Metode segmentacije podataka, te primjena na optimizaciju digitalnog marketinga u podatkovnom oblaku

Braić, Drago

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Drago Braić

METODE SEGMENTACIJE PODATAKA, TE PRIMJENA NA OPTIMIZACIJU DIGITALNOG MARKETINGA U PODATKOVNOM OBLAKU

Diplomski rad

Voditelj rada:
prof. dr. sc. Siniša Slijepčević

Zagreb, ožujak 2021.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Zahvaljujem se mentoru, prof. dr. sc. Siniši Slijepčeviću, na svim smjernicama, savjetima i prenesenom znanju.
Zahvaljujem se svojoj obitelji na bezuvjetnoj podršci i pomoći pruženoj za vrijeme studiranja.
I svojim prijateljima, uz koje je bilo lakše proći kroz sve probleme i prepreke na putu do diplome.

Sadržaj

Uvod

U ovom radu obradit ćemo nekoliko metoda segmentacije podataka. Te metode za cilj ima podjelu skupa podataka u disjunktne podskupove (klastere), i to na način da se podaci unutar istog klastera razlikuju manje od podataka iz različitih klastera. Dakle, potrebno je na neki način strukturirati podatke, ili još ispravnije, pronaći prirodnu strukturu unutar samih podataka. Time dobijamo mogućnost reduciranja velikog skupa podataka, obzirom da podatke unutar istog klastera možemo opisati ili aproksimirati karakterističnim svojstvima tog klastera (prototipovima), kao i mogućnost analiziranja fenomena specifičnih za određeni podskup podataka.

Potreba za segmentiranjem vjerojatno je stara koliko i čovjek. Primjerice segmentacija plodova na otrovne i neotrovne, živih bića na ptice, ribe, sisavce, itd., ljudi po spolu, bila je nužna za razvoj čovjeka. Danas, prateći razvoj visoke tehnologije, metode segmentiranja nam mogu na temelju jako sitnih detalja pomoći u problemima s kojima se susrećemo svakodnevno. Primjene su razne, od sekvenciranja DNK, CT dijagnostike, segmentacije slika, pa sve do segmentacije određenih ljudskih populacija na temelju njihovog ponašanja ili demografskih karakteristika.

Čest primjer korištenja ovih metoda je analiza klijenata. Njihova podjela u homogene klastere, grupirane oko nekih zajedničkih interesa, svojstava i sličnih koncepata može pomoći u boljem shvaćanju istih. Na taj način moguće je olakšati razne procese u interakciji s klijentima, promatrati fenomene karakteristične za određenu grupu, izabirati reprezentativne uzorke, te jednostavno pridružiti nove klijente nekoj od već postojećih grupa.

U prvom poglavlju dajemo formalan zapis problema i uvodimo pojam mjere razlikovanja. U drugom poglavlju detaljnije opisujemo centroidne metode segmentacije. U trećem obrađujemo najpoznatiju metodu, *k-means* algoritam, kao tipičnog predstavnika centroidnih metoda segmentacije. U četvrtom poglavlju dajemo kratak pregled ostalih odabranih metoda, a u petom formalizaciju spektralnih metoda segmentacije. U šestom poglavlju primjenjujemo segmentacijske metode na skupu klijenata jedne banke.

Poglavlje 1

Segmentacija podataka

Pristup segmentaciji podataka ovisi o krajnjim ishodima koje želimo postići. Stoga je potrebno zadati jasan kriterij kojeg ćemo koristiti u procesu.

Kriterije možemo podijeliti na dvije velike grupe:

(i) Homogeni

- minimalne razlike među podacima unutar istog klastera
- mogućnost opisivanja podataka unutar klastera pomoću vjerojatnosnih distribucija
- reprezentativnost klastera pomoću njegovog prototipa
- mogućnost opisivanja klastera pomoću malog broja komponenti
- nezavisnot komponenti unutar klastera

(ii) Separacijski

- maksimalne razlike među klasterima
- stabilnost klastera obzirom na opetovano segmentiranje
- minimalan broj klastera
- slične veličine klastera
- pozitivan odgovor klastera na komponente koje nisu korištene prilikom segmentacije

Osim zadavanja kriterija, potrebno je i odrediti zadovoljavajuću razinu istog. Jedna od mogućnosti je i korištenje kombinacije kriterija. Tada je potrebno uzeti u obzir da oni do neke razine mogu biti suprostavljeni te će biti potrebno postizanje određenih kompromisa.

Kao što smo vidjeli, pojam segmentiranja teško je precizno definirati. To za posljedicu ima postojanje mnogobrojnih algoritama segmentacije sa zajedničkim nazivnikom, grupiranjem podataka.

1.1 Formalan zapis problema

Pretpostavimo da je zadan m-dimenzionalan skup podataka $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$, pri čemu je $m \in \mathbb{N}$ i svaki podatak ima $n \in \mathbb{N}$ komponenti (eng. *features*), pa ga možemo zapisat vektorski u obliku $\mathbf{x}_i = (x_{i1}, ..., x_{in})^T, \forall i = 1, ..., m$.

Sada na jednostavan način možemo konstruirat matrični prikaz podataka, i to na sljedeći način:

$$X = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x_m} \end{bmatrix}^T = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{1.1}$$

U ovom obliku prikaza podataka svaki stupac predstavlja jednu od n komponenti, a svaki redak jedan on m podataka.

Komponente mogu poprimati kontinuirane, kategorijske ili ordinalne vrijednosti. U svakom slučaju, većina teoretskih rezultata bazira se na pretpostavci da svaka komponenta poprima vrijednosti iz skupa realnih brojeva. Stoga ćemo u i ovom radu (osim ako nije navedeno drugačije) koristiti tu pretpostavku, dakle $\mathbf{x}_i \in \mathbb{R}^n, \forall i = 1, ..., m$.

Definicija 1.1.1. *Segmentacija skupa podataka $\mathcal{X}$ je skup $\mathcal{S} = \{C_1, ..., C_k\}$, $k \in \mathbb{N}$, takav da je $C_j \subseteq \mathcal{X}, \forall j = 1, ..., k$ i vrijede sljedeća svojstva:*

(S1) $C_j \neq \emptyset, \forall j = 1, ..., k$

(S2) $\cup_{j=1}^{k} C_j = \mathcal{X}$

(S3) $C_{j'} \cap C_{j''} = \emptyset, \forall j', j'' = 1, ..., k$ *t.d. je* $j' \neq j''$

Element $C_j \in \mathcal{S}$ nazivamo klasterom (eng. cluster*), a njegov kardinalitet označavamo sa $m_j = |C_j|$.*

Dakle, segmentaciju skupa podataka možemo promatrati kao njegovu particiju. Svojstvo (S1) nam govori da svaki klaster sadrži barem jedan podatak, (S2) da svaki podatak pripada barem jednom od klastera i (S3) da su klasteri međusobno disjunktni skupovi.

U slučaju kada je $k = m$, zbog prethodno navedenih svojstava, radi se o trivijalnoj segmentaciji, onoj u kojoj svaki podatak možemo poistovjetiti s jednim klasterom. Ovaj slučaj nećemo uzimati u obzir, štoviše, cilj je postići $k << m$.

