, da manjša razdalja pomeni večjo podobnost in obratno.

Če so enote določene s številskimi spremenljivkami, najpogosteje uporabljamo *evklidsko razdaljo*. Če $X = (x_1, x_2, ..., x_n)$ in $Y = (y_1, y_2, ..., y_n)$ predstavljata enoti, potem je evklidska razdalja definirana kot

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{4.1}$$

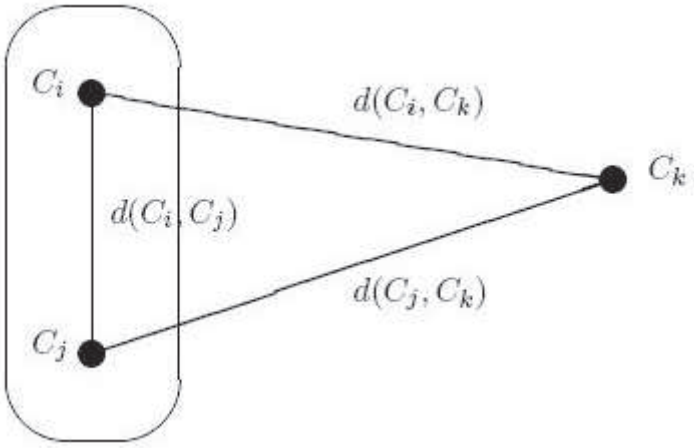
Le-ta predstavlja geometrično razdaljo med točkami v večdimenzionalnem prostoru. Večkrat uporabljana je tudi *razdalja Manhattan* (angl. city-block), ki jo izračunamo po enačbi

$$d(X, Y) = \sum_{i=1}^{n} |x_i - y_i|. \tag{4.2}$$

Obe razdalji, evklidska in Manhattan, predstavljata posebna primera *razdalje Minkowskega*, ki je definirana kot

$$d(X,Y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}, p > 0. \tag{4.3}$$

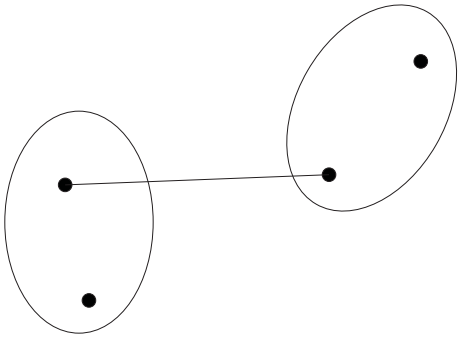
Če je $p = 1$, gre za razdaljo Manhattan, v primeru $p = 2$ pa za evklidsko razdaljo. Prav tako obstajajo še nekatere druge razdalje, kot sta *Čebiševa ali trdnjavska razdalja* in *potenčna razdalja*.



Slika 4.1: Izračun mer podobnosti med tremi skupinami, vir: Ferligoj, [5]

Potrebno je omeniti, da pri binarnih oziroma nominalnih objektih ujemanje med parom vzorcev merimo s pomočjo frekvenc v kontingenčni tabeli. Izbira posamezne mere podobnosti d je odvisna od naše izbire in seveda od podatkov. Pomembno je, da pred samim razvrščanjem dobro izberemo ustrezno mero, saj bodo rezultati v veliki meri odvisni od nje. V tej diplomski nalogi pa bomo uporabljali evklidsko razdaljo.

Na sliki 4.1 imamo tri objekte oziroma gruče C_i, C_j in C_k, ki jih bomo na naslednjem koraku združili z eno izmed metod združevanja. Ker sta gruči C_i in C_j najbližji, oziroma je njuna razdalja najmanjša, ju združimo v novo skupino $C_i \cup C_j$. Nato je potrebno v skladu z zgornjim postopkom ponovno določiti mero različnosti med novo nastalo skupino $C_i \cup C_j$ in skupino C_k (Ferligoj, [5]). To novo mero različnosti lahko določimo po eni izmed naslednjih metod:



Slika 4.2: Minimalna metoda

4.2 Minimalna metoda

Minimalna metoda (angl. Single Link ali Single Linkage Clustering) ali tudi *enojna povezanost* temelji na izračunu najmanjše razdalje med dvema enotama v *matriki razdalj* oziroma združuje najbolj podobne gruče.

Matrika razdalj na sliki 4.3 je simetrična, v njej pa so predstavljene razdalje med objekti npr. a, b, c in d. Ta metoda je dobra pri rokovanju z neeliptičnimi oblikami, vendar je

	a	b	c	d
a	0	f	g	h
b	f	0	i	j
c	g	i	0	k
d	h	j	k	0

Slika 4.3: Matrika razdalj

občutljiva na šum in zunanje vplive [24].

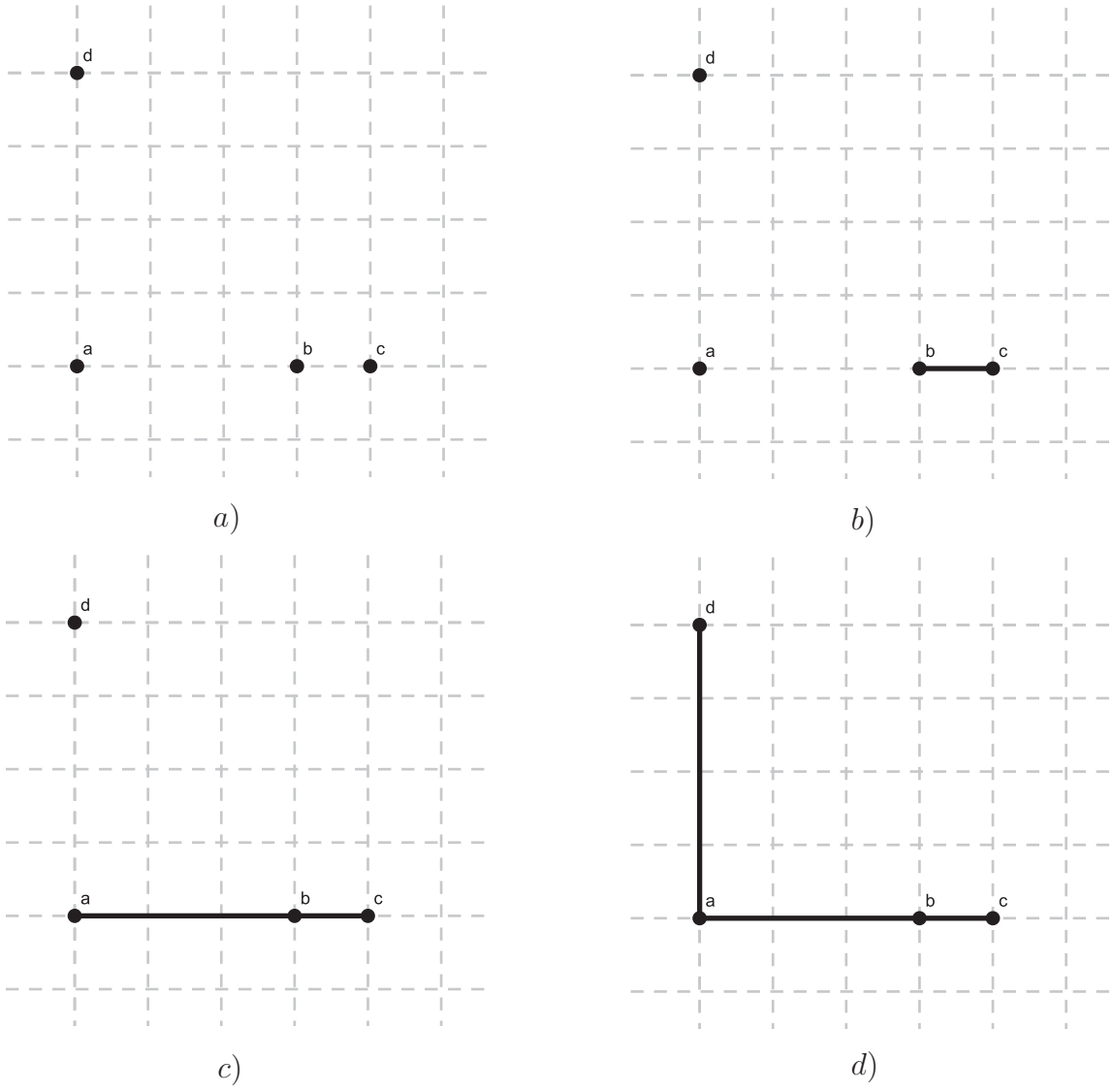
Razdalja med že nastalo skupino ter novim elementom se izračuna po *pravilu najbližjega soseda* (angl. Nearest Neighbor), ki predstavlja minimalno razdaljo.

$$d_{skupina,objekt} = \text{minimalna razdalja članov skupine do objekta}$$

S to metodo izračunamo razdaljo med najbližjima sosedoma izmed dveh različnih gruč, kot kaže slika 4.17. Po sliki 4.1 bi torej naš izračun nove mere različnosti izgledal kot

$$d(C_i \cup C_j, C_k) = min(d(C_i, C_k), d(C_j, C_k)). \tag{4.4}$$

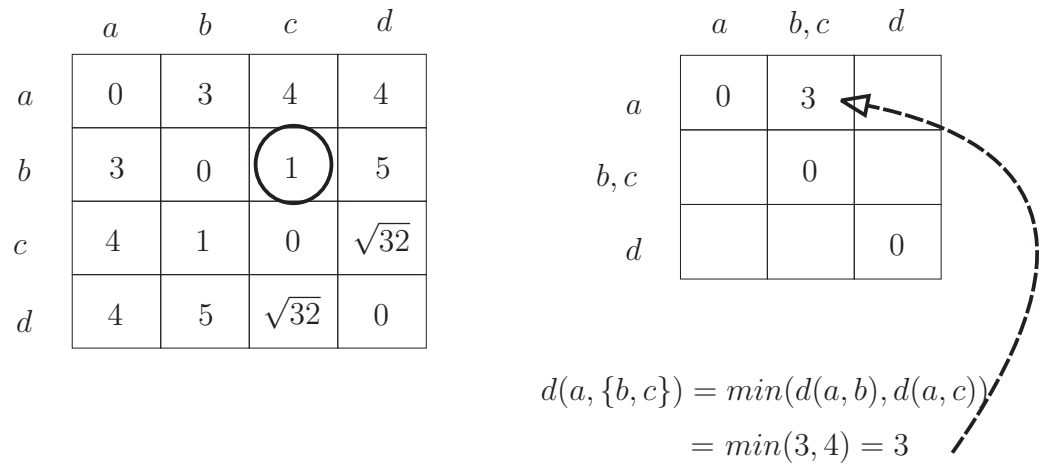
Združevalni kriterij minimalne metode je *lokalen*, saj združujemo le gruče, ki so blizu druga druge. Osredotočimo se izključno na območje, kjer sta dve gruči blizu in preostale gruče ne pridejo v poštev. Prav nasprotno pa je pri maksimalni metodi, ki si jo bomo pogledali kasneje [9].



