UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna matematika – 1. stopnja

Adriana Kamnik

# Hierarhično grupiranje podatkov

Delo diplomskega seminarja

Mentorica: izred. prof. dr. Marjetka Krajnc

Ljubljana, 2015

Kazalo

## Hierarhično grupiranje podatkov

Povzetek

Z visoko in hitro rastjo informacijske tehnologije se srečujemo z veliko količino podatkov, ki morajo biti urejeni oziroma grupirani, da bi bili primerni za nadaljno obravnavo. Grupiranje ima pomemben pomen v znanosti in se uporablja predvsem v biologiji, statistiki, medicini, računalništvu in geografiji. Ena od metod urejevanja podatkov je hierarhično grupiranje v smiselne gruče, ki je predstavljeno v diplomskem seminarju. V prvem delu se seznanimo z osnovnimi pojmi, kot na primer s kakšnimi podatki se srečujemo, kakšna je ideja združevalnih algoritmov, z merili za merjenje podobnosti oziroma različnosti med podatki ter kako se hierarhična razvrstitev prikaže. V nadaljevanju so predstavljene različne združevalne metode, to so metode, ki združujejo podatke v gruče, novonastale gruče v večje gruče in tako naprej. Tri od obravnavanih metod so natančneje ilustrirane na primerih. Opisane so lastnosti metod ter primerjave med njimi. Na koncu je obravnavana še metoda, ki je primerna za grupiranje velikega števila podatkov.

## Hierarchical clustering

Abstract

With a large and fast growth in information technology, comes a large amount of data that needs to be organized in order to be appropriate for further analysis. One of these organizational methods is the clustering of data into meaningful clusters, which is widely used in science, particularly in biology, geography, and medicine, as well as in statistics and computer science. In this work a hierarchical clustering of data into meaningful clusters is presented. First the basic notions of hierarchical clustering are introduced along with the idea of agglomerative clustering, and the structure of the data. Agglomerative clustering is the notion of combining data objects into clusters, and those clusters into larger clusters, and so forth, creating a hierarchy. An important part of hierarchical clustering are the measurements of the similarities and differences between the objects. The result of the hierarchical clustering group of the training data is a tree of clusters, which is called a dendrogram. Next, agglomerative methods are explored including three detailed examples with their properties explained. Finally a method that can be used for large amount of data is examined.

**Math. Subj. Class. (2015):** 68Pxx, 62Hxx, 68Wxx

**Ključne besede:** Gruča, grupiranje, mera različnosti, dendrogram, združevalne metode

**Keywords:** Cluster, clustering, meansurement of dissimilarity, dendrogram, agglomerative method

# 1. Uvod

Z visoko rastjo informacijske tehnologije se srečujemo z obilico podatkov. Želimo imeti čim enostavnejši dostop do podatkov, zato jih moramo znati razvrstiti v smiselne gruče glede na določeno lastnost. Tako lahko z njimi lažje operiramo, jih analiziramo ali jih uporabimo za razne modele napovedi v različnih znanostih. Podatke želimo razvrstiti v smiselne gruče glede na izbrano podobnost. To storimo na različne načine in v splošnem poznamo več kategorij razvrščanj: zaporedni algoritmi, hierarhično grupiranje podatkov, grupiranje podatkov, ki temeljijo na optimizaciji funkcije ter druge kategorije. V diplomskem seminarju je predstavljeno hierarhično grupiranje podatkov.

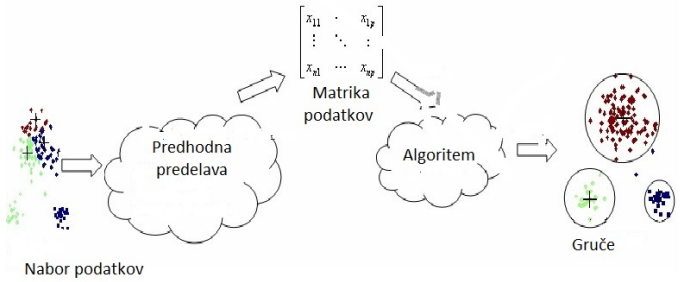
Hierarhično grupiranje podatkov ima pomemben pomen v podatkovnem rudarjenju. Grupiranje podatkov je delitev le teh na gruče s podobnimi lastnostmi. Uporablja se predvsem v statistiki, biologiji, medicini, računalništvu, robotiki, geografiji, itn.

# 2. Na splošno o hierarhičnem grupiranju

## 2.1. **Osnovni pojmi.**

**Gruča** (angl. Cluster) je skupina podobnih stvari ali ljudi. **Grupiranje** (angl. Clustering) je razvrstitev objektov na gruče, tako da so si objekti znotraj gruč čim bolj podobni, objekti različnih gruč pa čim bolj različni.

**Naloga razvrščanja v gruče** je sledeča. Imamo nabor podatkov, ki jih želimo razvrstiti v smiselne gruče tako, da so si podatki znotraj gruče čim bolj podobni. Preden začnemo z razvrščanjem moramo naš problem dobro preučiti.




Slika 1. Potek grupiranja.

Na sliki 1 je v grobem prikazan potek grupiranja podatkov, ki je sledeč:

- nabor podatkov,
- predhodna obdelava podatkov:
  - čiščenje podatkov,
  - integracija podatkov,
  - redukcija podatkov,
  - transformiranje podatkov,

– standardizacija podatkov,

- tvorba matrike podatkov,
- izbira ustrezne metode,
- določitev števila gruč,
- algoritem nam vrne gruče.

V realnosti nimamo idealnih podatkov, zato jih moramo dostikrat predhodno predelati v ustreznejšo obliko. Čiščenje podatkov tako pomeni, da manjkajoče vrednosti nadomestimo z nekih povprečjem ali z vrednostjo nič. Z integracijo združujemo podatke v večje in nove podatke. Redukcija podatkov pomeni, da nezanimive podatke izločimo iz našega grupiranja. V nekaterih primerih naši podatki niso v številski obliki, zato jim moramo prirediti realno število, da lahko z njimi operiramo. Slednjemu pravimo transformacija podatkov. Ko podatke predhodno predelamo, jih shranimo v **matriko podatkov**

$$M = \begin{bmatrix} x_{11} & x_{12} & \dots x_{1m} \\ x_{21} & x_{22} & \dots x_{2m} \\ \vdots & \vdots & \ddots \\ x_{n1} & x_{n2} & \dots x_{nm} \end{bmatrix},$$

kjer $i$-ta vrstica predstavlja $i$-ti podatek, $j$-ti stolpec pa predstavlja $j$-to spremeljivko (lastnost) podatka. Na tako pripravljenih podatkih lahko uporabimo metode, ki so opisane v poglavju 3. Pri predhodni predelavi pogosto uporabimo tudi standardizacijo. Uporabimo jo, ko so v povprečju vrednosti nekaterih podatkov mnogo večje kot vrednosti drugih. V tem primeru bi tiste z večjimi vrednostmi imele večjo težo pri razvrščanju. Da se temu izognemo, moramo vrednosti spremenljivk standardizirati. Najpogosteje se uporablja standardizacija, ki jo zapišemo kot $z_{ij} = \frac{x_{ij}-\mu_j}{\sigma_j}$, kjer je $\mu_j = \sum_{i=1}^{n} \frac{x_{ij}}{n}$ aritmetična sredina in $\sigma_j = \sqrt{\sum_{i=1}^{n} \frac{(x_{ij}-\mu_j)}{n}}$ standardni odklon te spremeljivke.

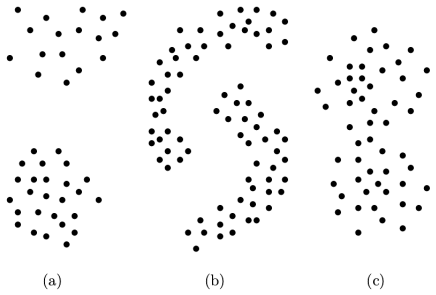
Podatke, ki so določeni z eno ali več spremenljivkami, je možno grafično predstaviti v eno ali več-razsežnem prostoru. Gručo, ki jo sestavljajo gosto poseljene točke, okoli nje pa prazen prostor ali prostor z redko posejanimi točkami, imenujemo naravna gruča. Cormack (1971) in Gordon (1981) sta za razkritje naravnih gruč podala dve želeni lastnosti gruč. To sta **homogenost** in **ločenost**. Na sliki 2 imamo v primeru $(a)$ gruči, ki sta homogeni in ločeni. V primeru $(b)$ sta gruči ločeni in nehomogeni, ker sta točki na začetku in na koncu verige bolj oddaljeni kot točki na začetku obeh verig. V primeru $(c)$ pa sta gruči homogeni in neločeni, saj ju veže nekaj točk ([1]).

Matematičen cilj grupiranja je najti takšno razvrstitev, pri kateri bo vsak podatek v samo eni gruči. Gruče se torej ne prekrivajo. To zapišemo kot

$$X = C_1 \cup C_2 \cup \dots \cup C_k \cup C_{osamelci}, \quad C_i \cap C_j = \emptyset, \quad i \neq j,$$

kjer je $X$ nabor vseh podatkov in $C_1, \dots, C_k$ predstavljajo gruče.