Definicija 1.1.2. *Za dani skup podataka $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$ i njegovu segmentaciju $\mathcal{S} = \{C_1, ..., C_k\}$ definiramo incidencijsku matricu segmentacije $U = [u_{ij}] \in \mathbb{R}^{m \times k}$ sa*

$$u_{ij} = \begin{cases} 1 & \text{, ako je } \mathbf{x}_i \in C_j \\ 0 & \text{, inače} \end{cases} \tag{1.2}$$

Incidencijska matrica segmentacije ima sljedeća svojstva:

(i) $\sum_{j=1}^{k} u_{ij} = 1, \forall i = 1, ..., m$, tj. svaki podatak pripada točno jednom klasteru.

(ii) $1 \leqslant \sum_{i=1}^{m} u_{ij} < m, \forall j = 1, ..., k$, tj. svaki klaster sadrži barem jedan podatak, ali ih ne sadrži sve.

Općenito, moguće je umjesto definicije 1.1.2 matricu U definirati pomoću prethodno navedenih svojstava, uz dodatan uvjet da je $u_{ij} \in [0, 1], \forall i = 1, ..., m, \forall j = 1, ..., k$. Tako definirana matrica U odgovara metodama slabih ili *fuzzy* segmentacija. Međutim, mi ćemo se nadalje koncentrirati na prvotnu definiciju po kojoj vrijedi $u_{ij} \in \{0, 1\}, \forall i = 1, ..., m, \forall j = 1, ..., k$, tj. na metode jakih segmentacija.

Označimo sada sa $\mathcal{U}_{m\times k}$ skup svih $m \times k$ dimenzionalnih matrica sa gore navedenim svojstvima. Tada je kardinalnost tog skupa jednaka Stirlingovom broju druge vrste (vidi [1]), tj. $\vartheta(m, k) := card(\mathcal{U}_{m\times k}) = S(m, k) = \frac{1}{k!}\sum_{j=1}^{k}(-1)^{k-j}\binom{k}{j}j^m$. Naime, ovako definiran skup možemo interpretirati kao skup svih mogućih segmentacija m podataka u k klastera. Primjerice, 3 podatka u 2 klastera možemo segmentirati na 3 različita načina, dok već kada imamo 100 podataka koje želimo segmentirati u 5 klastera, broj različitih načina na koje to možemo postići reda je veličine 10^{68}. Općenito, kardinalitet gore navedenog skupa možemo aproksimirati sa $\frac{k^n}{k!}$, tj. redom veličine $O(k^n)$.

Segmentacija podataka jedna je od metoda nenadziranog učenja (eng. *unsupervised learning*). Nemamo dostupnu informaciju o tome koji podatak pripada kojem klasteru, čak često ni informaciju koliko bi različitih klastera trebalo postojati.

Prisutnost određene strukture u podacima očituje se postojanjem separacijskih područja. Cilj je pronaći segmentaciju koja će ih što bolje oslikavati. Podaci koji pripadaju istom klasteru trebali bi biti sličniji međusobno nego bilo koja dva podatka odabrana iz različitih klastera. Drugim riječima, označimo li sa $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ mjeru sličnosti među podacima, želimo postići sljedeće:

$$s(\mathbf{x}', \mathbf{x}'') > s(\mathbf{y}', \mathbf{y}''), \forall \mathbf{x}', \mathbf{x}'' \in C_i, \forall \mathbf{y}' \in C_{j'}, \forall \mathbf{y}'' \in C_{j''}$$
$$\text{pri čemu su } i, j', j'' = 1, ..., k \text{ t.d. je } j' \neq j''$$

1.2 Mjere sličnosti i razlikovanja

U prethodnom potpoglavlju prešutno smo uveli pojam mjere sličnosti između podataka. Naime, radi se o preslikavanju $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ za koje nemamo službenu definiciju, ali intuitivno je jasno da se radi o nekom smislu inverza pojmu metrike. Jednom kada imamo zadanu mjeru sličnosti, moguće je na razne načine zadati i njoj dualan pojam, mjeru razlikovanja. Vrijedi i obrat, a u praksi se češće zadaje mjera razlikovanja i to u obliku metrike.

Definicija 1.2.1. *Za funkciju* $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *kažemo da je metrika ili udaljenost na skupu* $\mathcal{X}$ *ako vrijede sljedeća svojstva:*

(M1) $d(\mathbf{x}, \mathbf{y}) \geqslant 0, \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$

(M2) $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}, \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$

(M3) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}), \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$

(M4) $d(\mathbf{x}, \mathbf{y}) \leqslant d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y}), \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}$

Uvjete (M1) i (M2) nazivamo pozitivnom definitnošću, uvjet (M3) simetrijom te uvjet (M4) nejednakošću trokuta.

Pretpostavimo da je zadana metrika d na skupu $\mathcal{X}$. Kako je skup $\mathcal{X}$ u našem slučaju konačan, moguće je pronaći vrijednost $d_{max} = \max\{d(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in \mathcal{X}\}$, pa možemo konstruirati pripadnu mjeru sličnosti pravilom $s(\mathbf{x}, \mathbf{y}) = d_{max} - d(\mathbf{x}, \mathbf{y}), \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$.

Ovo je samo jedan od načina kako iz zadane metrike možemo definirati mjeru sličnosti. Navedimo neke od ostalih transformacija metrike u mjeru sličnosti. Dakle, uz pretpostavku da je zadana metrika $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ moguće je definirati mjeru sličnosti $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ na neki od sljedećih načina:

(a) $s(\mathbf{x}, \mathbf{y}) = 1 - d(\mathbf{x}, \mathbf{y})/d_{max}$

(b) $s(\mathbf{x}, \mathbf{y}) = exp\left[-d^2(\mathbf{x}, \mathbf{y})/\sigma^2\right], \sigma > 0$

(c) $s(\mathbf{x}, \mathbf{y}) = 1/\left[d(\mathbf{x}, \mathbf{y}) + \epsilon\right], \epsilon > 0$

Chen, Ma i Zeng u svom radu iz 2009. [39] uvode pojam metrike sličnosti.

Definicija 1.2.2. *Za funkciju* $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *kažemo da je metrika sličnosti na skupu* $\mathcal{X}$ *ako vrijede sljedeća svojstva:*

(MS1) $s(\mathbf{x}, \mathbf{y}) = s(\mathbf{y}, \mathbf{x}), \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$

(MS2) $s(\mathbf{x}, \mathbf{x}) \geqslant 0, \forall \mathbf{x} \in \mathcal{X}$

(MS3) $s(\mathbf{x}, \mathbf{x}) \geqslant s(\mathbf{x}, \mathbf{y}), \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$

(MS4) $s(\mathbf{x}, \mathbf{y}) + s(\mathbf{y} + \mathbf{z}) \leqslant s(\mathbf{x}, \mathbf{z}) + s(\mathbf{y}, \mathbf{y}), \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}$

(MS5) $s(\mathbf{x}, \mathbf{x}) = s(\mathbf{y}, \mathbf{y}) = s(\mathbf{x}, \mathbf{y}) \Leftrightarrow \mathbf{x} = \mathbf{y}, \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$

Općenito, ako je $f : \mathbb{R} \to \mathbb{R}$ konveksna, monotono padajuća funkcija takva da je $f(0) > 0$ i $\lim_{\epsilon\to\infty} f(\epsilon) = a \geqslant 0$, tada je sa $s(\mathbf{x}, \mathbf{y}) = f(d(\mathbf{x}, \mathbf{y})), \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$ dobro definirana metrika sličnosti [40, str. 17-18].