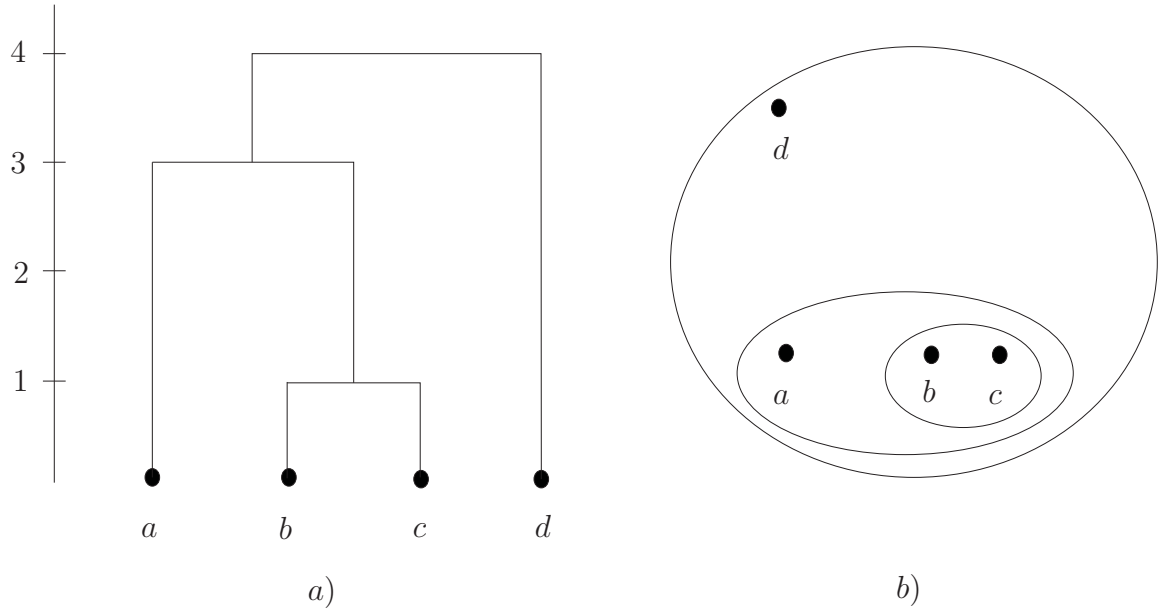
Slika 4.4: Poglejmo si združevanje z minimalno metodo na štirih objektih na sliki **a)**. Najprej pod **b)** združimo najbližja elementa, torej $\{b\}$ in $\{c\}$, nato pod **c)** h gruči $\{b, c\}$ pridružimo še $\{a\}$ in na koncu (**d)**) združimo še $\{d\}$ in $\{a, b, c\}$

.

Primer združevanja z minimalno metodo vidimo na sliki 4.4. Matrika razdalj za isti primer je predstavljena na sliki 4.5, njena grafična predstavitev pa na 4.6.



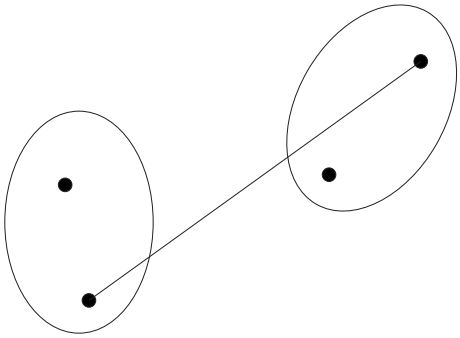
Slika 4.5: Matrika razdalj pri minimalni metodi na primeru s slike 4.4.



Slika 4.6: a) Dendrogram in b) gnezdni gručni diagram za primer pri minimalni metodi

4.3 Maksimalna metoda

Maksimalna metoda (angl. Complete Link ali Complete Linkage Clustering) ali tudi *polna povezanost* izračuna razdalje med skupinami po pravilu *najbolj oddaljenega soseda* (angl. Furthest Neighbor), kar predstavlja maksimalno razdaljo med objekti oziroma najmanjšo podobnost med dvema objektoma znotraj dveh različnih gruč. Primerna je za naravno različne skupke [10], neprimerna pa za podolgovate, verižne strukture. Slika 4.7 predstavlja



Slika 4.7: Maksimalna metoda

uporabo maksimalne metode.

$$d_{skupina,objekt} = \text{maksimalna razdalja članov skupine do objekta}$$

Po sliki 4.1 bi torej naš izračun nove mere različnosti izgledal kot

$$d(C_i \cup C_j, C_k) = max(d(C_i, C_k), d(C_j, C_k)). \tag{4.5}$$

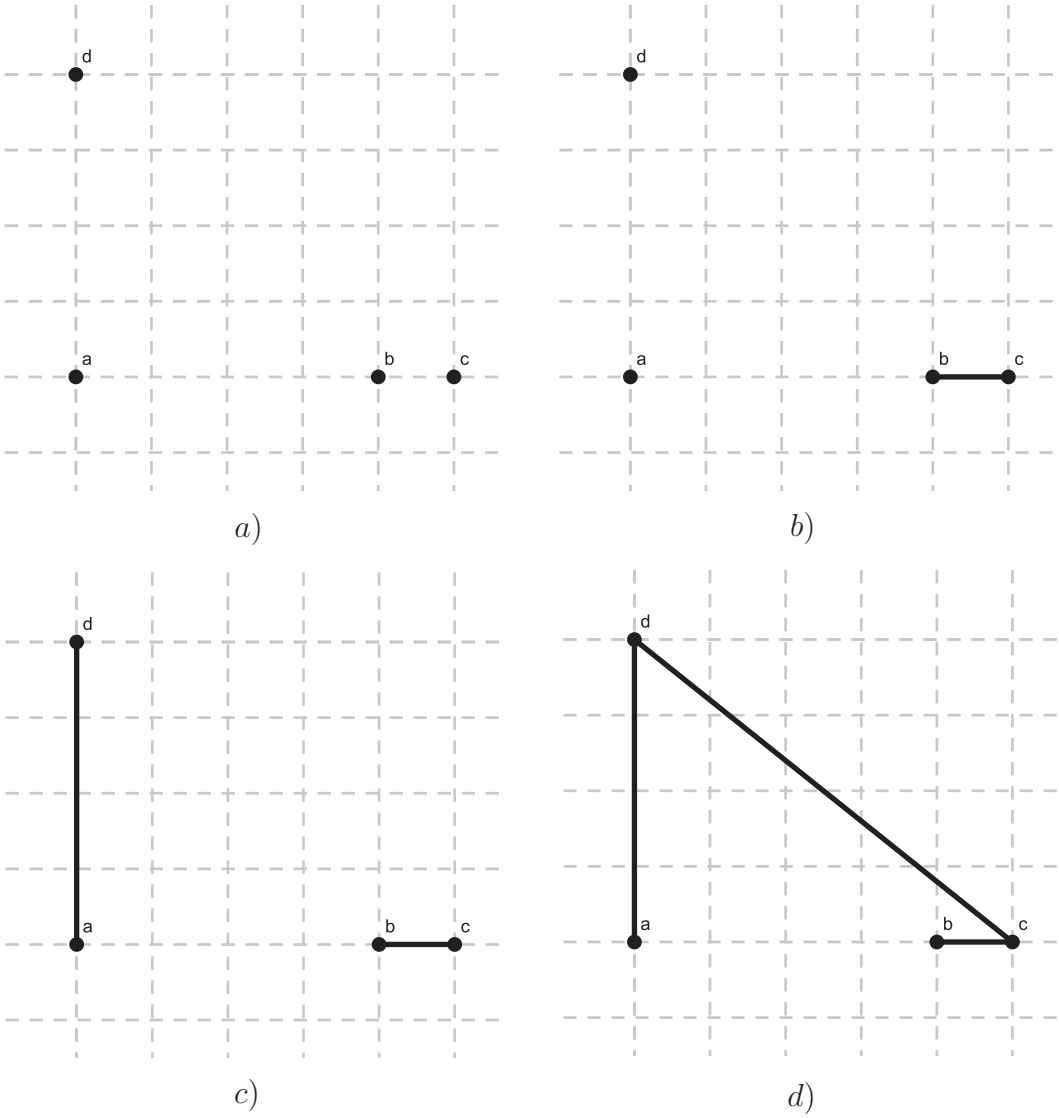
Združevanje po maksimalni metodi je nelokalno, saj celotna struktura združevanja vpliva na odločitve o tem, kaj bomo na naslednjem koraku združili.

Matriko razdalj za maksimalno metodo vidimo na sliki 4.9.

Tako minimalno kot maksimalno metodo lahko teoretično razložimo z grafi. Naj bo s_k podobnost dveh gruč, ki sta združeni na k-tem koraku in $G(s_k)$ graf, ki povezuje vse točke s podobnostjo najmanj s_k. Tako gruče po koraku k pri minimalni metodi predstavljajo največjo množico točk, kjer velja, da za vsak par obstaja pot, ki ju povezuje. Pri maksimalni metodi pa po k-tem koraku dobimo množico točk, ki so popolnoma povezane druga z drugo. Ta teoretična razlaga nam razodene pomen imena minimalne metode oziroma enojne povezanosti ter maksimalne metode oziroma popolne povezanosti. Gruče, povezane z minimalno metodo, so na koraku k največje množice objektov, ki so povezane z najmanj eno povezavo s podobnostjo $s \geq s_k$; gruče pri maksimalni metodi so na koraku k največje množice objektov, popolnoma povezane ena z drugo s povezavami, katerih podobnost je $s \geq s_k$ [9].

4.4 Povprečna metoda

Povprečna metoda (angl. Group-average agglomerative clustering) računa razdaljo glede na povprečje razdalj med vsemi elementi, tudi med elementi iste gruče. Defnirana je torej kot

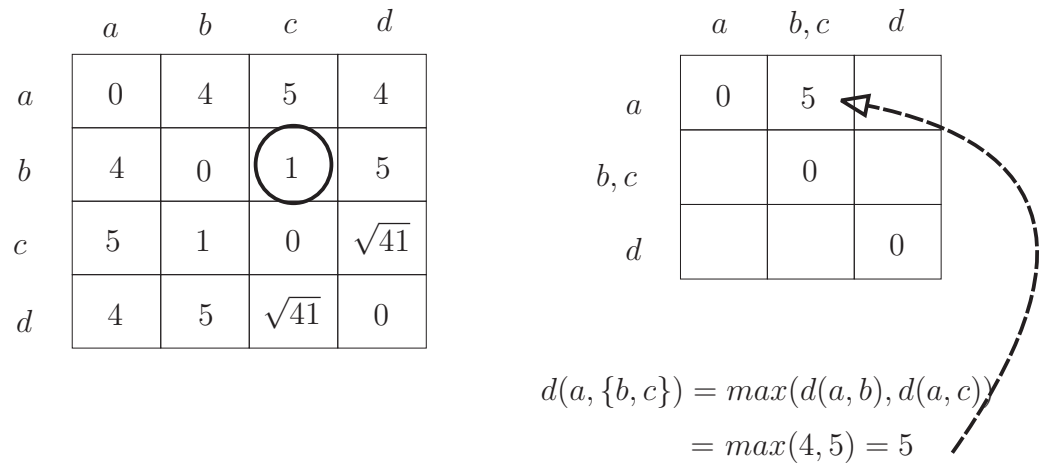


Slika 4.8: Primer združevanja po maksimalni metodi

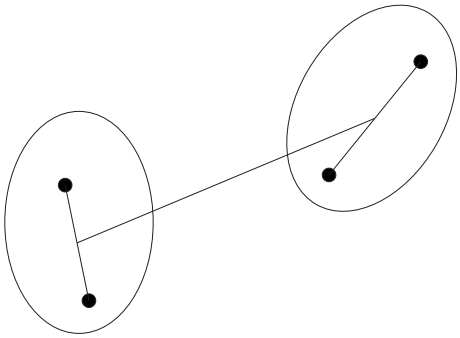
povprečje vseh parov točk v različnih gručah. Gre za vmesen pristop med minimalno in maksimalno metodo. Pri tej metodi je mera različnosti gruč C_i in C_j, katerih velikosti sta m_i in m_j, definirana kot

$$razlicnost(C_i, C_j) = \frac{\sum_{\substack{x \in C_i \\ y \in C_j}} razlicnost(x, y)}{m_i * m_j}. \tag{4.6}$$

Prav tako poznamo *netehtano skupinsko povprečjne* kot tudi *tehtano skupinsko povprečje*. Neobteženo povprečje je primerno za jasno razmejene verižne strukture, medtem ko obteženo povprečje uporabljamo, kadar sumimo, da se gruče bistveno razlikujejo v velikosti.