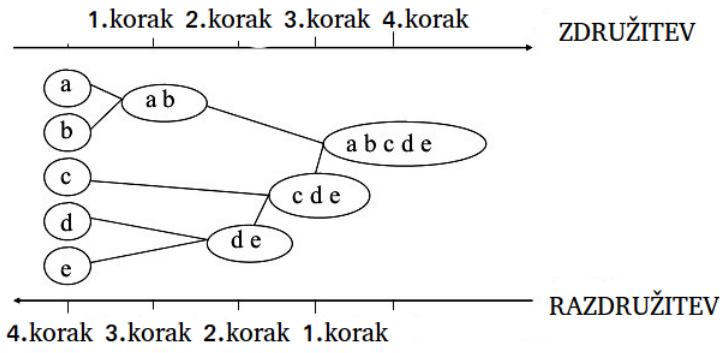
**Osamelci** so tisti podatki, ki se zelo razlikujejo od ostalih in jih algoritem ne razvrsti v nobeno gručo. Vsak algoritem na svoj način izloči osamelce in vsako




Slika 2. Tipi razvrstitve.

podatkovje vsebuje osamelce. Razdalja med osamelcem in najbližjo gručo je večja kot razdalja med podatki, ki so v najbližji gruči.

Hierarhične algoritme ločimo na združevalne in razdruževalne. Pri združevalnih predpostavimo, da je na začetku vsak podatek ena gruča (singelton). Podatki se postopoma združujejo in s tem se zmanjšuje število gruč. Če ne zahtevamo drugače, nam algoritem vrne eno gručo. Nasprotno, pri razdruževalnih algoritmih predpostavimo, da so na začetku vsi podatki v eni gruči, ki se postopoma razdružujejo (ločujejo). Ločujejo se, dokler tega ne določi zaustavitveni pogoj. Na sliki 3 je na enostaven način prikazana ideja, kako potekajo združevalne in razdruževalne metode. V algoritmu 1 pa je predstavljena ideja združevalnih algoritmov. Vhodna podatka sta nabor podatkov in število gruč, ki jih želimo imeti.




Slika 3. Združevalne in razdruževalne metode.

V diplomskem seminarju bodo predstavljeni združevalni algoritmi. Predstavljene bodo metode enojnega, popolnega in povprečnega povezovanja, metoda centroidov in medianov ter Wardova metoda. Predstavljena bo tudi ideja algoritma CURE.

Želimo imeti **stabilne razvrstitve**. To pomeni, da se dobljena razvrstitev bistveno ne spremeni z dodajanjem novih objektov v nabor podatkov, z dodajanjem novih spremenljivk v nabor podatkov ali z vsiljenimi napakami na nekaj posameznih vrednostih merjenih spremenljivk ([1]).

**Algorithm 1** Združevalni algoritmi (bootom-up)

1: **procedure** ZDRUŽITEV($X$, $k$)
2: Na začetku razvrstitve je vsak podatek v svoji gruči.
3: **repeat**
4: Poišče najbolj podobni gruči glede na izbrano mero podobnoosti in ju združi.
5: Izračuna mero podobnosti med novo gručo in ostalimi gručami.
6: **until** Ne dobi $k$ gruč oziroma dokler tega ne določi zaustavitveni pogoj.
7: **end procedure**

Pomembno je, da na vsakem problemu poskušamo izvesti čim več metod in da primerjamo razvrstitve med seboj. Nato se odločimo, katera razvrstitev je najprimernejša. Med drugim je pri izbiranju metode dobro vedeti, kakšen tip gruč želimo razkriti na naših podatkih, saj bomo kasneje videli, da metode določajo obliko. Naši podatki so lahko eliptične, okrogle, verižne oblike. Gruče so lahko med seboj dobro ločene ali prekrivajoče.

### 2.2. **Mera podobnosti oziroma različnosti.**

Združujemo podatke, ki so si podobni. V ta namen potrebujemo mero podobnosti oziroma mero različnosti saj želimo ugotoviti, kateremu podatku je določen podatek najbolj podoben. Mera različnosti je preslikava $d : X \times X \to \mathbb{R}$, ki vsakemu paru podatkov $(x, y)$, $x, y \in X$, priredi realno število in za katero veljajo naslednje lastnosti:

- $d(x, y) \geq 0$,
- $d(x, x) = 0$,
- $d(x, y) = d(y, x)$.

Če $d$ zadošča še pogojema

- $d(x, y) = 0 \Rightarrow x = y$,
- $d(x, y) \leq d(x, z) + d(y, z)$ za vsak $z \in X$,

ji pravimo razdalja oziroma metrika. Najpogostje se uporablja Evklidska razdalja. Za

$$x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \quad \text{in} \quad y = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n,$$

je Evklidska razdalja definirana kot

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} = ||x - y||_2.$$

Včasih uporabimo tudi razdaljo Manhattan oziroma prvo vektorsko normo razlike, ki je definirana kot

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| = ||x - y||_1.$$

Splošneje lahko definiramo razdaljo

$$d(x, y) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} = ||x - y||_p, \quad p > 0,$$

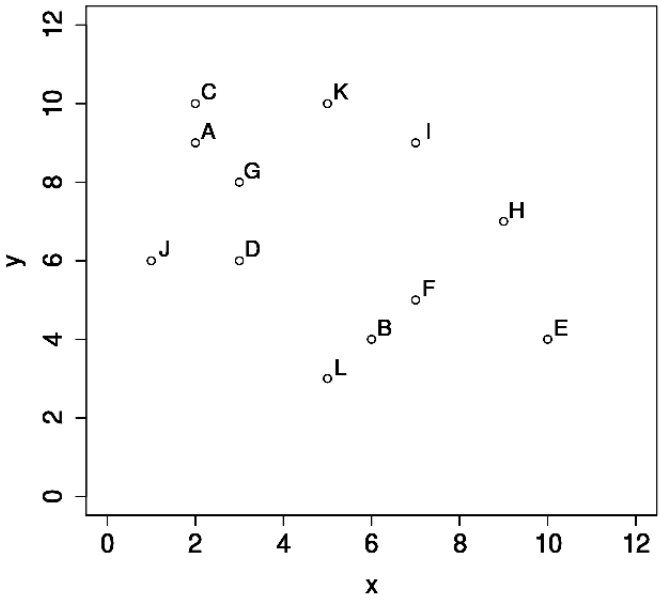
ki ji pravimo razdalja Minkowskega. Za $p = 1$ dobimo razdaljo Manhattan, za $p = 2$ pa Evklidsko razdaljo.

V diplomskem seminarju je največkrat uporabljena Evklidska razdalja. Naj opomnimo, da imamo dve vrsti mer različnosti, in sicer mero različnosti med podatkoma in mero različnosti med gručama. V diplomskem seminaraju bosta obe meri označeni z $d$. Mero različnosti $d$ med novonastalo gručo in ostalimi gručami izračunamo na več različnih načinov in ti določajo različne metode hierarhičnega grupiranja, kar je natančneje predstavljeno v naslednjem poglavju.

## 2.3. **Denrogram.**

Grupiranje predstavimo z dendrogramom. Beseda dendrogram izhaja iz grščine. *Dendron* pomeni drevo, *gramma* risanje. Dendrogramu pravimo drevo združevanja in z njim prikažemo hierarhično grupiranje. Listi tega drevesa so podatki, točke združitve so gruče. Naslednice vsake točke združitve so gruče, iz katerih je nova gruča nastala. Os $x$ predstavlja podatke, $y$ os pa mero različnosti oziroma nivo združevanja. Najprej se združijo podatki, ki imajo najmanjšo mero različnosti. Z dendrogramom predstavimo, kateri podatki so si podobni in kateri ne.

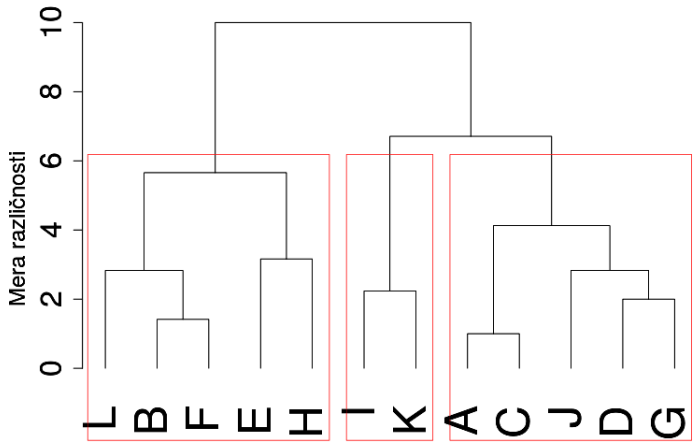
Na sliki 4 imamo v koordinatnem sistemu 10 točk. Zanima nas, kako se te točke postopoma hierarhično združujejo. Za hierarhično grupiranje smo uporabili metodo popolnega povezovanja, ki je predstavljena v podpoglavju 3.3.



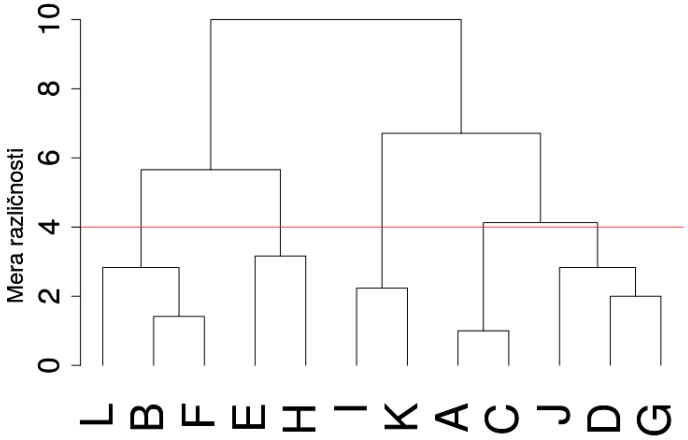

Slika 4. Točke, ki jih želimo grupirati.

Prednost hierarhičnega grupiranja je v tem, da število gruč izberemo šele na koncu. Na sliki 5 je predstavljeno, kako dobimo gruče, če določimo, koliko le teh želimo imeti. Na sliki 6 pa vidimo, kako dobimo gruče, pri katerih mera različnosti ni več

kot izbrana vrednost $h$, v tem primeru več kot 4. V prvem primeru dobimo 3 gruče, v drugem pa 5.