Jednom kada smo definirali mjeru sličnosti između podataka, možemo konstruirati matricu sličnosti $S = [s_{ij}] \in \mathbb{R}^{m\times m}$ s elementima $s_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$. Na analogan način moguće je konstruirati i matricu razlikovanja $D = [d_{ij}] \in \mathbb{R}^{m\times m}$.

Usporedba kontinuiranih vrijednosti

Kada su podaci takvi da im sve komponente poprimaju kontinuirane vrijednosti, tj. uz pretpostavku da je $\mathcal{X} \subseteq \mathbb{R}^n$, najčešće se koristi Euklidska metrika.

Definicija 1.2.3. *Metriku $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ definiranu sa*

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1.3}$$

nazivamo Euklidskom metrikom.

Napomena 1.2.4. *Moguće je generalizirati konstrukciju prethodne metrike. Naime, neka je $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ pozitivno definitna matrica. Tada je funkcija $d_W : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ definirana sa*

$$d_W(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_W = \sqrt{(\mathbf{x} - \mathbf{y})^T W (\mathbf{x} - \mathbf{y})} \tag{1.4}$$

metrika na skupu $\mathcal{X}$.

Ako je W jedinična matrica, tada formulom (1.4) definiramo Euklidsku metriku. S druge strane, ako je W dijagonalna matrica s elementima

$$w_{ij} = \begin{cases} \omega_i & \text{, ako je } i = j \\ 0 & \text{, inače} \end{cases}$$

tada je formulom (1.4) definirana težinska Euklidska metrika

$$d_W(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} \omega_i (x_i - y_i)^2} \tag{1.5}$$

Metrika Minkowskog

Definicija 1.2.5. *Neka je $p \in [1, +\infty\rangle$. Metriku $d_p : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ definiranu sa*

$$d_p(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p = \left[\sum_{i=1}^{n} |x_i - y_i|^p \right]^{1/p} \tag{1.6}$$

nazivamo metrikom Minkowskog.

Promotrimo neke posebne slučajeve ovako definirane metrike.

(a) (Manhattan metrika) Za $p = 1$ vrijedi

$$d_1(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1 = \sum_{i=1}^{n} |x_i - y_i| \tag{1.7}$$

(b) (Euklidska metrika) U slučaju kada je $p = 2$ radi se o Euklidskoj metrici i umjesto $\|\cdot\|_2$ skraćeno pišemo $\|\cdot\|$.

(c) (Čebiševljeva metrika) Za $p = \infty$ možemo definirati

$$d_\infty(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_\infty = \max_{i=1,\ldots n} |x_i - y_i| \tag{1.8}$$

Mahalanobisova metrika

Metriku definiranu u (1.4) koristimo kada je zadovoljena pretpostavka o nekoreliranosti među komponentama. Kada ova pretpostavka nije zadovoljena, koristimo Mahalanobisovu metriku.

Definicija 1.2.6. *Metriku $d_\Sigma : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ definiranu sa*

$$d_\Sigma(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})} \tag{1.9}$$

pri čemu je Σ kovarijacijska matrica uzorka $\mathcal{X}$, nazivamo Mahalanobisovom metrikom.

Napomena 1.2.7. *Kod definiranja Mahalanobisove metrike dodatno smo pretpostavili da je $\mathcal{X}$ m-dimenzionalan uzorak n-dimenzionalnog slučajnog vektora. Naime, tada možemo definirati:*

(i) Uzoračko očekivanje sa

$$\bar{\mu} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \tag{1.10}$$

(ii) Kovarijacijsku matricu uzorka sa

$$\Sigma = \frac{1}{m-1}\sum_{i=1}^{m}(\mathbf{x}_i - \bar{\mu})(\mathbf{x}_i - \bar{\mu})^T \tag{1.11}$$

Dakle, radi se o varijanti generalizirane metrike (1.4) u kojoj je matrica W inverz kovarijacijske matrice uzorka.

Kada su komponente u podacima međusobno nezavisne, kovarijacijska matrica je dijagonalna matrica čija je vrijednost na mjestu (i, i) jednaka varijanci i-te komponente, tj.

$$d_\Sigma(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{n}\left(\frac{x_{il} - x_{jl}}{\sigma_l}\right)^2} \tag{1.12}$$

pri čemu je

$$\sigma_l = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(x_{il} - \bar{\mu}_l)^2}, \forall l = 1, ..., n \tag{1.13}$$

Pearsonova korelacija

Navedimo primjer nemetričke mjere razlikovanja.

Definicija 1.2.8. *Mjeru razlikovanja* $d_{cor} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *definiranu sa*

$$d_{cor}(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{\sum_{l=1}^{n}(x_{il} - \bar{\mathbf{x}}_i)(x_{jl} - \bar{\mathbf{x}}_j)}{\sqrt{\sum_{l=1}^{n}(x_{il} - \bar{\mathbf{x}}_i)^2 \sum_{l=1}^{n}(x_{jl} - \bar{\mathbf{x}}_j)^2}} \tag{1.14}$$

pri čemu je

$$\bar{\mathbf{x}}_i = \frac{1}{n}\sum_{l=1}^{n} x_{il}, \forall i = 1, ..., m \tag{1.15}$$

nazivamo mjerom Pearsonove korelacije.

U tom slučaju, vektore $\mathbf{x}_i$ i $\mathbf{x}_j$ promatramo kao n-dimenzionalne uzorke slučajnih varijabli, te pomoću Pearsonovog koeficijenta korelacije mjerimo razlikovanje između njih. Poznato je da Pearsonov koeficijent korelacije poprima vrijednosti iz skupa $[-1, 1]$, pri čemu vrijednosti blizu 1 označavaju jaku pozitivnu linearnu vezu između varijabli, one blizu -1 jaku negativnu linearnu vezu, a vrijednosti oko 0 nepostojanje istih.

Usporedba kategorijskih vrijednosti

Dosada smo opisali načine na koje možemo usporediti podatke čije sve komponente poprimaju kontinuirane vrijednosti. Na trenutak pretpostavimo da komponente u podacima mogu poprimati i kategorijske vrijednosti.

U svrhu uspješne usporedbe takvih podataka, Gower u svom radu iz 1971. [22] definira (Gowerov) koeficijent.

Definicija 1.2.9. *Neka je $\mathcal{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ skup podataka čije komponente mogu poprimati kategorijske i/ili kontinuirane vrijednosti.*

(i) Koeficijent definiran sa

$$\delta(i,j,l) = \begin{cases} 1 & \text{, ako je } l\text{-ta komponenta kategorijska i } x_{il} = x_{jl} \\ 0 & \text{, ako je } l\text{-ta komponenta kategorijska i } x_{il} \neq x_{jl} \\ \frac{|x_{il}-x_{jl}|}{x_l^{max}-x_l^{min}} & \text{, inače} \end{cases} \tag{1.16}$$

pri čemu su

$$i, j = 1, ..., m,\ l = 1, ..., n,\ x_l^{max} = \max\{x_{il} : i = 1, ..., m\} \text{ i } x_l^{min} = \min\{x_{il} : i = 1, ..., m\}$$

nazivamo Gowerovim koeficijentom.