Slika 4.9: Matrika razdalj pri maksimalni metodi.



Slika 4.10: Povprečna metoda

$d_{skupina,objekt}$ = povprečna razdalja članov skupine do obejkta

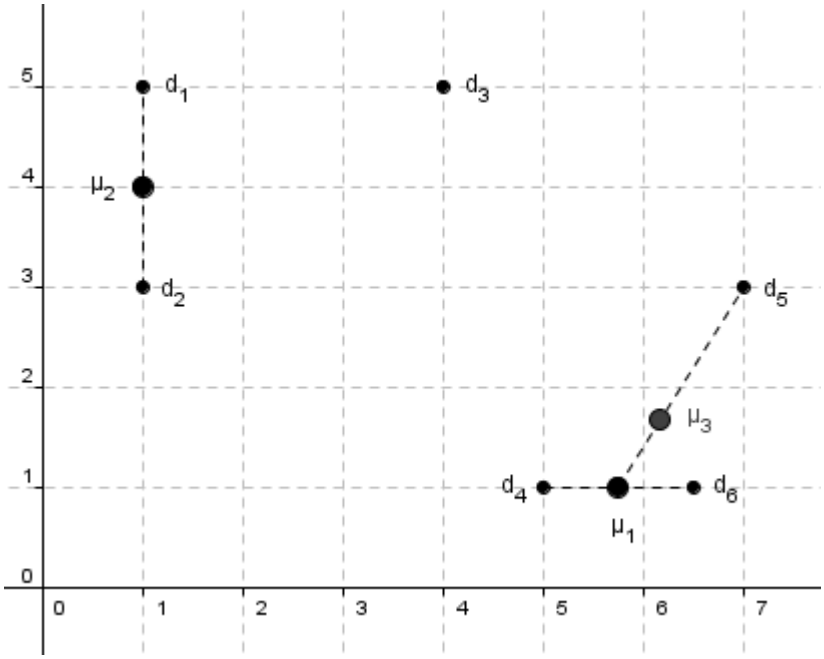
4.5 Metoda težišč

Metoda težišč (angl. Centroid method) ali *Gowerjeva metoda* (angl. Gower method) izračuna razdaljo med skupinami glede na položaj njunih težišč. Poznamo tako *tehtano* kot tudi *netehtano* težišče, kjer so pri tehtanemu gruče obtežene še z velikostjo posamezne gruče oziroma s številom objektov znotraj gruče.

Po sliki 4.1 bi tako po Gowerjevi metodi izračunali mero različnosti kot

$$d(C_i \cup C_j, C_k) = d^2(T_{ij}, T_k), \tag{4.7}$$

kjer je T_{ij} težišče $C_i \cup C_j$, T_k pa je težišče gruče C_k.



Slika 4.11: Postopek združevanja gruč po metodi težišč [9]. Opazimo tri ponovitve računanja, μ_1, μ_2 in μ_3.

Za razliko od preostalih HAC algoritmov (minimalne, povprečne in maksimalne metod) metoda težišč ni monotona. Pojavijo se tako imenovane inverizije oziroma obrati, kjer se podobnost poveča med gručenjem. Podobnost lahko definiramo kot negativno razdaljo [9].

4.6 Wardova metoda

Wardova metoda (angl. Ward's method) je poseben primer hierarhične metode združevanja, ki uporablja analizo variance. Mero različnosti med novo nastalo gručo, ki je sestavljena iz C_i in C_j, in drugo gručo C_k izračunamo kot

$$d(C_i \cup C_j, C_k) = \frac{(n_i + n_j)n_k}{(n_i + n_j + n_k)} d^2(T_{i,j}, T_k), \tag{4.8}$$

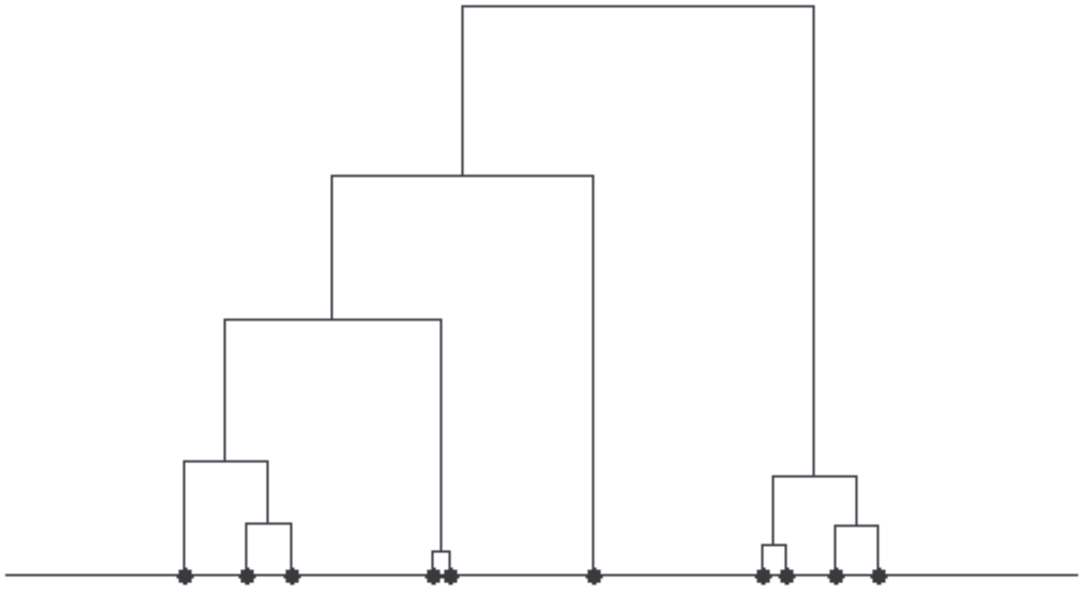
pri čemer je n_i število enot v C_i, T_i pa težišče skupine C_i.

4.7 Dendrogram

Beseda *dendrogram* izhaja iz grščine, kjer *dendron* pomeni drevo in *gramma* risanje. Dendrogram je drevesni diagram, ki se pogosto uporablja za predstavitev razporeditve gruč

v algoritmih hierarhičnega gručenja. Prav tako se pogosto uporablja v biologiji za predstavitev združevanja genov (Wikipedia, [27]).

Dendrogram je grafična predstavitev rezultatov hierarhičnega gručenja, kjer je vsak korak predstavljen kot združitev dveh vej drevesa v eno samo vejo. Veje predstavljajo gruče, pridobljene na vsakem koraku hierarhičnega gručenja.



Slika 4.12: Dendrogram

Dendrogram grafično predstavlja informacije, ki se navezujejo glede na opazovane podatke, ki so povezani skupaj na različnih nivojih podobnosti/različnosti. Na dnu dendrograma je vsak opazovan objekt svoja gruča. Navpične črte, ki izhajajo iz teh posameznih gruč, so povezane z drugimi navpičnimi črtami iz drugih gruč s pomočjo vodoravne črte. Vodoravna črta se nahaja na različnih vrednostih podobnosti/različnosti. Povezovanje gruč se nadaljuje vse dokler na vrhu dendrograma niso vse gruče združene v eno.

Višina navpičnih črt in lestvica na podobnosti/različnosti osi nam daje vizualno predstavo o jakosti gručenja. Dolge navpične črte kažejo na bolj izrazito ločljivost med gručami. Dolge navpične črte na vrhu dendrogama nakazujejo, da so gruče, predstavljene s temi črtami, dobro ločene ena od druge. Kratke črte prikazujejo gruče, ki niso tako različne (Stata, [22]).

Dendrogrami nam takoj odgovorijo na vprašanje, ali sta dve gruči bližje kot tretja in ju bomo dodali v novo, večjo gručo. Kljub temu pa je nadziranje originalnih razdalj in prav tako tudi koeficientov podobnosti pogosto pokazalo, da čeprav sta dva objekta blizu skupaj, po

algoritmu gručenja pristaneta v dveh povsem različnih gručah. Zato je gručenje enostavnejše po le eni dimenziji, ki združi podobne in oddalji že tako različne objekte. Gručenje podatkov v eni dimenziji, v dendrogramu, zelo posploši postopek razporeditve gruč in spremeni NP-težek problem v N-težek problem (Bulbeck, [3]).

4.8 Lastnosti HAC

Temeljna predpostavka združevalne metode gručenja je, da združevanje poteka *monotonično*. Monotonično združevanje pomeni, da če so $s_1, s_2, ..., s_k$ podobnosti uspešnih združevanj metode HAC, potem velja

$$s_1 \geq s_2 \geq s_3 \geq ... \geq s_{k-1}. \tag{4.9}$$

V nemonotoničnem hierarhičnem gručenju obstaja vsaj ena *inverzija* oziroma obrat, kot je $s_i < s_{i+1}$, kar spodbija temeljno predpostavko, da vedno izberemo najboljšo možno združitev, ki je na vsakem koraku na voljo [9]. Takšno nemonotonično hierarhično združevanje je metoda težišč, kar smo že omenili.

Kot smo že omenili, pri HAC ni potrebno vnaprej določiti števila gruč. Vseeno pa običajno želimo kot rezultat dobiti razdelitev na določeno število gruč. V tem primeru je potrebno dendrogram na določeni točki prerezati. Obstaja več možnih načinov, kje razrezati drevo ([9]):

(i) Razrez dendrograma v točki, kjer je prostor med združitvama največji oziroma je največja razlika v podobnosti. V takšnih primerih kot rezultat dobimo *naravno gručenje*.

(ii) Če želimo dobiti K gruč, drevo razrežemo tako, da dobimo kot rezultat K gruč.