Slika 5. Združene točke iz slike 4. Izberemo 3 gruče.




Slika 6. Združene točke iz slike 4. Izbira gruč glede na predpisano maksimalno mero različnosti.

## 3. Združevalne metode

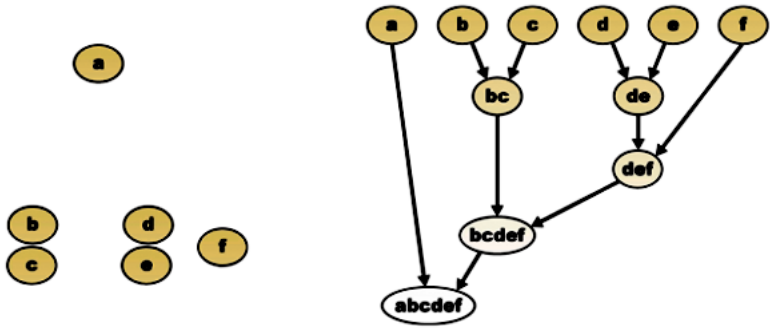
Te metode temeljijo na **povezovalni matriki**

$$P = \begin{bmatrix} 0 & y_{1,2} & \cdots & y_{1,n-1} & y_{1,n} \\ y_{1,2} & 0 & \cdots & y_{2,n-1} & y_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ y_{1,n-1} & y_{2,n-1} & \cdots & 0 & y_{n-1,n} \\ y_{1,n} & y_{2,n} & \cdots & y_{n-1,n} & 0 \end{bmatrix},$$

kjer $ij$ - ti element predstavlja mero različnosti med $i$-tim in $j$-tim podatkom. Matrika je simetrična in se po združitvi gruč spremeni. Kako takrat izračunamo razdalje med gručami, bomo videli v nadaljevanju. Kadar za razdaljo uporabimo Evklidsko razdaljo, matriko P imenujemo matrika razdalj. Zaradi simetrije je dovolj shraniti le elemente v zgornjem trikotnem delu matrike. V nadaljevanju bodo zato povezovalne matrike predstavljene kot zgornje trikotna matrika.

### 3.1. **Uvodni primer.**

Poglejmo si enostaven primer, ki je prikazan na sliki 7. Imamo točke $(a), (b), (c), (d), (e), (f)$. Lahko si predstavljamo, da so to točke v koordinatnem sistemu, ki jih želimo smiselno združiti v gruče. Najprej se združita gruči $(b)$ in $(c)$ v gručo $(bc)$ ter $(d)$ in $(e)$ v $(de)$. Na drugem koraku se združita gruči $(de)$ in $(f)$ v $(def)$. Na tretjem koraku se gruči $(bc)$ in $(def)$ združita v $(bcdef)$. Na koncu se pa gruči $(a)$ in $(bcedf)$ združita v $(abcdef)$. Izkaže se, da vse metode, ki so predstavljene v naslednjem poglavju združijo gruče v enakem vrstnem redu združevanja, vendar na različnih nivojih združevanja. Vprašanje je, kaj vzameti za razdaljo od gruče $(de)$ do $(f)$. Ali razdaljo od $(d)$ do $(f)$, ali razdaljo od $(e)$ do $(f)$, ali kaj drugega. To nam pove metoda in v nadaljevanju bomo videli glavne razlike med njimi.



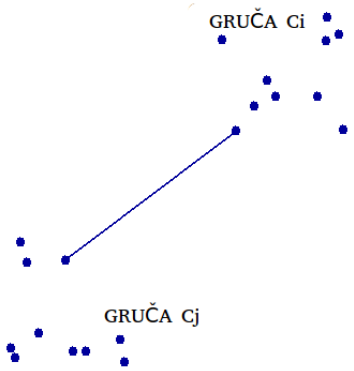

Slika 7. Primer združevanja.

### 3.2. **Metoda enojnega povezovanja.**

Metodi enojnega povezovanja pravimo tudi minimalna metoda. Pri metodi enojnega povezovanja uporabimo pravilo o najbližjemu sosedu. To pravilo pravi, da za razdaljo med gručami vzamemo razdaljo med podatkoma (vsak iz ene gruče), ki sta si najbližje (slika 8).

Torej je razdalja med dvema gručama enaka

$$d_{min}(C_i, C_j) = \min_{c_i \in C_i, c_j \in C_j} \|c_i - c_j\|.$$



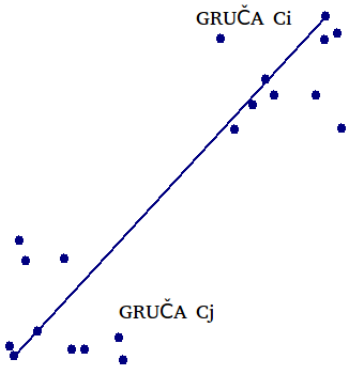

Slika 8. Razdalja med gručama pri metodi enojnega povezovanja.

### 3.3. **Metoda popolnega povezovanja.**

Metodi popolnega povezovanja pravimo tudi maksimalna metoda. Pri metodi popolnega povezovanja uporabimo pravilo o najbolj odddaljenemu sosedu. To pravilo pravi, da za razdaljo med gručami vzamemo razdaljo med podatkoma (vsak iz ene gruče), ki sta najbolj oddaljena (slika 9).

Torej je razdalja med dvema gručama enaka

$$d_{max}(C_i, C_j) = \max_{c_i \in C_i, c_j \in C_j} \|c_i - c_j\|.$$



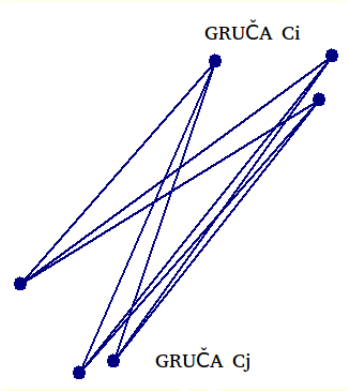

Slika 9. Razdalja med gručama pri metodi popolnega povezovanja.

### 3.4. **Metoda povprečnega povezovanja.**

Metodi povprečnega povezovanja pravimo tudi povprečna metoda. Z $n_i$ označimo število podatkov v gruči $C_i$. Za razdaljo med gručami vzamemo povprečje razdalj med vsemi možnimi pari podatkov (slika 10).

Razdalja med dvema gručama je torej enaka

$$d_{ave}(C_i, C_j) = \frac{1}{n_i \cdot n_j} \sum_{c_i \in C_i} \sum_{c_j \in C_j} \|c_i - c_j\|.$$




SLIKA 10. Razdalja med gručama pri metodi povprečnega povezovanja.

### 3.5. **Metoda centroidov.**

Centroid gruče $C_i$ označimo s $t_i$. To je aritmetična sredina gruče $C_i$ in se izračuna kot

$$t_i = \frac{1}{n_i} \sum_{c_i \in C_i} c_i,$$

kjer je $n_i$ število podatkov v $C_i$. Pri metodi centroidov za razdaljo med dvema gručama vzamemo kvadrirano Evklidsko razdaljo med njunima centroidoma, to je

$$d_{cen}(C_i, C_j) = ||t_i - t_j||_2^2.$$

Ko sta gruči $C_i$ in $C_j$ združeni, je centroid gruče $(C_i \cup C_j)$ enak $t_{ij} = \frac{n_i \cdot t_i + n_j \cdot t_j}{n_i + n_j}$.

### 3.6. **Metoda medianov.**

Gruči $C_i$ in $C_j$ združimo v $(C_i \cup C_j)$ po metodi centroidov. Če je $n_i > n_j$, potem se lahko zgodi, da je nov centroid bližje gruči $C_i$ kot $C_j$. Da se temu izognemo, ta metoda uporabi mediano $m_{ij} = \frac{1}{2}(t_i + t_j)$ za točko, od katere računamo nove razdalje od $(C_i \cup C_j)$ do ostalih gruč.

### 3.7. **Wardova metoda.**

Wardovo metodo oziroma Wardovo metodo minimalne variance je uvedel Joe H. Ward (1963). Uporabil je vsoto kvadriranih odklonov. V tem razdelku uporabljamo Evklidsko razdaljo in označimo $||.|| = ||.||_2$.

Razdalja med gručama se vrednoti z izgubo informacije, ki jo povzroča združevanje dveh gruč v novo gručo. Pri tem se poveča vsota kvadriranih odklonov znotraj gruč (VKO). To je količina, ki jo poznamo iz analize variance. Za gručo $C_i$ se ta količina izračuna kot vsota kvadriranih odklonov od centroida gruče $C_i$ ([4]), to je

$$\text{VKO}(C_i) = \sum_{c_i \in C_i} ||c_i - t_i||^2 = \sum_{c_i \in C_i} c_i^T c_i - 2 \sum_{c_i \in C_i} c_i^T t_i + \sum_{c_i \in C_i} t_i^T t_i \tag{1}$$

kjer je $t_i$ centroid gruče $C_i$. Upoštevamo

$$n_i t_i = \sum_{c_i \in C_i} c_i \tag{2}$$

in vstavimo v (1). Dobimo

$$\text{VKO}(C_i) = \sum_{c_i \in C_i} c_i^T c_i - 2n_i t_i^T t_i + n_i t_i^T t_i = \sum_{c_i \in C_i} c_i^T c_i - n_i t_i^T t_i. \tag{3}$$

Analogno poračunamo VKO za gručo $C_j$.