(ii) Funkciju $d_G : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ definiranu sa

$$d_G(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{n} \delta(i,j,l) \tag{1.17}$$

nazivamo Gowerovom mjerom razlikovanja.

Standardizacija podataka

U praksi se često susrećemo sa podacima čije komponente nisu prikazane u istim mjernim jedinicama. Štoviše, često se ne radi o istom tipu podatka. Primjerice, mjesečni dohodak i broj djece. Kod prve komponente razlike za pojedine podatke mogu ići od nekoliko stotina do nekoliko tisuća jedinica, dok se za drugu komponentu radi o značajno manjim razlikama, od svega nekoliko jedinica. Kako bismo uspješno usporedili takve podatke potrebno ih je skalirati. U tu svrhu uvodimo pojam standardizacije podataka.

Transformacijom

$$\tilde{x}_{ij} \leftarrow \frac{x_{ij} - \bar{\mu}_j}{\sigma_j} \tag{1.18}$$

dobijamo uzoračko očekivanje svake od komponenti jednako 0 te uzoračku standardnu devijaciju jednaku 1. Na taj način riješili smo problem podataka u kojima komponente međusobno nisu usporedive, svodeći ih na istu skalu.

1.3 Kriteriji segmentiranja

U radu iz 1997. Hansen i Jaumard [35] daju dobar pregled mogućih kriterija segmentacije i dijele ih u dvije velike grupe. Homogenost klastera orijentirana je na postizanje visoke razine sličnosti među podacima unutar istog klastera, dok je s druge strane separabilnost klastera orijentirana na postizanje visoke razine razlikovanja među podacima iz različitih klastera. Idealne segmentacije su one koje zadovoljiavaju oba kriterija. Neka je dana segmentacija $\mathcal{S} = \{C_1, ..., C_k\}$ skupa $\mathcal{X}$ i matrica razlikovanja $D = [d_{ij}]_{m\times m}$.

Homogenost klastera $C_l \in \mathcal{S}$ procjenjujemo koristeći jedan od sljedećih kriterija:

$$h_1(C_l) = \sum_{\mathbf{x}_i,\mathbf{x}_j\in C_l} d_{ij}$$
$$h_2(C_l) = \max_{\mathbf{x}_i,\mathbf{x}_j\in C_l} d_{ij}$$
$$h_3(C_l) = \min_{\mathbf{x}_i\in C_l} \max_{\mathbf{x}_j\in C_l} d_{ij}$$
$$h_4(C_l) = \min_{\mathbf{x}_i\in C_l} \sum_{\mathbf{x}_j\in C_l} d_{ij}$$

S druge strane, njegovu separabilnost koristeći jedan od sljedećih kriterija:

$$s_1(C_l) = \sum_{\mathbf{x}_i\in C_l} \sum_{\mathbf{x}_j\notin C_l} d_{ij}$$
$$s_2(C_l) = \min_{\mathbf{x}_i\in C_l,\mathbf{x}_j\notin C_l} d_{ij}$$

Nakon što odaberemo jedan od prethodno navedenih kriterija (homogeni ili separacijski), i označimo ga sa m_l, $\forall l = 1, ..., k$, možemo definirati kriterij segmentacije na jedan od sljedećih načina:

$$J_1(C_1, ..., C_k) = \frac{1}{k} \sum_{i=1}^{k} m_i$$
$$J_2(C_1, ..., C_k) = \max_{i=1,...,k} m_i$$
$$J_3(C_1, ..., C_k) = \min_{i=1,...,k} m_i$$

U slučaju homogenosti cilj je postići što niže vrijednosti navedenih kriterija, dok u slučaju separabilnost vrijedi obratno.

Poglavlje 2

Centroidne metode segmentacije

U ovom poglavlju obrađujemo centroidne metode segmentacije. Karakteristika ovih metoda reprezentativnost je klastera pomoću njihovih centara, tj. karakterističnih predstavnika.

Kao što smo ranije naveli, koristimo pretpostavku da je $\mathcal{X}$ podskup n-dimenzionalnog Euklidskog prostora $\mathbb{R}^n$.

Definicija 2.0.1. *Centar skupa $\mathcal{X}$ definiramo sa*

$$\bar{\mu} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \tag{2.1}$$

Definicija 2.0.2. *Za danu segmentaciju $\mathcal{S} = \{C_1, ..., C_k\}$ skupa $\mathcal{X}$ definiramo centar j-tog klastera $C_j \in \mathcal{S}$ sa*

$$\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i = \frac{1}{\sum_{i=1}^{m} u_{ij}} \sum_{i=1}^{m} u_{ij}\mathbf{x}_i = \frac{1}{m_j} \sum_{i=1}^{m} u_{ij}\mathbf{x}_i \tag{2.2}$$

te matricu centara klastera $M = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_k \end{bmatrix}^T$.

Centri klastera primjer su prototipova klastera. Naime, njih promatramo kao karakteristične objekte svakog klastera, a u centroidnim metodama igraju važnu ulogu u odabiru prigodne segmentacije.

Definicija 2.0.3. *Za danu segmentaciju $\mathcal{S} = \{C_1, ..., C_k\}$ skupa $\mathcal{X}$ definiramo:*

(i) unutar-klastersku kovarijacijsku matricu

$$W = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij}(\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T \tag{2.3}$$

(ii) među-klastersku kovarijacijsku matricu

$$B = \sum_{j=1}^{k} \left(\sum_{i=1}^{m} u_{ij} \right) (\mu_j - \bar{\mu})(\mu_j - \bar{\mu})^T \tag{2.4}$$

Ovime smo uveli dekompoziciji kovarijacijske matrice uzroka

$$T = (m-1)\Sigma = \sum_{i=1}^{m} (\mathbf{x}_i - \bar{\mu})(\mathbf{x}_i - \bar{\mu})^T \tag{2.5}$$

na sumu $T = W + B$.

U slučaju kada je $n = 1$ radi se o podijeli sume kvadratnih pogrešaka na unutar-klasterske i među-klasterske. Prirodno bismo željeli minimiziratu one unutar-klasterske i/ili maksimizirati među-klasterske. Međutim, u slučaju kada je $n > 1$ nije trivijano jasno koji bi kriterij trebali koristiti. Neke od mogućnosti su sljedeće:

(a) (Minimizacija traga matrice W)
Ovaj kriterij prvi je uveo Ward u svom radu iz 1963. [25]. Radi se o prirodnom proširenju prethodno navedenog kriterija minimiziranja unutar-klasterskih kvadratnih pogrešaka. Ili drugim riječima, minimizaciji kriterija homogenosti $h_1(C_j)/m_j$.