4.9 Časovna in prostorska zahtevnost

Za osnovni združevalni hierahični algoritem gručenja potrebujemo matriko velikosti n x n. Predvidevamo, da je le-ta simetrična, torej nam zavzame $\frac{1}{2}n^2$ prostora, pri čemer je n število objektov, ki jih združujemo. Prostor, ki ga potrebujemo, da sledimo združevanju gruč, je sorazmeren s številom gruč, kar je $n-1$, če izključimo objekte, ki so sami v gruči. Torej je skupna prostorska zahtevnost enaka $O(n^2)$.

Časovna zahtevnost, ki je potrebna za izračun matrike razdalj, je $O(n^2)$. Po tem koraku sledi $n-1$ iteracij oziroma ponovitev koraka, v katerem ponovno izračunamo razdalje novo

nastalih gruč do preostalih, saj je na začetku n gruč, na vsakem koraku pa se združita le dve gruči. Nato sledi pregledovanje matrike, kjer iščemo njen najmanjši element. Z linearnim iskanjem potrebujemo časovno zahtevnost $O((n-i+1)^2)$ za i-to ponovitev. Za posodobitev matrike potrebujemo le $O(n-i+1)$ časa. Brez prilagoditev bi imeli časovno zahtevnost $O(n^3)$.

Če razdalje do različnih gruč shranimo v seznam oziroma ustvarimo prioritetno vrsto, lahko zahtevnost iskanja dveh najbližjih gruč zmanjšamo na $O(n-i+1)$. Kljub vsemu pa še vedno potrebujemo časovno zahtevnost $O(n^2 log(n))$ (Tan, Steinbach in Kumar, [24]).

Časovna in prostorska zahtevnost hierarhičnega gručenja sta zelo zahtevni in z njima zaradi same zahtevnosti ne združujemo velikih količin podatkov.

4.10 HAC algoritmi

Podrobneje si poglejmo preprost primer algoritma HAC.

PreprostHAC($d_1, d_2, ..., d_N$);
for $n = 1$ **to** N **do**
  for $i = 1$ **to** N **do**
    $C[n][i] = podobnost(d_n, d_i)$ (matriko razdalj napolnimo s podobnostmi);
    $I[n] = 1$ (shanjujemo aktivne gruče);
  end
end
$A = []$ (shranjujemo zgodovino gručenja);
for $k = 1$ **to** $N - 1$ **do**
  $\langle i, m \rangle = argmax_{\{\langle i,m \rangle : i \neq m \wedge I[i]=1 \wedge I[m]=1\}} C[i][m]$ (poiščemo najbolj podobni gruči);
  A.Dodaj($\langle i, m \rangle$) (shrani zlitje);
  for $j = 1$ **to** N **do**
    $C[i][j] = podobnost(i, m, j)$;
    $C[j][i] = podobnost(i, m, j)$ (zaradi simetričnosti matrike);
  end
  $I[m] = 0$ (izbriše gručo iz vrste aktivnih);
end
return A

Algorithm 1: Preprost HAC algoritem, vir: [9]

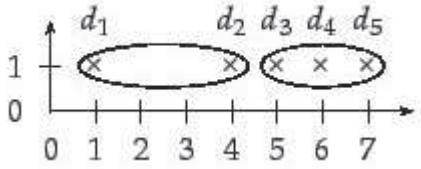
Opazimo, da v preprostem algoritmu najprej izračunamo matriko podobnosti C velikosti n x n. Nato algoritem izvede $n-1$ korak združevanja trenutno najbolj podobnih gruč. V vsaki ponovitvi sta združeni gruči, ki sta si najbolj podobni, matrika C pa se zatem

posodobi. Vsa združevanja se shranjujejo v vrsti združevanj A. V vrsti I pa se shranjujejo gruče, ki so še na voljo za združevanje. S funkcijo *podobnost*(i, m, j) izračunamo podobnost gruče j s preostalimi. Pri minimalni metodi na primer izmed $C[j][i]$ in $C[j][m]$ izberemo tisto, ki je največja [9].

4.11 Prednosti in slabosti hierarhičnih metod

Ena izmed glavnih prednosti je vsekakor grafična predstavitev rezultatov algoritmov, ki jih lahko predstavimo z drevesom, imenovanim dendrogram. Mnoge študije so tudi ugotovile, da s temi algoritmi dobimo tudi gruče boljših kvalitet.

Na drugi strani slabosti predstavljata velika časovna in prostorska zahtevnost. Z drugimi algoritmi, kot je K-voditeljev, je prostorska zahtevnost enaka $O((m+K)n)$, kjer je m število objektov in n število atributov, spremenljivk. Tudi časovna zahtevnost je pri K-voditeljev boljša: $O(I * K * m * n)$, kjer je I število ponovitev. Hierarhično združevanje je torej neprimerljivo za večje število objektov, saj je zelo drago. Prav tako lahko težave povzročajo podatki s šumi, večdimenzionalni podatki.



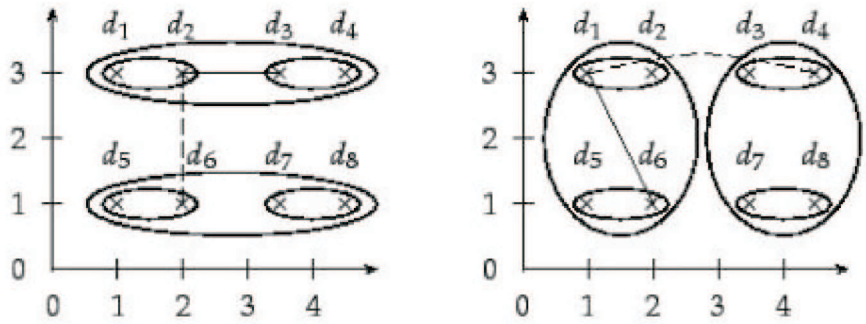
Slika 4.13: Slabost maksimalne metode HAC, vir: [9]

Slabost minimalne metode je *veriženje* (angl. chaining), kar lahko opazimo na sliki 4.16 a), saj zaradi lokalnega združevalnega kriterija nastajajo dolge verige. V nasprotju s tem pa se maksimalna metoda tej težavi izogne, s čimer je ta organizacija gruč bolj uporabna. Kljub temu pa pri tej metodi naletimo na drug problem. Metoda veliko pozornosti nameni objektom, ki se ne prilagajajo globalni strukturi gruče [9]. Takšen primer vidimo na sliki 4.13. Opazimo, da nam maksimalna metoda ustvari dve gruči $\{\{d_1, d_2\}, \{d_3, d_4, d_5\}\}$. Najbolj primerno bi seveda bilo, če bi kot rezultat dobili gruči $\{d_1\}$ in $\{d_2, d_3, d_4, d_5\}$, tako pa nam d_1 razdeli $\{d_2, d_3, d_4, d_5\}$.

4.12 Primerjava hierarhičnih algoritmov gručenja

V tem podpoglavju bomo med seboj primerjali različne metode hierarhičnega gručenja.

Že po sami ideji združevanja minimalna metoda poteka po istem algoritmu kot *Kruskalov algoritem* za iskanje minimalnega vpetega drevesa. Vendar pa je zaporedje, v katerem so gruče združene, pri minimalni metodi pomembno. Prav tako maksimalna in povprečna metoda predstavljata le različici minimalne metode, kjer uporabimo drugo enabo računanja razdalje med gručami.



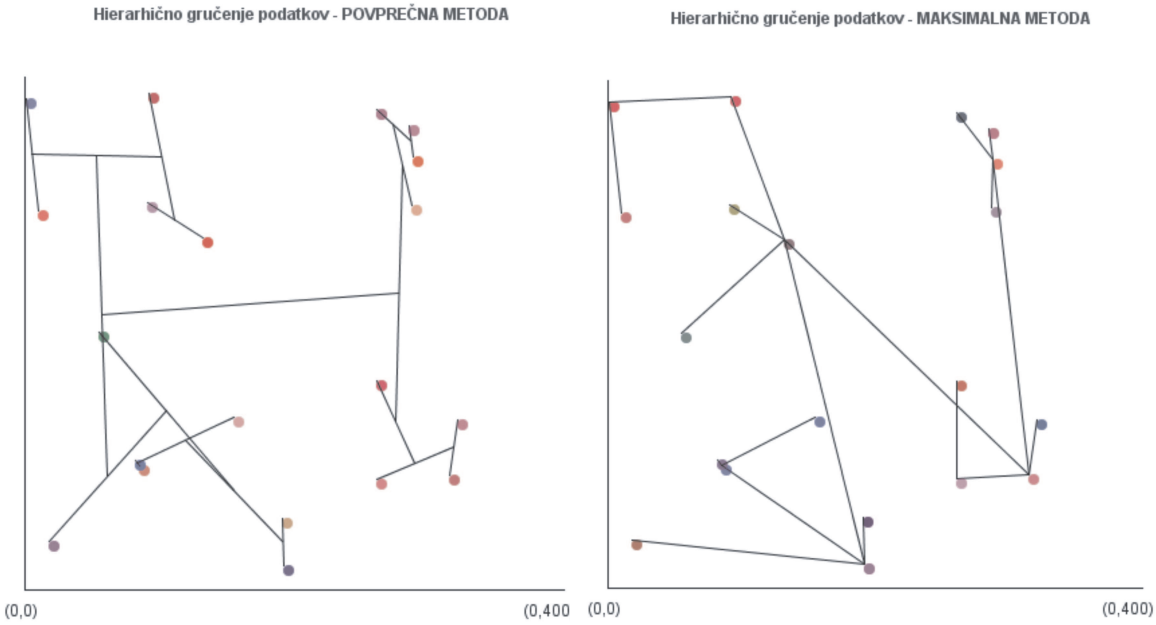
Slika 4.14: Minimalna metoda (levo) in maksimalna metoda (desno), vir: [9]

Razlike v tvorjenju gruč lahko opazimo že na sliki 4.14. Minimalna metoda (levo) združi d_2 in d_3, saj je podobnost med njima največja. Pri maksimalni metodi (desno) pa je razdalja oziroma podobnost med d_1 in d_4 manjša kot med d_1 in d_6, zato sta tokrat združena v drugačne gruče kot pri minimalni metodi. Gruče in tudi pripadajoče grafične predstavitve z dendrogramom so za različne metode različne.

Na slikah 4.15 a) in b) ter na sliki 4.16 a) lahko med seboj primerjamo različne metode gručenja na istih podatkih. Na tri različne načine smo točke uspešno razvrstili v eno veliko gručo, vendar opazimo, da so načini povezovanja povsem različni. Opazimo, da se pri minimalni metodi (slika 4.16 a)) pokaže njena slabost, saj kot rezultat dobimo dolgo verigo. Temu se pri povprečni in maksimalni metodi (sliki 4.15 a) in b)) izognemo. Glede na rezultate gručenja povprečna metoda, ki je med minimalno in maksimalno, deluje še najbolj optimalno.

Na slikah 4.16 a), b), c) in d) lahko opazujemo slabost minimalne metode - veriženja na večji množici podatkov.