Pri Wardovi metodi je mera različnosti med dvema gručama definirana kot sprememba vrednosti VKO, ko smo $C_i$ in $C_j$ združili v $(C_i \cup C_j)$, in sicer

$$d(C_i, C_j) = \text{VKO}(C_i \cup C_j) - (\text{VKO}(C_i) + \text{VKO}(C_j)). \tag{4}$$

V (4) vstavimo (3) in dobimo

$$\begin{aligned} d(C_i, C_j) &= \sum_{c_{ij} \in C_i \cup C_j} ||c_{ij} - t_{ij}||^2 - \sum_{c_i \in C_i} ||c_i - t_i||^2 - \sum_{c_j \in C_j} ||c_j - t_j||^2 \\ &= \sum_{c_{ij} \in C_i \cup C_j} c_{ij}^T c_{ij} - n_{ij} t_{ij}^T t_{ij} - \sum_{c_i \in C_i} c_i^T c_i + n_i t_i^T t_i - \sum_{c_j \in C_j} c_j^T c_j + n_j t_j^T t_j \\ &= n_i t_i^T t_i + n_j t_j^T t_j - n_{ij} t_{ij}^T t_{ij}. \end{aligned} \tag{5}$$

V zadnjem koraku upoštevamo, da je $\sum_{c_{ij} \in C_i \cup C_j} c_{ij}^T c_{ij} = \sum_{c_i \in C_i} c_i^T c_i + \sum_{c_j \in C_j} c_j^T c_j$. Centroid združene gruče računamo kot pri metodi centroidov. Torej je

$$t_{ij} = \frac{n_i t_i + n_j t_j}{n_i + n_j}.$$

Upoštevamo, da je

$$t_{ij}^T t_{ij} = \frac{1}{(n_i + n_j)^2}(n_i^2 t_i^T t_i + 2n_i n_j t_i^T t_j + n_j^2 t_j^T t_j) \tag{6}$$

in vstavimo v (5). Dobimo

$$\begin{aligned} d(C_i, C_j) &= n_i t_i^T t_i + n_j t_j^T t_j - n_{ij} t_{ij}^T t_{ij} \\ &= n_i t_i^T t_i + n_j t_j^T t_j - n_{ij} \frac{1}{(n_i + n_j)^2}(n_i^2 t_i^T t_i + 2n_i n_j t_i^T t_j + n_j^2 t_j^T t_j) \\ &= \frac{n_i(n_i + n_j) t_i^T t_i + n_j(n_i + n_j) t_j^T t_j - n_i^2 t_i^T t_i - 2n_i n_j t_i^T t_j - n_j^2 t_j^T t_j}{n_i + n_j} \\ &= \frac{n_i n_j (t_i^T t_i - 2t_i^T t_j + t_j^T t_j)}{n_i + n_j} = \frac{n_i n_j}{n_i + n_j} ||t_i - t_j||^2. \end{aligned} \tag{7}$$

Razdalja med novonastalo gručo $(C_i \cup C_j)$ in gručo $C_k$ pa se izračuna kot

$$d(C_i \cup C_j, C_k) = \frac{(n_i + n_j) n_k}{(n_i + n_j + n_k)} ||t_{ij} - t_k||^2,$$

kjer je $t_{ij}$ centorid združene gruče $(C_i \cup C_j)$, $t_k$ centorid gruče $C_k$ in $n_\ell$ število podatkov v gruči $C_\ell$, $\ell = i, j, k$ ([7]).

Če imata gruči samo en objekt, $n_i = 1$ in $n_j = 1$, sta vsoti kvadriranih odklonov enaki 0 in se (7) spremeni v

$$d(C_i, C_j) = \frac{1}{2} ||c_i - c_j||^2.$$

Wardova metoda ima več različic pri računanju mer različnosti. Vgrajena funkcija v programskem jeziku R računa mero različnosti kot

$$d(C_i, C_j) = \frac{2n_i n_j}{n_i + n_j} ||t_i - t_j||^2,$$

mera različnosti med novonastalo gručo $(C_i \cup C_j)$ in gručo $C_k$ pa je

$$d(C_i \cup C_j, C_k) = \frac{(n_i + n_k)d(C_i, C_k) + (n_j + n_k)d(C_j, C_k) - n_k d(C_i, C_j)}{n_i + n_j + n_k}.$$

## 4. Lance -Williams-ova formula

Lance in Williams ([1]) sta uvedla formulo, po kateri se razdalja med novonastalo in obstoječo gručo izračuna po formuli

$$d(C_i \cup C_j, C_k) = \alpha_i d(C_i, C_k) + \alpha_j d(C_j, C_k) + \beta d(C_i, C_j) + \gamma |d(C_i, C_k) - d(C_j, C_k)|, \quad (8)$$

kjer so koeficienti $\alpha, \beta, \gamma$ primerno izbrani. Tabela 1 prikazuje koeficiente za prej opisane metode.

Tabela 1. Koeficienti metod

| METODA | $\alpha_i$ | $\alpha_j$ | $\beta$ | $\gamma$ |
|---|---|---|---|---|
| Minimalna | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | $-\frac{1}{2}$ |
| Maksimalna | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ |
| Povprečna | $\frac{n_i}{n_i+n_j}$ | $\frac{n_j}{n_i+n_j}$ | 0 | 0 |
| Centroidov | $\frac{n_i}{n_i+n_j}$ | $\frac{n_j}{n_i+n_j}$ | $-\frac{n_i n_j}{(n_i+n_j)^2}$ | 0 |
| Medianov | $\frac{1}{2}$ | $\frac{1}{2}$ | $-\frac{1}{4}$ | 0 |
| Wardova | $\frac{n_i+n_k}{n_i+n_j+n_k}$ | $\frac{n_j+n_k}{n_i+n_j+n_k}$ | $-\frac{n_k}{n_i+n_j+n_k}$ | 0 |

Za primer bomo izpeljali koeficiente za metodo popolnega povezovanja. Vemo, da je

$$\max(a, b) = \frac{1}{2}(a + b + |a - b|). \quad (9)$$

Mera različnosti pri metodi popolnega povezovanja je enaka

$$d_{max}(C_i, C_j) = \max_{c_i \in C_i, c_j \in C_j} \|c_i - c_j\|,$$

mera različnosti med novonostalo in obstoječo gručo pa je enaka

$$d_{max}(C_i \cup C_j, C_k) = \max(d(C_i, C_k), d(C_j, C_k)). \quad (10)$$

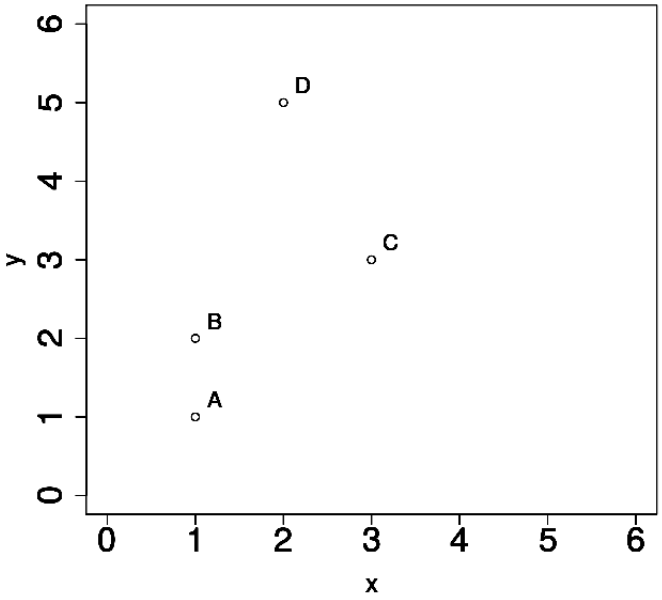
Če se v (10) upošteva dejstvo (9) dobimo

$$d_{max}(C_i \cup C_j, C_k) = \frac{1}{2}(d_{\max}(C_i, C_k) + d_{\max}(C_j, C_k) + |d_{\max}(C_i, C_k) - d_{\max}(C_j, C_k)|).$$

Torej je $\alpha_i = \frac{1}{2}$, $\alpha_j = \frac{1}{2}$, $\beta = 0$, $\gamma = \frac{1}{2}$.

## 5. Primer 1

V koordinatnem sistemu na sliki 11 imamo točke $A = (1, 1)$, $B = (1, 2)$, $C = (2, 3)$, $D = (2, 5)$. Te podatke vstavimo v matriko podatkov




Slika 11. Točke, ki jih bomo združevalni s tremi različnimi metodami.

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 3 & 3 \\ 2 & 5 \end{bmatrix} \end{array}.$$

Poračunamo vse medsebojne razdalje, pri čemer uporabimo Evklidsko razdaljo in te razdalje shranimo v povezovalno matriko

$$P = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & \sqrt{1} & \sqrt{8} & \sqrt{17} \\ 0 & 0 & \sqrt{5} & \sqrt{10} \\ 0 & 0 & 0 & \sqrt{5} \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

### Metoda enojnega povezovanja

**1.korak**: Na začetku je vsak podatek v svoji gruči.
**2.korak**: Gruči z najkrajšo razdaljo poiščemo tako, da najdemo najmanjši pozitiven element v matriki $P$. V našem primeru ta element predstavlja razdaljo med gručo $(A)$ in gručo $(B)$. Nato združimo gruči $(A)$ in $(B)$ v gručo $(A \cup B)$.