(b) (Minimizacija determinante matrice W)
U multivarijantnoj analizi varijance jedan od testova korištenih pri analizi razlikovanja očekivanja među grupama baziran je na odnosu ukupne varijabilnosti i varijabilnosti unutar grupa. Visoke vrijednosti izraza $\det(T)/\det(W)$ indikator su postojanja tih razlika. Na temelju toga, Friedman i Rubin u radu iz 1967. [20] predlažu maksimizaciju izraza $\det(T)/\det(W)$, ili obzirom da se matrica T ne mijenja, minimizaciju izraza $\det(W)$ kao mogući kriterij segmentacije.

(c) (Maksimizacija traga matrice BW^{-1})
U istom radu, autori predlažu i maksimizaciju izraza $tr(BW^{-1})$. Ponovno se radi o izvedenici postupaka korištenih pri multivarijantnoj analizi varijance.

2.1 Minimizacija traga matrice W

Minimizacija traga matrice W najčešće je korišten kriterij u praksi. Iako se na prvu čini kako odgovor leži u njegovoj jednostavnosti, radi se o iznenađujuće visokom stupnju generaliziranosti.

Dakle, cilj ovog kriterija minimizacija je funkcije

$$J_1(C_1, ..., C_k) = \sum_{j=1}^{k} \sum_{i=1}^{m} u_{ij} \|\mathbf{x}_i - \mu_j\|^2$$
$$= \sum_{j=1}^{k} \frac{1}{m_j} \sum_{\mathbf{x}_i, x_l \in C_j} \|\mathbf{x}_i - \mathbf{x}_l\|^2$$

Podsjetimo se, ova funkcija može poprimati $\vartheta(m,k) = S(m,k) \approx \frac{k^n}{k!}$ različitih vrijednosti. Zbog toga, određivanje njenog globalnog minimuma gotovo je nemoguće za velike skupove podataka (vidi [11]).

Koristeći formulaciju problema optimizacije, cilj je

$$\min_{u_{ij} \in \{0,1\}} \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} \left\| \mathbf{x}_i - \frac{\sum_{l=1}^{m} u_{lj} \mathbf{x}_l}{\sum_{l=1}^{m} u_{lj}} \right\|^2 \tag{2.6}$$

$$\text{uz } 1 \leqslant \sum_{i=1}^{m} u_{ij} < m, \forall j = 1, ..., k \tag{2.7}$$

$$\sum_{j=1}^{k} u_{ij} = 1, \forall i = 1, ..., m \tag{2.8}$$

Neka je $F = [f_{ij}]_{m \times m}$ matrica s elementima

$$f_{ij} = \begin{cases} \frac{1}{m_l} & \text{, ako je } \{\mathbf{x}_i, \mathbf{x}_j\} \subseteq C_l \\ 0 & \text{, inače} \end{cases} \tag{2.9}$$

Ako podatke u $\mathcal{X}$ numeriramo tako da prvih m_1 podataka pripada klasteru C_1, sljedećih m_2 klasteru C_2, itd., tada je F blok-dijagonalna matrica, $F = \text{diag}(F_1, ..., F_k)$. Svaki blok F_j je m_j-dimenzionalna kvadratna matrica sa elemetnima $1/m_j$, tj. $F_j = (1/m_j)\mathbf{e}\mathbf{e}^T, \forall j = 1, ..., k$.

Lema 2.1.1. *Matrica F definirana u (2.9) ima sljedeća svojstva:*

(i) F *je nenegativna simetrična matrica, tj.* $f_{ij} = f_{ji} \geqslant 0, \forall i, j = 1, ..., m$

(ii) F *je dvostruko stohastična matrica, tj.* $F\mathbf{e} = F^T\mathbf{e} = \mathbf{e}$

(iii) F *je idempotentna, tj.* $FF = F$

(iv) $tr(F) = k$

(v) $\sigma(F) = \{0, 1\}$*, a kratnost svojstvene vrjednost* 1 *jednaka je* k

Dokaz. Svojstva (*i*)-(*iv*) vrijede po definiciji. Dokažimo svojstvo (*v*).
Obzirom da je svaki blok matrice F oblika $F_j = (1/m_j)\mathbf{e}\mathbf{e}^T$, vrijedi da je točno jedna svojstvena vrijednosti ovog bloka jednaka 1, a preostale su jednake 0. Spektar matrice F dobijemo koristeći jednakost $\sigma(F) = \cup_{j=1}^{k}\sigma(F_j)$. Dakle, F ima točno k svojstvenih vrijednosti jednakih 1, a preostale su jednake 0. □

Sada možemo definirati matricu $Q = FX$ čiji i-ti redak predstavlja centar klastera kojem i-ti podatak pripada. Funkciju cilja tada možemo zapisati u matričnoj formi

$$\begin{aligned} J_1(C_1, ..., C_k) &= \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \mu_j\|^2 \\ &= tr\left((X - Q)^T (X - Q)\right) \\ &= tr(X^T X + Q^T Q - 2X^T Q) \end{aligned}$$

Koristeći svojstva aditivnosti i komutativnosti traga matrice, kao i simetričnost i idempotentnost matrice F gornji izraz jednak je

$$\begin{aligned} J_1(C_1, ..., C_k) &= tr(X^T X + X^T F^T F X - 2X^T F X) \\ &= tr(X^T X + X^T F X - 2X^T F X) \\ &= tr(X^T X - X^T F X) \\ &= tr(X^T X - X^T X F) \end{aligned}$$

Neka je

$$K = X^T X$$

Tada je K simetrična, nenegativna matrica. Izraz $tr(X^T X F) = tr(KF) = \sum_{ij} k_{ij} f_{ij}$ linearna je kombinacija elemenata matrice K. Zaključno, zadaća segmentacije u Euklidskom prostoru svodi sa na ili minimizaciju kriterija

$$J_1(C_1, ..., C_k) = tr\left(K(\mathbb{I} - F)\right) \tag{2.10}$$

ili ekvivlanento, ako ignoriramo konstantu K, maksimizaciju kriterija

$$J_1'(C_1, ..., C_k) = tr(KF) \tag{2.11}$$

U drugom slučaju radi se o problemu optimizacije

$$\max_{F \in \mathbb{R}^{m \times m}} tr(KF) \tag{2.12}$$

$$\text{uz } F \geqslant 0, F^T = F, F\mathbf{e} = \mathbf{e} \tag{2.13}$$

$$F^2 = F, tr(F) = k \tag{2.14}$$

2.2 Aproksimacija matrice podataka

Definicija 2.2.1. *Matričnu normu* $\|\cdot\|_F : \mathbb{R}^{m\times n} \to \mathbb{R}$ *definiranu sa*

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{tr(A^TA)} \tag{2.15}$$

nazivamo Forbeniusovom normom.

Funkciju cilja (2.10) moguće je zapisati u sljedećem obliku

$$J_1(C_1, ..., C_k) = \|X - UM\|_F^2 \tag{2.16}$$

pri čemu je $M \in \mathbb{R}^{k\times n}$ matrica centara klastera, a matrica $U \in \mathbb{R}^{m\times k}$ incidencijska matrica segmentacije.