Vsako gručenje lahko, kot smo že povedali, predstavimo z dendrogramom, ki se od primera do primera razlikujejo. Slika 4.17 prikazuje tri množice podatkov in združevanje le teh v gruče z minimalno metodo. Vsako gručenje z leve strani se odraža tudi v dendrogramu na pripadajoči desni strani slike, kjer lahko natančneje opazujemo, katere gruče se na posameznem koraku oziroma na posamezni razdalji združijo. Če je več gruč na podobni razdalji, so razlike v višini dendrograma (sredinska slika) minimalne ali celo ničelne.



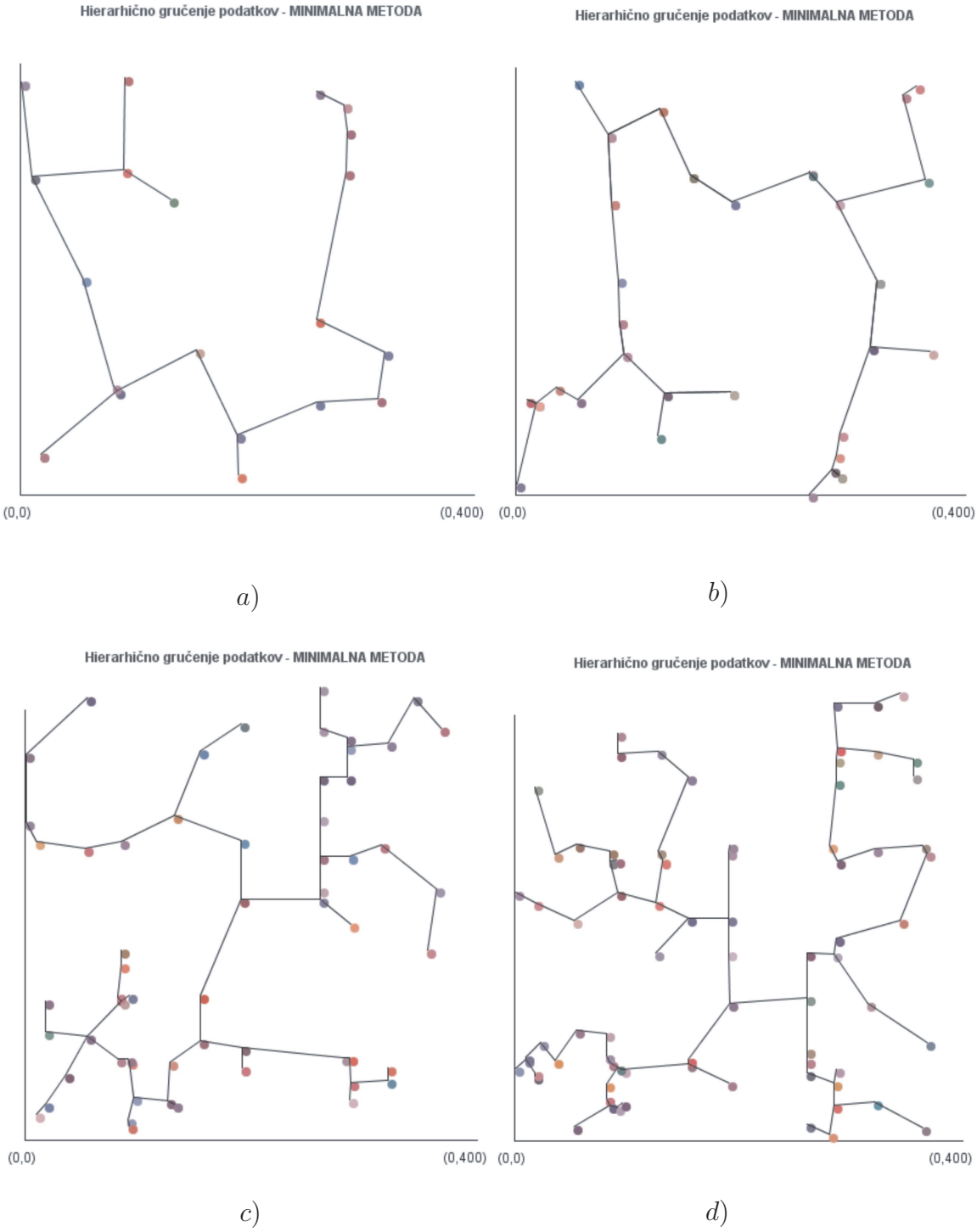
a) *b*)

Slika 4.15: Združevanje 20 točk s povprečno (a)) in maksimalno (b)) metodo.

Slika 4.18 prikazuje združevanje po maksimalni (zgoraj) in povprečni (spodaj) hierarhični metodi gručenja ter pripadajoča dendrograma. Pri maksimalni metodi opazimo, da v gruče povezuje med seboj res različne oziroma oddaljene podatke, medtem ko so si posamezni podatki bližje in podobnejši. Povprečna metoda zaenkrat še vedno deluje najbolj optimalno. Zato si bomo pogledali vse tri metode hierarhičnega gručenja in njihove dendrograme na množici istih podatkov in nato zaključili podpoglavje z najprimernejšo metodo hierarhičnega gručenja.

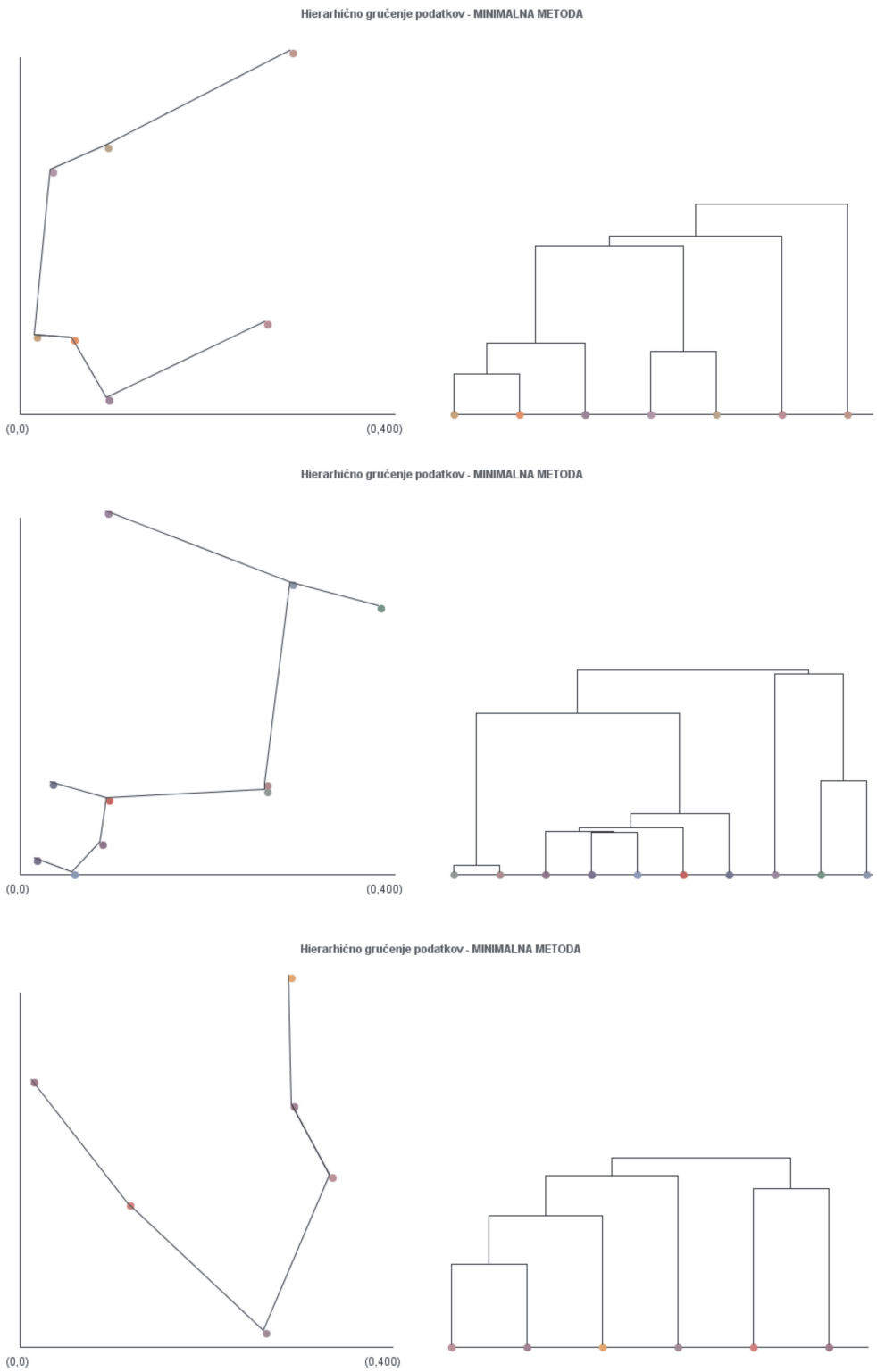
Slike 4.19, 4.20 in 4.21 prikazujejo hierarhično gručenje z minimalno, povprečno in maksimalno metodo na istih podatkih. Že na prvi pogled opazimo, da je gručenje pri povprečni metodi najboljša izbira tako po 2D gručenju na levi strani slike, kot pri dendrogramu. Najbolj očitna razlika med dendrogrami je, da je le-ta pri minimalni metodi najnižji oziroma stisnjen, saj so tudi razdalje, glede na katere gručimo podatke, najkrajše, pri maksimalni metodi pa je ravno obratno in je dendrogram navpično razpotegnjen. Medtem pa je dendrogram povprečne metode nekje vmes. Opazimo, da sta si dendrogram za povprečno in maksimalno metodo zelo podobna, čeprav do gručenj prihaja na različnih nivojih oziroma razdaljah, vendar je zaporedje gručenj pri obeh enako. Kljub temu pa sta njuni predstavitvi v koordinatnih sistemih bistveno drugačni.