**3.korak**: Sedaj imamo gruče $(A \cup B)$, $(C)$ in $(D)$.
Poračunamo nove razdalje:

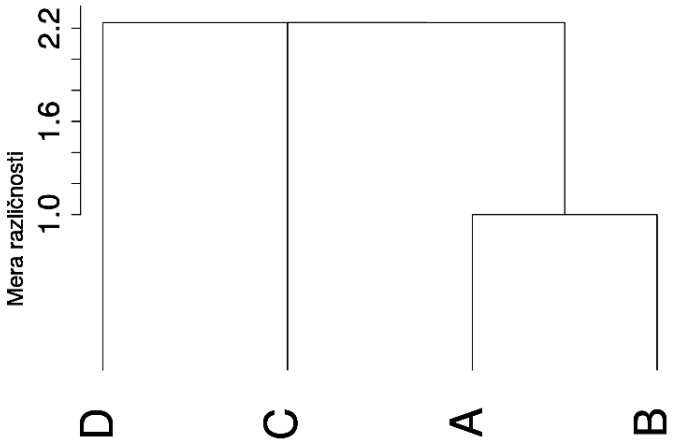
$$d(A \cup B, C) = \min(d(A, C), d(B, C)) = \min(\sqrt{8}, \sqrt{5}) = \sqrt{5},$$

$$d(A \cup B, D) = \min(d(A, D), d(B, D)) = \min(\sqrt{17}, \sqrt{10}) = \sqrt{10}.$$

Naša povezovalna matrika se spremeni v

$$P_{1E} = \begin{array}{c} \\ A \cup B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{ccc} A \cup B & C & D \end{array} \\ \begin{bmatrix} 0 & \sqrt{5} & \sqrt{10} \\ 0 & 0 & \sqrt{5} \\ 0 & 0 & 0 \end{bmatrix} \end{array},$$

kjer npr. element v prvi vrstici in tretjem stolpcu predstavlja razdaljo med $(A \cup B)$ in $(D)$.
**4.korak**: Ponovno poiščemo gruči z najkrajšo medsebojno razdaljo, tako da poiščemo najmanjši pozitiven element v matriki. Združimo gruči $(A \cup B)$ in $(C)$ in $(D)$ v novo gručo $(A \cup B \cup C \cup D)$. Prikaz z dendrogramom je na sliki 12.




Slika 12. Dendrogram pri minimalni metodi.

### Metoda popolnega povezovanja

**1.korak** Na začetku je vsak podatek v svoji gruči.
**2.korak** Gruči z najkrajšo razdaljo poiščemo tako, da najdemo najmanjši pozitiven element v matriki $P$. V našem primeru ta element predstavlja razdaljo med gručo $(A)$ in gručo $(B)$. Združimo gruči $(A)$ in $(B)$ v gručo $(A \cup B)$.
**3.korak** Sedaj imamo gruče $(A \cup B)$, $(C)$ in $(D)$.
Poračunamo nove razdalje:

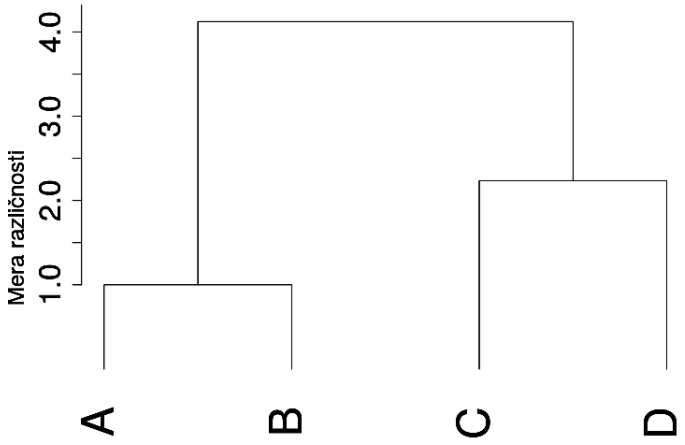
$$d(A \cup B, C) = \max(d(A, C), d(B, C)) = \max(\sqrt{8}, \sqrt{5}) = \sqrt{8},$$

$$d(A \cup B, D) = \max(d(A, D), d(B, D)) = \max(\sqrt{17}, \sqrt{10}) = \sqrt{17}.$$

Naša povezovalna matrika se spremeni v

$$P_{1K} = \begin{array}{c} \\ A \cup B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{ccc} A \cup B & C & D \end{array} \\ \begin{bmatrix} 0 & \sqrt{8} & \sqrt{17} \\ 0 & 0 & \sqrt{5} \\ 0 & 0 & 0 \end{bmatrix} \end{array}.$$

**4.korak** Ponovno poiščemo gruči z najkrajšo medsebojno razdaljo, tako da poiščemo najmanjši pozitiven element v matriki. Združimo gruči $(C)$ in $(D)$ v novo gručo $(C \cup D)$.
**5.korak** Združimo gruči $(A \cup B)$ in $(C \cup D)$ v eno gručo $(A \cup B \cup C \cup D)$. Prikaz z dendrogramom je na sliki 13.




Slika 13. Dendrogram pri maksimalni metodi.

### Wardova metoda

Poračunamo vse medsebojne razdalje in te razdalje shranimo v povezovalno matriko

$$P = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & \frac{1}{2} & \frac{8}{2} & \frac{17}{2} \\ 0 & 0 & \frac{5}{2} & \frac{10}{2} \\ 0 & 0 & 0 & \frac{5}{2} \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array},$$

kjer je npr. $d(A, B) = \frac{1}{2} d(t_A, t_B) = \frac{1}{2} 1 = \frac{1}{2}$.
**1.korak** Na začetku je vsak podatek v svoji gruči.
**2.korak** Gruči z najkrajšo razdaljo, poiščemo tako, da najdemo najmanjši pozitiven element v matriki $P$. V našem primeru ta element predstavlja razdaljo med gručo $(A)$ in gručo $(B)$, zato združimo gruči $(A)$ in $(B)$ v gručo $(A \cup B)$.
Centroid gruče $(A \cup B)$ je enak

$$t_{AB} = \frac{1 \cdot (1,1) + 1 \cdot (1,2)}{2} = \left(1, \frac{3}{2}\right).$$

Vsota kvadriranih odklonov gruče $(A \cup B)$ pa je

$$(11)\quad \mathrm{VKO(A{\cup}B)} = \sum_{\mathrm{ci}\in \mathrm{A}\cup\mathrm{B}} (\mathrm{c_i}-\mathrm{t_{AB}}) = (1-1)^2+\left(1-\frac{3}{2}\right)^2+(1-1)^2+\left(2-\frac{3}{2}\right)^2 = \frac{1}{2}.$$

**3.korak** Sedaj imamo gruče $(A \cup B)$, $(C)$ in $(D)$. Za izračun $d((A \cup B) \cup C)$ uporabimo formulo

$$(12)\qquad d((A \cup B) \cup C) = \mathrm{VKO((A \cup B) \cup C)} - \mathrm{VKO((A \cup B))} - \mathrm{VKO(C)},$$

kjer za izračun $\mathrm{VKO((A \cup B) \cup C)}$ potrebujemo centroid gruče $(A \cup B \cup C)$, ki ga izračunamo kot

$$t_{ABC} = \frac{2 \cdot (1, \frac{3}{2}) + 1 \cdot (3, 3)}{3} = \left(\frac{5}{3}, 2\right).$$

Tako dobimo

$$\begin{aligned}(13)\qquad \mathrm{VKO((A \cup B) \cup C)} &= \left(\frac{3}{3}-\frac{5}{3}\right)^2 + (1-2)^2 + \left(\frac{3}{3}-\frac{5}{3}\right)^2 + \\ &\quad + (2-2)^2 + \left(\frac{9}{3}-\frac{5}{3}\right)^2 + (3-2)^2 \\ &= \frac{42}{9}.\end{aligned}$$

Vrednosti izraza (11) in (13) vstavimo v (12), kjer upoštevamo, da je $\mathrm{VKO(C)} = 0$. Dobimo

$$d((A \cup B) \cup C) = \frac{42}{9} - (\frac{1}{2} + 0) = \frac{75}{18} \doteq 4.1\overline{6}\ .$$

Do istega rezultata bi prišli z uporabo formule
(14)

$$d((A{\cup}B), C) = \frac{n_{AB} \cdot n_C}{n_{AB} + n_C} \cdot d(t_{AB}, t_C) = \frac{2}{3}\left((1-3)^2 + \left(\frac{3}{2}-3\right)^2\right) = \frac{2}{3} \cdot \frac{25}{2} = 4.1\overline{6}.$$

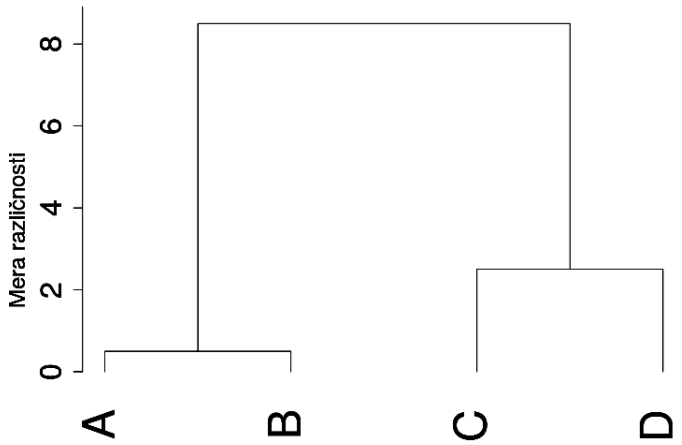
Podobno dobimo razdaljo med gručo $(A \cup B)$ in $(D)$, ki je enaka

$$(15)\qquad d((A \cup B), D) = \frac{2}{3} \cdot \frac{53}{4} \doteq 8.8\overline{3}.$$

Naša povezovalna matrika se spremeni v

$$P_{1W} = \begin{array}{c} \\ A \cup B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{ccc} A \cup B & C & D \end{array} \\ \begin{bmatrix} 0 & 4.1\overline{6} & 8.8\overline{3} \\ 0 & 0 & 2.5 \\ 0 & 0 & 0 \end{bmatrix} \end{array},$$

**4.korak** Ponovno poiščemo gruči z najkrajšo medsebojno razdaljo, tako da poiščemo najmanjši pozitiven element v matriki. Združimo gruči $(C)$ in$(D)$ v gručo $(C \cup D)$.
**5.korak** Združimo gruči $(A \cup B)$ in $(C \cup D)$ v eno gručo $(A \cup B \cup C \cup D)$. Prikaz z dendrogramom je na sliki 17.