Minimizacija ovako definiranog indikatora daje nam drugu perspektivu zadaće segmentacije. Naime, tražimo dobru aproksimaciju skupa podataka preko produkta dviju matrica, U i M. Ako je dan uvjet $u_{ij} \in \{0, 1\}$ tada se rješenje ovog problema može postići sljedećim koracima:

Algoritam 1

1: Ako je $\hat{M} = (\hat{\mu}_1, ..., \hat{\mu}_k)^T$ trenutna aproksimacija matrice M, onda matricu $\hat{U}$ konstruiramo sa

$$\hat{u}_{ij} = \begin{cases} 1 & \text{, ako je } j = \arg\min_{1\leqslant t\leqslant k} \|\mathbf{x}_i - \hat{\mu}_t\|_F^2 \\ 0 & \text{, inače} \end{cases} \tag{2.17}$$

2: Ako je $\hat{U}$ trenutna aproksimacija matrice U, tada se određivanje matrice $\hat{M}$ minimiziranjem indikatora J_1 svodi na klasični problem regresije. Iz uvjeta $\partial J_1/\partial\hat{M} = \hat{U}^T(\hat{U}\hat{M} - X) = 0$ dobijemo

$$\hat{M} = (\hat{U}^T\hat{U})^{-1}\hat{U}^TX \tag{2.18}$$

3: Ponavljaj korake (1) i (2) do postizanja željene razine stabilnosti matrice $\hat{U}$ ili do konačnog broja ponavljanja.

Ovaj algoritam zahtjeva inicijalnu verziju jedne od matrica $\hat{U}$ ili $\hat{M}$.

2.3 Iterativni algoritmi

Algoritmi particijskih metoda segmentacije u pravilu su iterativni. U svom radu iz 2003. Steinley [10] je pokazao da oni najčešće konvergiraju lokalnom optimumu. Primjer je prethodno opisan algoritam. Slabost mu leži u nužnosti računanja izraza (2.18).

S druge strane, obzirom na specifičnu strukturu matrice U vrijedi:

(a) $\tilde{U} = \hat{U}^T\hat{U}$ je dijagonalna matrica s elementima

$$\tilde{u}_{jj} = \sum_{i=1}^{m} \hat{u}_{ij} = m_j, \forall j = 1, ..., k$$

Štoviše, matrica $\tilde{U}^{-1}$ je dijagonalna matrica, s elementima $\tilde{u}_{jj}^{-1} = 1/m_j$.

(b) Matrica $\tilde{M} = \hat{U}^T X$ dimenzije je $k \times n$, i njen i-ti redak suma je redaka matrice X koji odgovaraju i-tom klasteru. Dakle, $\tilde{U}^{-1}\tilde{M} = \hat{M}$.

Generalna forma iterativnih procedura opisana je u algoritmu (1). U sljedećoj varijanti uvodimo težišnu funkciju kojom opisujemo koliko pojedini podataka doprinosi u kalkulaciji centra klastera. Bit algoritma iterativna je modifikacija incidencijske matrice segmentacije. Najčešće se koristi jednostavno pravilo pridruživanja podatka klasteru čiji mu je centar najbliži

$$u_{ij} = \begin{cases} 1 & \text{, ako je } j = \arg\min_{1 \leqslant l \leqslant k} \|\mathbf{x}_j - \mu_l\| \\ 0 & \text{, inače} \end{cases} \tag{2.19}$$

Takvo pravilo opisujemo izrazom *the winner takes all*. Klasteri, u ovako definiranom algoritmu, nazivaju se Voronovljevim klasterima. Radi se o klasterima za koje je dovoljna informacija o njihovim prototipovima (centrima) $\mu_1, ..., \mu_k$. Tada ih možemo jednostavno odrediti sa

$$C_j = \{\mathbf{x} \in \mathbb{R}^n : j = \arg\min_{1 \leqslant l \leqslant k} \|\mathbf{x} - \mu_l\|\} \cap \mathcal{X} \tag{2.20}$$

Algoritam 2 Generalizirani Lloydov algoritam

Ulazni parametri: Skup podataka $\mathcal{X}$ i broj klastera k
Izlazni parametri: Prototipovi klastera $\{\mu_1, ..., \mu_k\}$

1: Inicijaliziraj centre klastera i težinsku funkciju $w : \mathcal{X} \to \mathbb{R}$.
2: Izračunaj incidencijsku matricu segmentacije U i vrijednosti težinske funkcije za sve podatke.
3: Modificiraj centre klastera formulom

$$\mu_j = \frac{\sum_{i=1}^{m} u_{ij} w(\mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{m} u_{ij} w(\mathbf{x}_i)}$$

4: Ponavljaj korake (2) i (3) do postizanja zadovoljavajuće razine stabilnosti matrice U.

Poglavlje 3

k-means algoritam

Najtipičniji predstavnik centoridnih metoda segmentacija je *k-means* algoritam. Ovaj naziv prvi put koristi James MacQueen 1967., iako se ideja prvi put pojavljuje 1956. u radu Huga Steinhausa. Bazu na kojoj se kasnije izgradio ovaj algoritma uvodi Stuart Lloyd 1957. Ipak, Lloyd svoj rad publicira kao članak tek 1982., a obzirom da je Edward W. Forgy 1965. objavio praktički istu metodu, ona se često naziva *Lloyd-Forgy* metodom.

Kao kriterij segmentacije *k-means* koristi minimizaciju traga matrice W. Funkcija cilja u tom slučaju zadana je sa

$$J(U, M) = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} \|\mathbf{x}_i - \mu_j\|^2 \tag{3.1}$$

Minimizacija gornjeg izraza ekvivalentna je minimizaciji sume kvadratnih razlika, izrazu s kojim se često susrećemo u nadziranom učenju, posebice regresijskim modelima. U *k-means* algoritmu razliku definiramo kao udaljenost podatka od centra klastera kojem pripada.

Euklidsku metriku $\|\mathbf{x}_i - \mu_j\|$ možemo zamijeniti metrikom Minkowskog $d_p(\mathbf{x}_i - \mu_j)$ i tada funkcija cilja poprima oblik

$$J_p(U, M) = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} d_p^p(\mathbf{x}_i, \mu_j) = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} |\mathbf{x}_i - \mu_j|^p \tag{3.2}$$

Dobre kandidate za točke u kojima ova funkcija poprima minimum, pod pretpostavkom da

je poznata matrica U, možemo odrediti izjednačavnjem parcijalnih derivacija s nulom

$$\begin{aligned}\frac{\partial}{\partial \mu_t} J_p(U, M) &= \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} \frac{\partial}{\partial \mu_t} |\mathbf{x}_i - \mu_j|^p \\ &= \sum_{i=1}^{m} u_{it} \frac{\partial}{\partial \mu_t} |\mathbf{x}_i - \mu_t|^p \\ &= \sum_{\mathbf{x}_i \in C_t} p |\mathbf{x}_i - \mu_t|^{p-1}\end{aligned}$$

U slučaju kada je $p = 2$ (Euklidska metrika) dobijemo

$$\sum_{\mathbf{x}_i \in C_t} 2(x_{il} - \mu_{tl}) = 0 \Rightarrow \mu_{tl} = \frac{1}{m_t} \sum_{\mathbf{x}_i \in C_t} x_{il} \tag{3.3}$$

Čime smo opravdali centroidnost ove metode.