V tabeli 4.1 med seboj primerjamo štiri metode in opazimo, kot smo že prej ugotovili na realnih podatkih, da najboljšo izbiro metode predstavlja povprečna metoda, čeprav vse

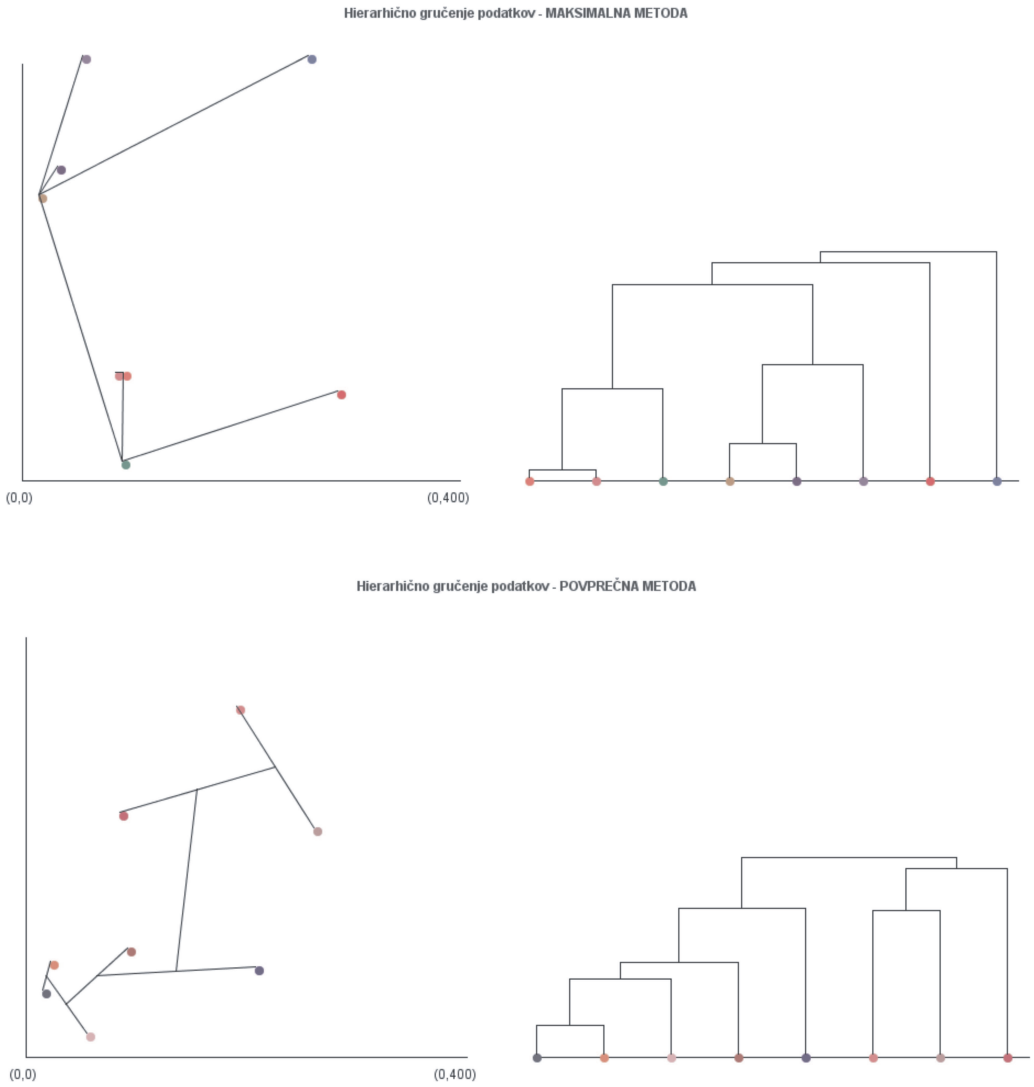


Slika 4.16: Združevanje z minimalno metodo: a) 20 točk, b) 40 točk, c) 60 točk in d) 80 točk.

metode delujejo po enakem algoritmu, le načini združevanja so različni. Kljub temu pa izbira metode odvisna tako od posameznika, ki z metodo dela, kot tudi od podatkov, ki jih



Slika 4.17: Združevanje po minimalni metodi in pripadajoči dendrogrami.

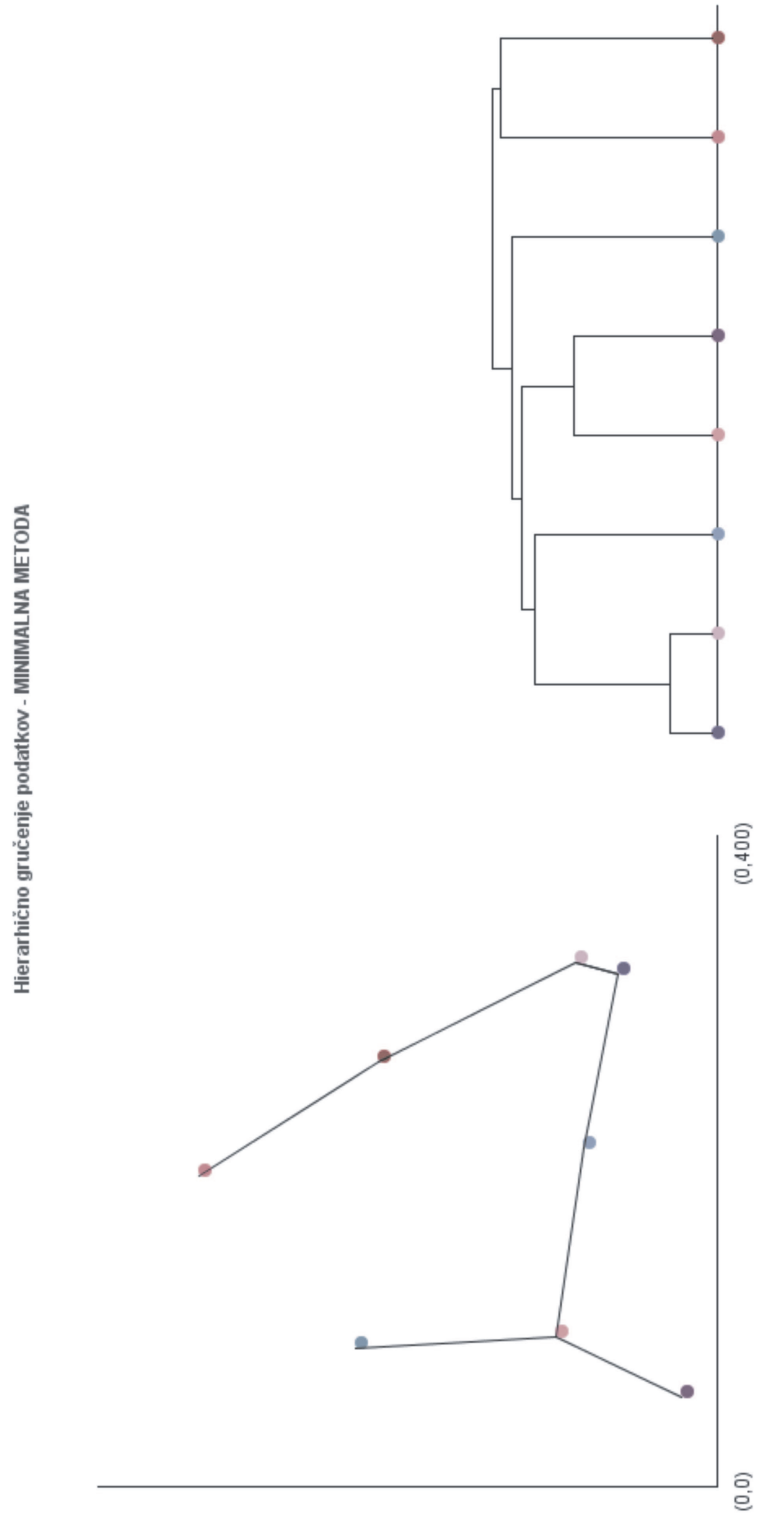


Slika 4.18: Združevanje po maksimalni (zgoraj) ter povprečni (spodaj) metodi in pripadajoča dendrograma.

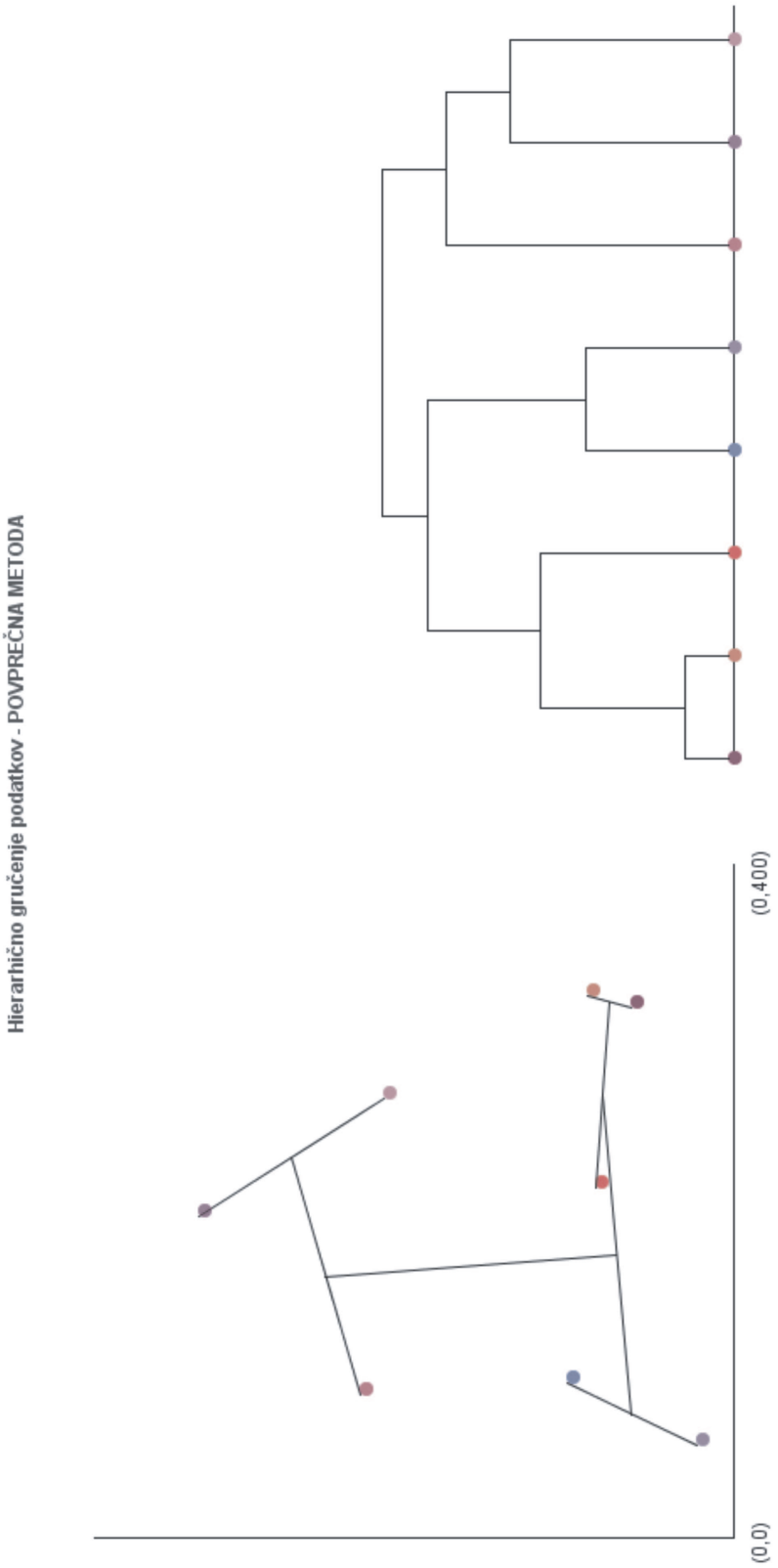
za posamezni primer imamo.