Slika 14. Dendrogram pri Wardovi metodi.

## 6. Primer 2

Na spletni strani Svetovne banke smo dobili podatke o obsegu letne rasti BDP (v procentih) od leta 2009 do 2013 po različnih državah. Države smo želeli razvrstiti v smiselne gruče z uporabo hierarhičnega grupiranja. Kot smo že omenili, moramo ponavadi podatke predhodno predelati. Tako smo iz analize izločili države, o katerih nismo imeli informacij o rasti BDP-ja. Po čiščenju je to podatkovje vsebovalo 184 enot. Za grupiranje smo uporabili metodo enojnega povezovanja in popolnega povezovanja ter Wardovo metodo. Ko so se vsi algoritmi izvedli smo se odločili, da želimo imeti pet gruč.

### Metoda enojnega povezovanja

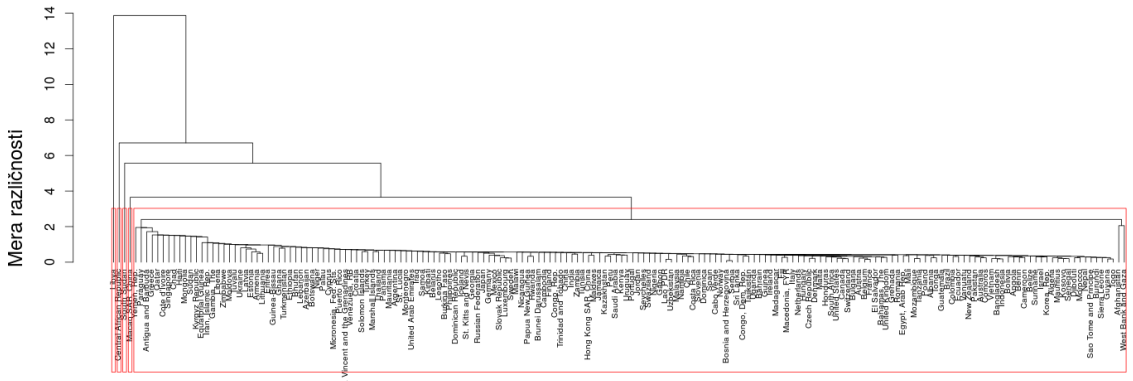
Iz slike 15 je razvidno, da izbira te metode ni v redu. Na tem primeru nazorno vidimo problem veriženja. Namesto, da bi se tvorile nove gruče, se te pridružujejo obstoječi in na koncu dobimo gručo, ki vsebuje skoraj vse podatke, preostale gruče pa vsebujejo le po en podatek.

### Metoda popolnega povezovanja

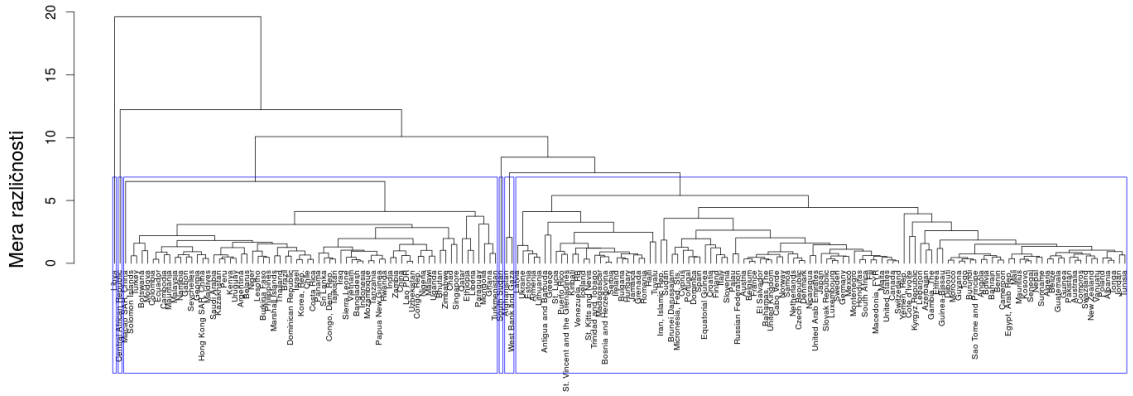
Iz slike 16 vidimo, da nam ta metoda zgenerira dve veliki gruči in tri manjše. Ta metoda je v tem primeru malenkost boljša kot metoda enojnega povezovanja. Na podlagi teh dveh dendrogramov lahko sklepamo, da sta ti dve metodi primernješi za manjše podatkovje. Posledično se pojavi problem veriženja in občutljivost osamelcev.

### Wardova metoda

Iz slike 17 vidimo, da je ta metoda najboljša od vseh treh. Lepo nam zgenerira štiri gruče, le ene države metoda ne razvrsti nikamor. Tudi če vzamemo tri ali šest gruč ta država ostane sama. Nato pogledamo kakšne lastnosti imajo države, ki so v

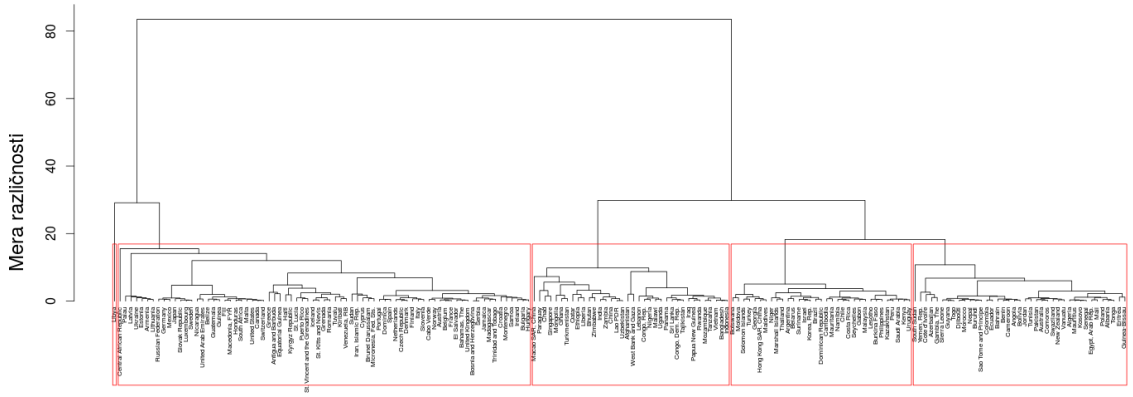



Slika 15. Metoda enojnega povezovanja.



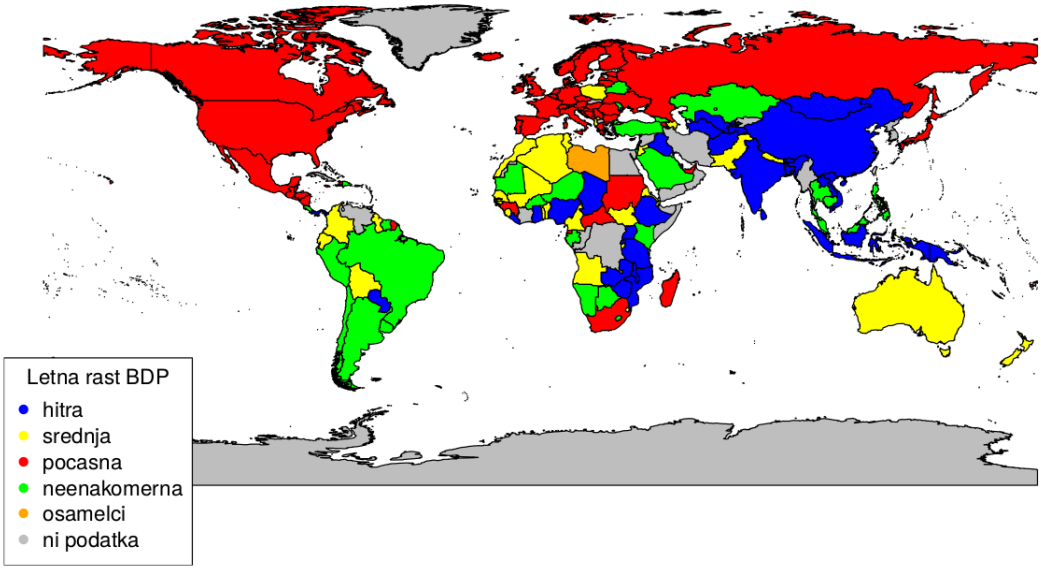

Slika 16. Metoda popolnega povezovanja.

isti gruči in jih primerno poimenujemo. Država, ki je algoritem ne razvrsti nikamor je Libija. Če pogledamo podatke, ugotovimo, da letna rast BDP-ja (v procentih) Libije drastično niha.




Slika 17. Wardova metoda.

Ker se imena držav na dendrogramu ne vidijo dobro, je rezultat Wardove metode predstavljen še z zemljevidom (slika 18).




Slika 18. Letna rast BDP-ja, Wardova metoda.

## 7. Splošni algoritem

V algoritmu 2 je predstavljena splošna ideja psevdokode za različne metode, ki so bile prej predstavljene. Vhodna podatka sta nabor podatkov in število gruč $k$. Kot izhod pa dobimo gruče in nivo združevanja.