Važno je naglasiti da za zadani skup podataka $\mathcal{X}$ kardinaliteta m kojeg želimo segmentirati u k klastera minimizirajući trag matrice W vrijede sljedeća svojstva:

(a) Funkcija cilja može poprimati $\vartheta(m, k)$ različitih vrijednosti.

(b) Funkcija cilja može imati više točaka lokalnog minimuma.

Standardni *k-means* algoritam, poznatiji pod nazivima naivni *k-means* algoritam ili Lloydov algoritam varijanta je već opisanog algoritma (2) u kojoj pretpostavljamo da je $w(\mathbf{x}_i) = 1, \forall i = 1, ..., m$.

3.1 Inkrementalna varijanta *k-means* algoritma

Inkrementalna verzija ovog algoritma iterativno prolazi kroz svaki podatak i provjerava može li relociranje tog podatka u neki drugi klaster dodatno minimizirati funkciju cilja. U slučaju da je uočena ta mogućnost, podatak se prebacuje u drugi klaster. Pretpostavimo da u određenoj iteraciji želimo podatak $\mathbf{x}^\star$ prebaciti iz klastera C_j u klaster C_l. Tada novi centar klastera C_l računamo formulom

$$\mu_l^\star = \frac{m_l \mu_l + \mathbf{x}^\star}{m_l + 1} = \mu_l + \frac{\mathbf{x}^\star - \mu_l}{m_l + 1} \tag{3.4}$$

Slično, novi centar klastera C_j računamo sa

$$\mu_j^\star = \mu_j - \frac{\mathbf{x}^\star - \mu_j}{m_j - 1} \tag{3.5}$$

Označimo sa $V(C_j)$ sumu kvadratnih pogrešaka u klasteru C_j. Tada vrijedi

$$V(C_j \setminus \{\mathbf{x}^\star\}) = V(C_j) - \frac{m_j}{m_j - 1}\|\mathbf{x}^\star - \mu_j\|^2 = V(C_j) - \delta_j(\mathbf{x}^\star)$$
$$V(C_l \cup \{\mathbf{x}^\star\}) = V(C_l) + \frac{m_l}{m_l + 1}\|\mathbf{x}^\star - \mu_l\|^2 = V(C_l) + \delta_l(\mathbf{x}^\star)$$

Dakle, relokacija podatka $\mathbf{x}^\star$ bila bi korisna ako je zadovoljen uvjet $\delta_j(\mathbf{x}^\star) > \delta_l(\mathbf{x}^\star)$, tj.

$$\frac{m_j}{m_j - 1}\|\mathbf{x}^\star - \mu_j\|^2 > \frac{m_l}{m_l + 1}\|\mu^\star - \mu_l|^2 \tag{3.6}$$

Na ovaj način definirali smo *BIMSEC* algoritam, poznatiji kao inkrementalna varijanta *k-means* algoritma.

Algoritam 3 *BIMSEC* algoritam

Ulazni parametri: Skup podataka $\mathcal{X}$ i broj klastera k
Izlazni parametri: Centri klastera $\{\mu_1, ..., \mu_k\}$

1: Odredi inicijalnu segmentaciju podataka u k klastera, i izračunaj centar svakog klastera.
2: Odaberi (sljedeći) podatak $\mathbf{x}^\star$ iz nekog klastera C_j.
3: Ako je $m_j = 1$ prebaci se na korak (6). U suprotnom za sve $l = 1, ..., k$ izračunaj inkrementalne vrijednosti

$$\delta_l(\mathbf{x}^\star) = \begin{cases} \frac{m_l}{m_l+1}\|\mathbf{x}^\star - \mu_l\|^2 & \text{, ako je } l \neq j \\ \frac{m_j}{m_j-1}\|\mathbf{x}^\star - \mu_j\|^2 & \text{, ako je } l = j \end{cases}$$

4: Pridruži podatak $\mathbf{x}^\star$ klasteru $C_{j^\star}$ ako je $\delta_{j^\star}(\mathbf{x}^\star) \leqslant \delta_l(\mathbf{x}^\star), \forall l = 1, ..., k$.
5: Izračunaj nove centre klastera C_j i $C_{j^\star}$ te vrijednost funkcije cilja $J(U, M)$.
6: Ako se vrijednost funkcije cilja nije promijenila nakon testiranja m podataka zaustavi algoritam. U suprotnom se vrati na korak (2).

Iako se radi o jeftinijoj (u smislu vremena izvršavanja) varijanti *k-means* algoritma, mana joj je veća vjerojatnost konvergencije u lokalni optimum (vidi [36, Ch. 10 str. 35-37]). Odabir jedne od ovih varijanti ovisi o tome kakav kompromis smo spremni postići između vremena računanja i kvalitete segmentacije.

3.2 Inicijalizacijske metode *k-means* algoritma

Velika prednost *k-means* algoritma u analizi velikog skupa podataka njegova je jednostavnost i linearno vrijeme izvršavanja obzirom na veličinu skupa. Ipak, ovaj algoritam sadrži i određene mane, pa navedimo neke od njih:

(a) Rezultat ovisi o načinu na koji su podaci sortirani i rasponu vrijednosti pojedinačnih komponenti.

(b) Radi se o pohlepnom (*greedy*) algoritmu obzirom da njegov ishod ovisi o inicijalnim postavkama.

(c) Iznimno je osjetljiv na postojanje *outliera*.

(d) Broj klastera mora biti zadan unaprijed.

(e) Moguće ga je koristiti jedino za numeričke podatke.

Za rješenje problema inicijalizacije, postoje razne "pametne" metode:

(i) Forgy u radu iz 1965. [16] predlaže odabir k nasumičnih podataka koji će biti tretirani kao centri klastera u početnoj iteraciji.

(ii) Hartigan i Wong 1979. [28] predlažu sortiranje svih podataka obzirom na udaljenost od centra cijelog skupa, $\bar{\mu}$. Nakon što numeriramo tako sortirani niz podataka $\{\mathbf{x}_{(1)}, ..., \mathbf{x}_{(m)}\}$, za svaki $j = 1, ..., k$ biramo inicijalni centar sa $\mu_j = \mathbf{x}_{(j')}$ pri čemu je $j' = 1 + \frac{m(j-1)}{k}$.

(iii) Gonzalez 1985. [41] predlaže odabir k podataka koje ćemo koristiti u inicijalizaciji, i to na način da odaberemo međusobno što udaljenije podatke. U prvom koraku postavljamo $\mu_1 = \arg\max_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}\|$. Daljnje korake definiramo induktivno. Naime, označimo s $\mathcal{M}$ skupa centara na početku j-tog korakaka, $\mathcal{M} = \{\mu_1, ..., \mu_{j-1}\}$. Sada odaberemo onaj $\mathbf{x} \in \mathcal{X} \setminus \mathcal{M}$ koji je najudaljeniji od skupa M. Podsjetimo, udaljenost točke $\mathbf{x}$ u metričkom prostoru $\mathcal{X}$ od skupa $\mathcal{M}$ definiramo sa $d(\mathbf{x}, \mathcal{M}) = min\{d(\mathbf{x}, \mathbf{m}) : \mathbf{m} \in \mathcal{M}\}$. Dakle,

$$\mu_j = \arg\max_{\mathbf{x}\in\mathcal{X}\setminus\mathcal{M}} \left(\min_{\mu\in\mathcal{M}} \|\mathbf{x} - \mu\| \right) \tag{3.7}$$

Nakon što ovaj postupak ponovimo k puta (uključujući i prvu iteraciju), dobijamo inicijalnu verziju skupa $\mathcal{M}$ s kojom nastavljamo dalje.