Tabela 4.1: Primerjava HAC metod, vir [9]

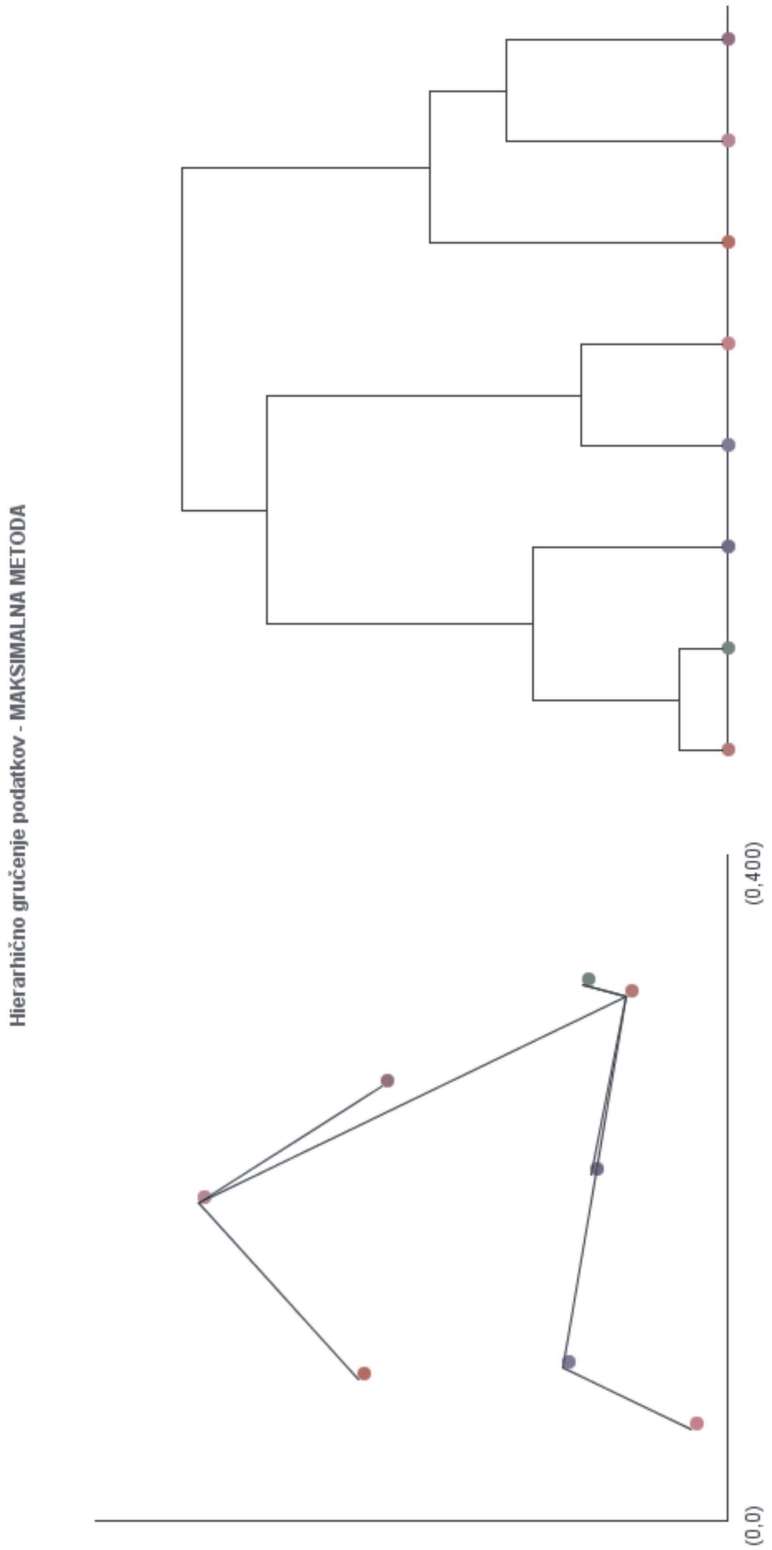
Metoda	**Podobnost**	**Opombe**
minimalna	največja	efekt veriženja
maksimalna	najmanjša	občutljivost na šume
povprečna	povprečje vseh	najboljša izbira
metoda težišč	povprečna	inverzije



Slika 4.19: Minimalna metoda hierarhičnega gručenja na podatkih za primerjavo s povprečno in maksimalno metodo.



Slika 4.20: Povprečna metoda hierarhičnega gručenja na podatkih za primerjavo z minimalno in maksimalno metodo.



Slika 4.21: Maksimalna metoda hierarhičnega gručenja na podatkih za primerjavo z minimalno in povprečno metodo.

Del III

Delne urejenosti in hierarhično gručenje

Poglavje 5

Grafi podatkovnih hierarhij, dobljeni tekom gručenja

V zadnjem poglavju si bomo pogledali povezavo med matematičnim in računalniškim delom diplomske naloge. Pokazali bomo, da so podatkovne hierarhije, ki jih dobimo tekom izvajanja algoritmov hiearhičnega gručenja, linearno urejene.

Podatki se na podlagi različnih metod združevanja, ki smo si jih pogledali v prejšnjih dveh poglavjih, razdelijo v gruče. Matematično gledano takšna razdelitev predstavlja ekvivalenčno relacijo. Dva podatka sta ekvivalentna, če sta v isti gruči. Ker smo podatke prej razporejali glede na podobnost/različnost, sta dva podatka torej v isti gruči natanko takrat, ko sta si podobna.

V primeru n podatkov je torej v algoritmu za hierarhično gručenje preteklo $n-1$ korakov, preden se je le ta izvedel do konca. Na vsakem koraku smo tako dobili novo razporeditev podatkov v gruče.

Na 1. koraku vsak podatek predstavlja svoj ekvivalenčni razred, torej dno dendrograma predstavlja n ekvivalenčnih razredov. Pri združevanju gruč nastajajo novi ekvivalenčni razredi in vrh dendrograma predstavlja en velik ekvivalečni razred oziroma gručo, kjer so vsi podatki ekvivalentni.

Tekom izvajanja algoritmov gručenja nad množico podatkov D dobimo množico ekvivalenčnih relacij V, za katero velja naslednji izrek:

Izrek 5.1 *Naj bo D množica podatkov in V množica ekvivalenčnih relacij, ki jih dobimo pri izvajanju algoritma združevalno hierarhičnega gručenja nad D (algoritem 1). Potem je reducirani graf podatkovne hierarhije $H = (D, V)$ pot.*

Dokaz. Dokaz z indukcijo.
I.P.: Graf je pot z E_i kot najbolj grobim vozliščem. Iz tega sledi, da je $E_j \preceq E_i$ za vsak $j \leq i$ in $E_j \not\preceq E_i$ za vsak $j < i$.

i) $k = 1$:
Le ena sama particija, ki je razdeljena na k ekvivalenčnih razredov.

Slika 5.1: $k = 1$

Graf je pot z le 1 vozliščem. Takšna pot je dolžine 0.

ii) $k \to k + 1$:
Recimo, da je reducirani graf (X, P_k) pot na k vozliščih.

E_{i+1} dobimo iz E_i z združevanjem dveh razredov E_i. Vsako novo ekvivalenčno relacijo namreč dobimo tako, da združimo dve prejšnji.

Dokazujemo:

- Za vsak korak bomo dokazali, da je relacija na prejšnjem koraku finejša od relacije na trenutnem, prav tako pa relaciji nista enaki. Želimo torej dokazati, da je $E_i \preceq E_{i+1}$ in $E_i \neq E_{i+1}$.

  Drugi del dokazovane trditve je očiten. Zaradi zgornje trditve, da vsako novo ekvivalenčno relacijo dobimo z združevanjem dveh iz prejšnjega koraka, velja $E_i[x] \subseteq E_{i+1}[x]$. Če je $E_i[x]$ eden od razredov, ki so bili združeni na koraku $(i + 1)$, potem je vsebovanost stroga, v nasprotnem primeru velja $E_{i+1}[x] = E_i[x]$.
  Ker je $E_i[x]$ vsebovana v $E_{i+1}[x]$ na vsakem koraku, po definiciji o finosti dveh relacij (definicija 2.5) sledi, da je $E_i \preceq E_{i+1}$.

□

- Da bo reducirani graf pot, je potrebno poleg finosti pokazati še, da E_{i+1} pokriva E_i.

  Po definiciji pokritja (definicija 1.19) velja, da je element x pokrit z y natanko

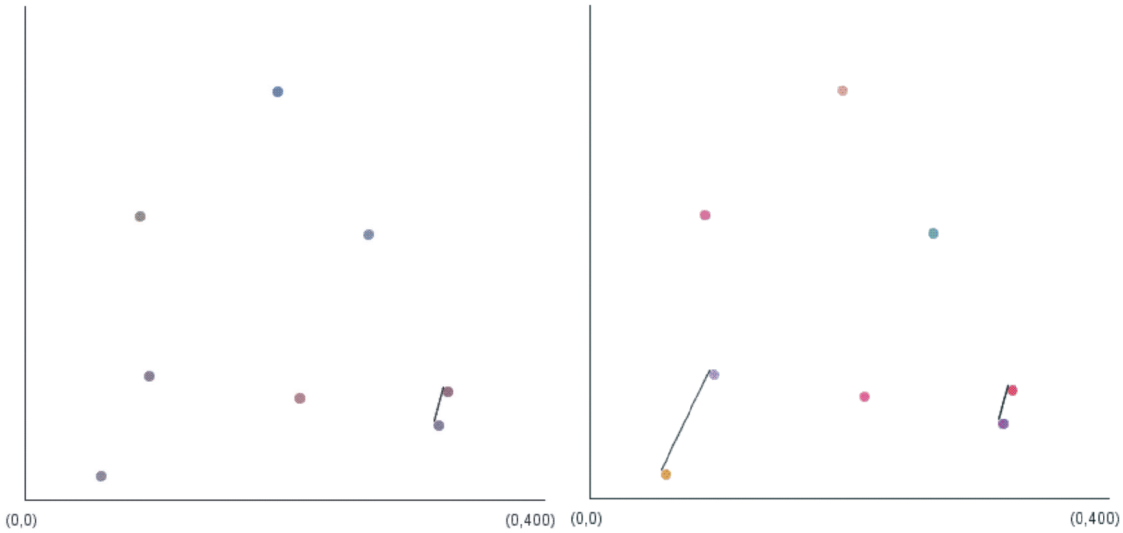
takrat, ko je $x < y$ in ne obstaja tak element z iz iste množice, da velja $x < z < y$.

Torej predpostavimo, da velja $E_i \preceq R \preceq E_{i+1}$. Po I.P. velja, da je $E_j \preceq E_i$ za vsak $j \leq i$ in da $E_j \not\preceq E_i$ za vsak $j < i$.
Torej velja, da je $R = E_i$ ali $R = E_{i+1}$ in E_{i+1} pokriva E_i po definiciji.

□

Iz tega sledi, da je (X, E_{i+1}) pot na $(i+1)$ vozliščih, kjer je E_{i+1} najmanj fino krajišče poti.