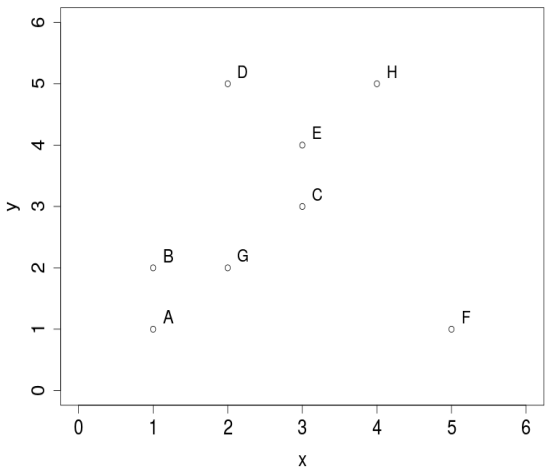
**Algorithm 2** Splošni algoritem

**procedure** HAC($X, k$)
  $C = \{C_i = \{x_i\} | x_i \in X\}$ ▷ Vsaka točka je ločena gruča.
  **for** $i = 1, \ldots, n$ **do**
    **for** $j = 1, \ldots, n$ **do**
      $P_{ij} = različnost(C_i, C_j)$ ▷ Povezovalna matrika.
    **end for**
  **end for**
  $m = 0$
  $L_m = 0$
  **while** $|C| > k$ **do** ▷ Algoritem se izvaja dokler imamo k gruč.
    $C_i, C_j = index.min(P)$ ▷ Poišče najbližji si gruči $C_i, C_j \in C$.
    $m = m + 1$ ▷ Beleži novonastalo gručo.
    $L_m = različnost(C_i, C_j)$ ▷ Beleži nivo združevanja.
    $C_{(i \cup j)} = C_i \cup C_j$ ▷ Gruči $C_i$ in $C_j$ združi v novo gručo $C_{(i \cup j)}$.
    $C = \{C - C_i - C_j\} \cup C_{(i \cup j)}$ ▷ Doda novo gručo obstoječim gručam.
    Spremeni matriko P ▷ Izbriše vrstici in stolpca v $P$, ki pripadata $C_i$ in $C_j$ in doda novo vrstico in stolpec, ki predstavljata mero različnosti med novonastalo gručo $C_{(i \cup j)}$ in ostalimi gručami iz $C$ glede na izbrano metodo.
  **end while**
  **return** $C, L$ ▷ Gruče,nivo združevanja
**end procedure**

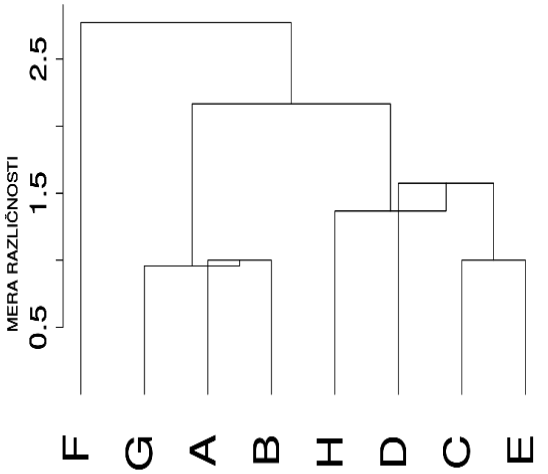
## 8.1. **Monotonost.**

Z Lance-Williamsovo formulo (8) lahko generiramo različne metode združevanja. Zanima nas, ali vsak nabor štirih koeficientov določa metodo, ki smiselno razkriva strukturo v podatkih. Kriterij, s katerim lahko ocenimo, ali je metoda smiselna ali ne, je **monotonost**. Ko združimo gruči $C_i$ in $C_j$ v novo gručo $(C_i \cup C_j)$, se namreč lahko zgodi, da je mera različnosti oziroma nivo združevanja, pri kateri združimo gruči $C_i$ in $C_j$, manjša od mere različnosti, pri katerih smo v prejšnih korakih združili podatke v gručo $C_i$ oziroma v gručo $C_j$ ([1]).

Hitro se da viditi, da sta metodi centroidov in medianov nemonotoni. Posledično dobimo nemonoton dendrogram. Primer nemonotonega dendrograma dobljenega z metodo centroidov je prikazan na sliki 20. Podoben dendrogram dobimo tudi z metodo medianov. Ostale metode so monotone.




Slika 19. Točke na katerih uvedemo metodo centroidov.




Slika 20. Nemonoton dendrogram dobljen z metodo centroidov na podatkih iz slike 19.

Metoda hierarhičnega združevanja v gruče, osnovana na formuli Lancea in Williamsa, zagotavlja monotone dendrograme natanko tedaj, ko so izponjeni naslednji trije pogoji:

- $\gamma \geq -\min(\alpha_1, \alpha_2)$
- $\alpha_1 + \alpha_2 \geq 0$
- $\alpha_1 + \alpha_2 + \beta \geq 1$

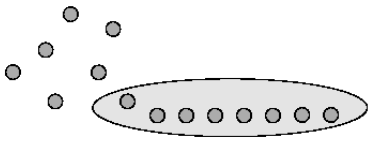
Za izpeljavo glej ([6]).

## 8.2. **Pozitivne in negativne lastnosti.**

Metode hierarhičnega grupiranja so preproste za razumevanje, ni potrebno vnaprej določiti števila gruč, potek združevanja nazorno prikažemo z dendrogramom in

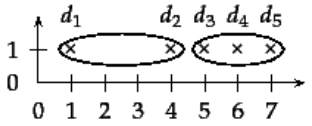
število gruč primerno izberemo na koncu, poleg tega pa imamo več različnih metod. Ko je podatek enkrat nekam razvrščen, ni možen prehod med gručami, čeprav se lahko v naslednjih korakih izkaže, da bi bilo bolje, če bi združevali drugače. V praksi največkrat uporabljamo minimalno, maksimalno in Wardovo metodo, saj se te med seboj najbolj razlikujejo.

Minimalna metoda dobro razkriva verižne podatke. Primerna je za podatke, ki so dobro ločeni med seboj ter nehomogeni. Problem te metode je veriženje. Namesto, da bi se tvorile nove gruče, se gruče singeltoni pridružijo že obstoječim gručam. Ta problem je prikazan na sliki 21.



Slika 21. Problem veriženja.

Maksimalna metoda dobro razkriva okrogle podatke. Primerna je za podatke, ki so dobro ločeni med seboj in homogeni. Ta metoda je občutljiva na osamelce. To pomeni, da metoda nameni veliko pozornosti podatkom, ki se ne prilagajajo že obstoječi gruči. Ta problem je prikazan na sliki 22. Maksimalna metoda nam da gruči $(d_1 \cup d_2)$ in $(d_3 \cup d_4 \cup d_5)$. Očitno pa bi bilo bolje, da bi bila gruča $(d_1)$ osamelec, ostali podatki pa bi bili v gruči $(d_2 \cup d_3 \cup d_4 \cup d_5)$.




Slika 22. Občutljivost osamelcev.

Na primeru o letni rasti BDP-ja vidimo, da maksimalna in minimalna metoda nista primerni za večje podatkovje, medtem ko je Wardova metoda dala lepo razvrstitev tudi v takem primeru.

Mera različnosti pri metodi centroidov je

$$d(C_i, C_j) = \|t_i - t_j\|_2^2,$$

pri Wardovi metodi pa

$$d(C_i, C_j) = \frac{n_i n_j}{n_i + n_j} \|t_i - t_j\|_2^2.$$

Wardova metoda se od metode centroidov razlikuje po tem, da mero različnosti pomnožimo še z

$$\frac{n_i \cdot n_j}{n_i + n_j} = \frac{1}{\frac{1}{n_i} + \frac{1}{n_j}}. \tag{16}$$

Če se $n_i$ in $n_j$ večata, se imenovalec v (16) manjša in posledično se (16) veča. Zaradi tega Wardova metoda združuje manjše gruče ali gruče enake velikosti.

### 8.3. **Časovna in prostorska zahtevnost.**

Za splošen združevalen hierarhičen algoritem potrebujemo povezovalno matriko velikosti $n \times n$, kjer je $n$ število podatkov. Ker je simetrična, zavzame $\frac{1}{2}n^2$ prostora. V naslednjem koraku je matrika v najslabšem primeru (ko se združita samo dve gruči) velikosti $(n-1) \times (n-1)$. Torej je prostor, ki ga potrebujemo, da nadaljujemo z združevanjem gruč, sorazmeren s številom gruč, kar je $n - 1$. Sledi, da je skupna prostorska zahtevnost enaka $O(n^2)$.

Časovna zahtevnost za splošen združevalen hierarhičen algoritem je $O(n^2 \log(n))$. Za izpeljavo glej [7]. Algoritmi so časovno potratni, zato niso primerni za večja podatkovja. V naslednjem poglavju je predstavljena ideja algoritema, ki se lahko uporabi, kadar so podatkovja zelo velika.

## 9. Metoda CURE - Clustering Using REpresentatives

Najprej vpeljimo nekaj oznak o gruči $C_i$. Na začetku je vsaka podatkovna točka ena gruča. Na vsakem koraku združimo najbližji si gruči. Vsaka gruča $C_i$ ima naslednje informacije:

- $C_i.cent$ je centroid gruče $C_i$, ki ga izračunamo kot

$$C_i.cent = \frac{1}{n_i} \sum_{c_i \in C_i} c_i,$$

- v $C_i.pred$ shranjujemo nabor točk predstavnic,
- mera različnosti med gručo $C_i$ in $C_j$ se računa kot

$$d(C_i, C_j) = \min_{c_i \in C_i.rep, c_j \in C_j.rep} d(c_i, c_j).$$

Ideja je, da za vsako gručo izberemo $c$ dobro razpršenih točk. Dobro razpršene točke skrčimo s skrčitvenim faktorjem $\alpha$ proti centroidu gruče, kjer je $\alpha \in (0, 1)$. Predstavnice gruče so dobro razpršene točke po skrčitvi. Razdalja med dvema gručama je razdalja med najbližnjima predstavnicama iz vsake gruče. Združi tiste gruče z najkrajšo medsebojno razdaljo. S tem se odpravi pristop grupiranja, ki upošteva vse točke. S tem da krčimo nize, povečujemo oddaljenost od vsake gruče, znižamo vpliv osamelcev in ublažimo učinek veriženja. Algoritem CURE je primeren za veliko podatkovje. Če delamo z ogromnim podatkovjem, najprej reduciramo vhodne podatke in s tem prihranimo veliko časa ([5]).

9.1. **Algoritem za iskanje točk predstavnic.**

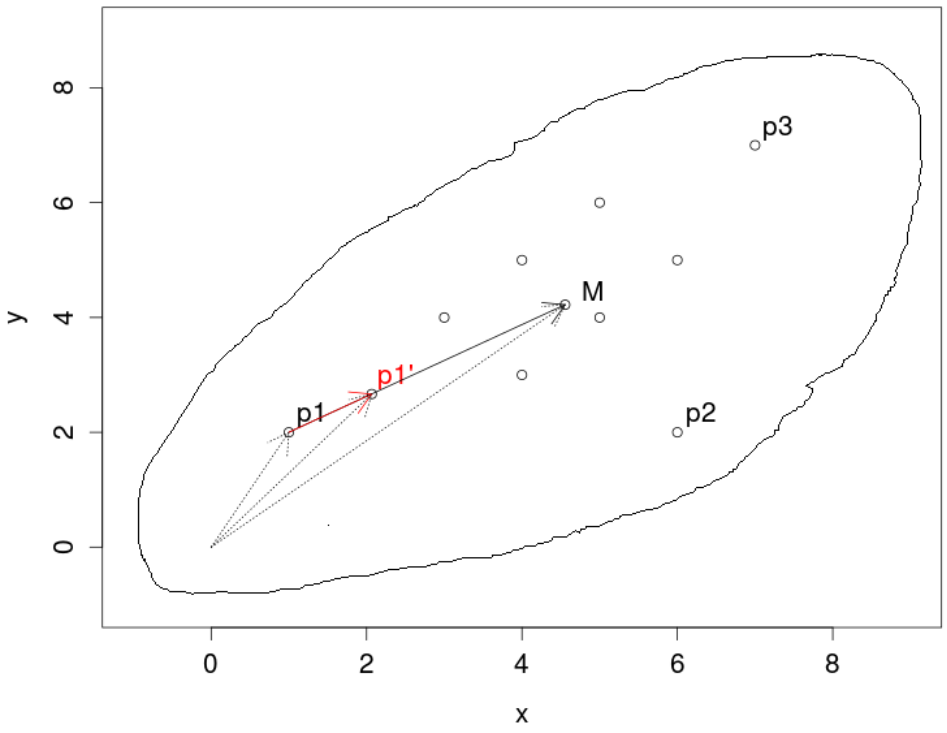
V algoritmu 3 je predstavljena psevdokoda, kako poiščemo dobro razpršene točke in kako poiščemo nove točke predstavnice za novonastalo gručo. Kot vhodne podatke vzamemo gruči, ki ju združimo, izberemo $c$ dobro razpršenih točk in skrčitveni faktor $\alpha$. Kot izhod pa dobimo točke predstavnice. Na enostavnem primeru si po-

**Algorithm 3** Algoritem za iskanje točk predstavnic

**procedure** ZDRUŽITEV($C_i, C_j, c, \alpha$)
  $C_{i\cup j} = C_i \cup C_j$ ▷ Gruča $C_{i\cup j}$ je unija vseh točk gruč $C_i$ in $C_j$.
  $C_{i\cup j}$.cent $= \frac{n_i t_i + n_j t_j}{n_i + n_j}$
  $A = \emptyset$ ▷ V množico $A$ shrani dobro razpršene točke gruče $C_{i\cup j}$.
  **for** $i = 1, \ldots, c$ **do** ▷ Določi dobro razpršene točke.
    maxRaz $= 0$
    **for** vsako točko $p$ v gruči $C_{i\cup j}$ **do**
      **if** i=1 **then**
        minRaz $= d(p, C_{i\cup j}$.cent) ▷ Za $d$ lahko vzamemo Evklidsko razdaljo ali razdaljo Manhattan.
      **else**
        minRaz $= \min\{d(p, q) : q \in A\}$
      **end if**
      **if** minRaz $\geq$ maxRaz **then**
        maxRaz $=$ minRaz
        maxTočka $= p$
      **end if**
    **end for**
    $A = A \cup \{\text{maxTočka}\}$
  **end for**
  $C_{i\cup j}$.pred $= \emptyset$ ▷ V množico $C_{i\cup j}$.pred shrani točke predstavnice gruče $C_{i\cup j}$.
  **for** vsako točko $p$ iz množice $A$ **do** ▷ Določi točke predstavnice.
    $C_{i\cup j}$.pred $= C_{i\cup j}$.pred $\cup \{p + \alpha(C_{i\cup j}\text{.cent} - p)\}$ ▷ To so točke iz $A$, ki se jih skrči s skrčitvenim faktorjem $\alpha$ proti centroidu.
  **end for**
  **return** $C_{i\cup j}$.pred
**end procedure**

glejmo kako algoritem išče dobro razpršene točke in kako jih skrči, da dobimo točke predstavnice (slika 23). Gruči se združita v novo gručo, ki ima 9 točk. Izberemo $c = 3$ in $\alpha = 0.3$. Centroid gruče označimo z $M$. Prva izračunana predstavnica je tista, ki je najbolj oddaljena od centroida gruče, označimo jo s $p1$. Druga izračunana predstavnica gruče $p2$ je tista, ki je najbolj oddaljena od prve izračunane predstavnice gruče. Za tretjo točko $p3$ poišče tako točko, da bo minimalna razdalja do ostalih točk predstavnic (v tem primeru ostalih dveh) največja. Dobro razpršene točke srkči proti centroidu gruče s faktorjem $\alpha$, ki je v tem primeru enak 0.3. Na sliki 23 je prikazano, kako se izračuna prva točka predstavnica, ki smo jo označili s $p1'$. Pri izračunu smo uporabili seštevanje in odštevanje vektorja. Torej $p1' = p + 0.3(M - p1)$.

Algoritem CURE je zelo uporaben na velikem podatkovju. Problem te metode je, da moramo vnaprej izbrati število predstavnic in skrčitveni faktor.




Slika 23. Iskanje točk predstavnic.

## 10. Zaključek

V diplomskem seminarju je opisanih več združevalnih metod. Vsaka na svoj način grupira podatke. Te metode so razumljive, enostavne za uporabo, skoraj vse so vgrajene v programske jezike, njihova pomankljivost pa je, da so razen Wardove metode in metode CURE primerne le za manjše podatkovje. Kljub temu te metode pogosto uporabljamo. Pomembno je, da na vsakem problemu poskušamo izvesti čim več metod, da primerjamo razvrstitve med seboj in se nato odločimo, katera razvrstitev je najprimernejša.

# Literatura


[1] A. Ferligoj, *Razvrščanje v skupine*, Zbirka Metodološki zvezki 4, Raziskovalni inštitut - založništvo, Ljubljana, 1989.

[2] P. Berkhin, *Survey of Clustering Data Minig Techniques, Grouping Multidimensional Data*, Springer Berlin Heidelberg, 2006.

[3] M. Kuchaki Rafsanjani, Z. Asghari Varzaneh, N. Emami Chukanlo, *A survey of hierarchical clustering algorithms*, TJMCS, **5** (2), (2012), 229–240.

[4] K. Košmelj, L. Breskvar Žaucer, *Metode za razvrščanje enot v skupine; osnove in primer*, Acta agriculturae Slovenica, **87** (2), (2006), 299–310.

[5] S. Guha, R. Rastogi, K. Shim, *CURE: An efficient clustering algorithm for large data bases*, Elsevier Science Ltd.,**26** (1), (2001), 35–58.

[6] V. Batagelj, *Note on ultrametric hierarchical clustering algorithms*, Psychometrika **46** (3), (1981) strani 351–352.

[7] M. J. Zaki, W. Meira, *Hierarchical Clustering*, 2014, [ogled 1. 9. 2015], dostopno na `http://www.cs.rpi.edu/~zaki/www-new/uploads/Dmcourse/Main/chap14.pdf`.

[8] E. Ferk, *Diplomsko delo, Delne urejenosti in hierarhično gručenje*, 2006, [ogled 24. 8. 2015], dostopno na `https://dk.um.si/IzpisGradiva.php?id=9968`.
verizenje:

[9] Single Linkage, [ogled 22. 08. 2015], dostopno na `http://www.molmine.com/magma/analysis/linkage.htm`.

[10] Hierarchical Clustering, [ogled 24. 08. 2015], dostopno na `http://www.solver.com/xlminer/help/hierarchical-clustering-intro`.

[11] V. Bahovec, *Klaster analiza*, [ogled 01. 09. 2015], dostopno na `http://web.efzg.hr/dok/sta/vbahovec//KLASTER_ANALIZA.pdf`.

[12] *Hierarchicical clustering*, [ogled 01. 12. 2014], dostopno na `http://en.wikipedia.org/wiki/Hierarchical_clustering`

[13] *GDP growth (annual %)*, [ogled 02. 09. 2015], dostopno na `http://data.worldbank.org/indicator/NY.GDP.MKTP.KD.ZG`