□

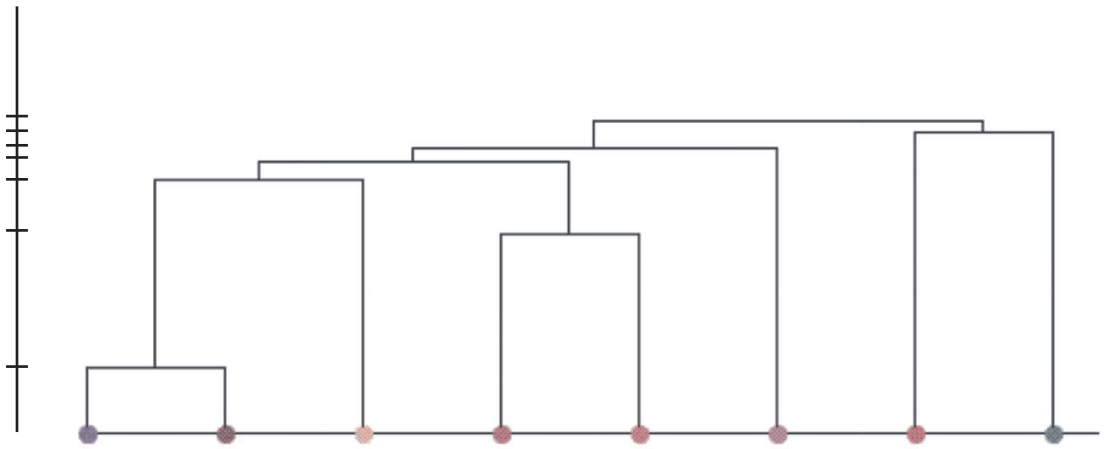


Slika 5.2: E_1 levo in E_2 desno.

Posledica 5.2 *Za vsak k velja $E_1 \preceq E_2 \preceq E_3 \preceq \ldots \preceq E_k$, kjer gre za združevalno hierarhično gručenje. Podobno bi lahko pokazali, da velja $E_1' \succeq E_2' \succeq E_3' \succeq \ldots \succeq E_k'$, kjer gre za cepitveno hierarhično gručenje.*

Posledica 5.3 *Vsaka razporeditev gruč na višjem nivoju dendrograma je "bolj groba" od razporeditve na prejšnjem nivoju.*

Izrek 5.4 *Naj bo D množica podatkov in V množica ekvivalenčnih relacij, ki jih dobimo tekom gručenja množice P z algoritmom K-voditeljev. Potem je graf podatkovne hierarhije $H = (D, V)$ brez povezav.*



Slika 5.3: Vsaka razporeditev na višjem nivoju je "bolj groba" od tiste na nižjem.

Dokaz. Dokaz s protislovjem.
Dokazujemo, da za vsak j, k ekvivalenčna relacija E_j ni primerljiva z E_k.

Predpostavimo nasprotno.
Brez škode za splošnost predpostavimo $E_1 \preceq E_2$.

Vsaka relacija E_1, E_2 ima k ekvivalenčnih razredov.
Naj bodo ekvivalenčni razredi relacije E_2 $R_1, R_2, ..., R_k$.

Za vsak i velja: za vsak $x \in R_i$ je $E_1[x] \subseteq R_i$, saj smo predpostavili, da velja $E_1 \preceq E_2$.

Če obstaja tak $i_0 \ni$: obstaja $x \in R_{i_0} \ni$: $E_1[x] \neq R_{i_0}$, potem ima E_1 vsaj $k+1$ ekvivalenčnih razredov, vsaj po enega za vsak R_i, pri čemer $i \neq i_0$ in vsaj dva za R_{i_0}. Prišli smo do protislovja, saj vemo, da ima E_j k ekvivalenčnih razredov za vsak j.

Torej $E_1[x] = R_i$ za vsak $x \in R_i$, za vsak i.
Torej $E_1 = E_2$. Spet smo dobili protislovje, saj za vsak j, k: $j \neq k$ velja $E_j \neq E_k$ po lastnostih algoritma K-voditeljev.

Torej naša zgornja predpostavka ne drži in E_j ni primerljiva z ekvivalenčno ralacijo E_k.

□

Na vsakem koraku algoritma dobimo razdelitev objektov na množice, katerih družina predstavlja podatkovno hierarhijo. Za hierarhične algoritme gručenja smo pokazali, da je le-ta delno urejena z relacijo "biti finejši" oziroma da je reducirani graf podatkovne hierarhije enak poti na $n-1$ vozliščih. Za nehierarhično metodo K-voditeljev pa kot rezultat dobimo graf podatkovne hierarhije, ki je brez povezav.

Literatura

[1] Bohanec, M. *Odkrivanje znanja v podatkih* (online). (citirano 24. 1. 2009). Dostopno na naslovu `http://kt.ijs.si/MarkoBohanec/ai/KDD.PDF`.

[2] Brandstädt, A. et al. *Graph classes*. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philiadelphia, 1999. ISBN 0-89871-432-X (pbk.).

[3] Bulbeck, D. *Seriation in Hierarchical Clustering* (online). (citirano 2. 3. 2009). Dostopno na naslovu `http://arts.anu.edu.au/bullda/dendrograms.html`.

[4] *Clustering Methods* (online). SAS Institute Inc., Cary, NC, USA. 2003. (citirano 31. 1. 2009). Dostopno na naslovu: `http://support.sas.com/onlinedoc/913/-getDoc/en/statug.hlp/cluster_sect12.htm`.

[5] Ferligoj, A. *Razvrščanje v skupine: zbirka Metodološki zvezki št. 4* (online). Raziskovalni inštitut Fakultete za sociologijo, politične vede in novinarstvo, Ljubljana, 1989. (citirano 25. 1. 2009). Dostopno na naslovu `http://vlado.fmf.uni-lj.si/vlado/podstat/mva/MZ4.pdf`.

[6] Gasar, S. in Bohanec, M. *Odkrivanje zakonitosti učnega uspeha z rudarjenjem podatkov* (online). (citirano 18. 1. 2009). Dostopno na naslovu `http://silvana.telesat.si/dokumenti/KDD2.pdf`.

[7] Gupta, G.K. et al. *Distance based clustering of association rules*. Intelligent Engineering Systems Through Artificial Neural Networks, (Proceedings 1999), 759–764, ASME Press, 1999.

[8] Han, J. in Kamber, M. *Data mininig, concepts and techniques*. Second edition. San Francisco: Morgan Kaufmann publishers, 2006.

[9] *Hierarchical agglomerative clustering* (online). (citirano 31. 1. 2009). Dostopno na naslovu: `http://nlp.stanford.edu/IR-book/html/htmledition/hierarchical-agglomerative-clustering-1.html`.

[10] *Hierarhična klastrska analiza* (online). Oddelek za psihologijo, Filozofska fakulteta Univerze v Ljubljani. (citirano 25. 1. 2009). Dostopno na naslovu: http://193.2.70.110/Vaje/PM1/sklop2/MPP-klastrska.pdf.

[11] Hou, D. *Lecture 18: Clustering & classification* (online). (citirano 1. 2. 2009). Dostopno na naslovu: http://www.cs.duke.edu/courses/fall03/cps260/notes/lecture18.pdf.

[12] Juvan, M. in Potočnik, P. *Teorija grafov in kombinatorika*. Ljubljana: DMFA, 2000.

[13] Kononenko, I. *Strojno učenje*. Ljubljana: Založba FE in FRI, 2005.

[14] Košmelj, K. in Breskvar Žaucer, L. *Metode za razvrščanje enot v skupine; osnove in primer* (online). (citirano 18. 1. 2009). Dostopno na naslovu: http://aas.bf.uni-lj.si/september2006/11kosmelj.pdf.

[15] Kožuh, B. *Uporaba rudarjenja po spletu za učinkovito poosebljanje in načrtovanje spletišč* (online). (citirano 18. 1. 2009). Dostopno na naslovu: http://www.cek.ef.uni-lj.si/magister/kozuh3276.pdf.

[16] Podpečan, V. *Koristnostno podatkovno rudarjenje* (online). (citirano 17. 1. 2009). Dostopno na naslovu: http://kt.ijs.si/theses/dipl_vid_podpecan.pdf.

[17] Prijatelj, N. *Matematične strukture I: Množice - relacije - funkcije*. 5. izdaja. Ljubljana: DMFA, 1996.

[18] Računalniška matematika. *Grafi (predavanja)* (online). (citirano 31. 12. 2008). Dostopno na naslovu: www.pef.upr.si/MARA/2/RaMa/predavanja/Lekcija15.ppt.

[19] Rupnik, R. in Krisper, M. *Aplikativni sistemi odkrivanja zakonitosti v podatkih* (online). (citirano 17. 1. 2009). Dostopno na naslovu http://amor.fri.uni-lj.si/mas/gradivo_in_rezultati.html.

[20] Sadoyan, H. *Cluster Analysis* (online). (citirano 1. 2. 2009). Dostopno na naslovu: http://www-personal.engin.umd.umich.edu/~shovhans/DMArticles/Clustering%20-%20Cluster%20Analysis.pdf.

[21] *Slovar infromatike* (online). (citirano 1. 2. 2009). Dostopno na naslovu: http://www.islovar.org/izpisclanka.asp?id=6674.

[22] Stata. *Stata help for cluster dendrogam* (online). (citirano 2. 3. 2009). Dostopno na naslovu: http://www.stata.com/help.cgi?cluster+dendrogram.

[23] Štajdohar, M. in Standeker, M. *Priredba algoritma za iskanje asociacijskih pravil za delovanje na porazdeljenem sistemu* (online). (citirano 17. 1. 2009). Dostopno na naslovu: `http://lrk.fri.uni-lj.si/vseminar/`.

[24] Tan, P.N. et al. *Introduction to Data Mining* (online). (citirano 31. 1. 2009). Dostopno na naslovu: `http://www-users.cs.umn.edu/ kumar/dmbook/index.php`.

[25] Trotter, W.T. *Combinatorics and partially ordered sets: dimension theory*. Baltimore: The Johns Hopkins University Press, MD, 1992.

[26] West, D.B. *Introduction to graph theory*. Second edition. Upper Saddle River (NJ): University of Illinois - Urbana,Prentice-Hall, 2001.

[27] Wikipedia. *Dendrogram* (online). (citirano 1. 3. 2009). Dostopno na naslovu: `http://en.wikipedia.org/wiki/Dendrogram`.

[28] Wilson, R.J. in Watkins, J.J. *Uvod v teorijo grafov*. Ljubljana: DMFA, 1997.

[29] Zhao, Y. in Karypis, G. *Hierarchical Clustering Algorithms for Document Datasets* (online). (citirano 2. 2. 2009). Dostopno na naslovu: `http://glaros.dtc.umn.edu/gkhome/node/193`.

[30] Zupan, B. *Clustering* (online). Fakulteta za računalništvo in informatiko Univerze v Ljubljani. (citirano 31. 1. 2009). Dostopno na naslovu: `http://ucilnica.fri.uni-lj.si/file.php/104/ostalo/clustering.pdf`.

[31] Žerovnik, J. *Osnove teorije grafov in diskretne optimizacije*. Maribor: Fakulteta za strojništvo Univerze v Mariboru, Tiskarna tehniških fakultet v Mariboru, 2003.