(iv) Kaufman i Rousseeuw 1990. [30] predlažu, slično kao i prethodno, još jednu induktivnu metodu definiranu na sljedeći način. U j-tom koraku definiramo matricu $[\beta_{il}]_{(m-j+1)\times(m-j+1)}$ pri čemu je $\beta_{il} = \max\{b_l - d(\mathbf{x}_i, \mathbf{x}_l), 0\}$ uz $b_l = \min_{t=1,...,j-1} d(\mathbf{x}_l, \mu_t)$. Sada odaberemo podatak $\mathbf{x}_i$ za koji je suma $\sum_l \beta_{il}$ maksimalna.

k-means++ algoritam

k-means++ inicijalzacijska je metoda korištena za poboljšanje *k-means* algoritma. Radi se o vjerojatnosnoj varijanti inicijalizacijske metode (iii), a uveli su je Arthur i Vassilvitskii u svom radu iz 2007. [12].

Neka je $u(\mathbf{x})$ udaljenost podatka $\mathbf{x} \in \mathcal{X}$ od skupa trenutnih centara $\mathcal{M}$. Tada sljedeći centar odaberemo nasumično prateći vjerojatnosnu distribuciju

$$p(\mathbf{x}) = \frac{u^2(\mathbf{x})}{\sum_{\mathbf{x}' \in \mathcal{X}} u^2(\mathbf{x}')} \tag{3.8}$$

Algoritam 4 *k-means++* algoritam

Ulazni parametri: Skup podataka $\mathcal{X}$ i broj klastera k
Izlazni parametri: Inicijalni centri klastera $\mathcal{M} = \{\mu_1, ..., \mu_k\}$
1: Nasumično odaberi $\mathbf{x} \in \mathcal{X}$ i pridruži $\mu_1 \leftarrow \mathbf{x}$. Neka je $\mathcal{M} = \{\mu_1\}$.
2: **for** j=2,...,k **do**
3: Za svaki podatak $\mathbf{x} \in \mathcal{X}$ izračunaj vrijednost $u(\mathbf{x}) = \min_{\mu \in \mathcal{M}} \|\mathbf{x} - \mu\|$.
4: Nasumično odaberi podatak $\mathbf{x} \in \mathcal{X}$ prateći distribuciju (3.8) i pridruži $\mu_j \leftarrow \mathbf{x}$.
5: Modificraj $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mu_j\}$.
6: **end for**
7: Izvrši *k-means* algoritam.

Pretpostavimo da imamo separabilni skup podataka i to takava da se podaci nalaze unutar kugala radijusa r (klastera), i da su centri svih klastera međusobno udaljeni za barem $4r$, te je kardinalnost svih klastera otprilike ista. Tada bi *k-means* algoritam gotovo sigurno pronašao globalni minimum ciljne funkcije, ako bi pri inicijalizaciji za centre klastera izabrali po jednu točku iz svakog klastera. U slučaju da koristimo nasumičnu inicijalizaciju vjerojatnost pozitivnog ishoda jednaka je $\frac{k!}{k^k}$. Dakle, za $k = 2$ radi se o vjerojatnosti pozitivnog ishoda od 50%, pa bi u slučaju ponavljanja cijelog postupka 7 puta pronašli globalni minimum sa sigurnošću od $1 - \left(\frac{2!}{2^2}\right)^7 > 99\%$. Za $k = 10$ vjerojatnost pojedinačnog pozitivnog ishoda iznosi $\frac{10!}{10^{10}} \approx 0.03\%$, pa bi za zadovoljavajuću razinu sigurnosti cijeli postupak trebali ponoviti 10000 puta. S druge strane, ako bi umjesto nasumične inicijalizacije koristili *k-means++*, u slučaju $k = 2$ bilo bi potrebno 3 puta ponoviti postupak, a za $k = 10$ otprilike 100 puta. Dakle, radi se o značajnom ubrzanju u broju koraka potrebnih da bi s određenom sigurnošću pronašli globalni minimum (vidi [40, str. 78]).

Iz ovih razloga, moguće je koristiti i sljedeću modifikaciju *k-means++* algoritma. Jedina promjena jest što kod inicijalizacije biramo centre koristeći distribuciju

$$p(\mathbf{x}) = \frac{u^s(\mathbf{x})}{\sum_{\mathbf{x}' \in \mathcal{X}} u^s(\mathbf{x}')} \tag{3.9}$$

Uz $s = 10$ i gore opisani slučaj za $k = 10$ vjerojatnost pojedinačnog uspjeha jednaka je $\approx 98\%$, pa je posljedično dovoljno izvršiti algoritam samo jednom. S druge strane, negativni efekt ovako visoko postavljenog parametra s povećanje je utjecaja *outliera* na konačan rezultat.

3.3 Efektivnost *k-means* algoritma

Jedan od problema kod izvršavanja *k-means* algoritma (u smislu vremena) računanje je udaljenosti $\|\mathbf{x}_i - \mu_j\|, \forall i = 1, ..., m, \forall j = 1, ..., k$. Dakle, u svakom koraku potrebno je $m \cdot k$ puta izračunati vrijednost zadane metrike. Očito dolazimo do problema s vremenom potrebnim za računanje u slučaju velikog skupa podataka i velikog broja klastera. Jedan od načina na koji možemo ubrzati cijeli proces je korištenje sljedećeg svojstva:

$$\|\mathbf{x}_i - \mu_j\|^2 = \|\mathbf{x}_i\|^2 + \|\mu_j\|^2 - 2\mathbf{x}_i^T\mu_j, \forall i = 1, ..., m, \forall j = 1, ..., k$$

Sada je u jednom koraku potrebno izračunati $\|\mathbf{x}_i\|^2, \forall i = 1, ..., m$, $\|\mu_j\|^2, \forall j = 1, ..., k$, te produkt matrica XM^T što nam može uvelike smanjiti vrijeme potrebno za izračun.

Također, *k-means* algoritam moguće je na jednostavan način paralelizirati na sljedeći način (vidi [44]).

Algoritam 5 Paralelizacija *k-means* algoritma

1: Uniformno podijeli skup $\mathcal{X}$ na P disjunktnih podskupova $\mathcal{X}_1, ..., \mathcal{X}_P$, i svaki od njih alociraj jednom od P dostupnih procesora.
2: Centralni (0-ti) procesor nasumično izabere k podataka koji predstavljaju centre klastera. Ta informacija šalje se svakom od procesora.
3: Svaki procesor $p = 1, ..., P$ odredi kojem klasteru njegovi podaci pripadaju, i izračuna sljedeće dvije veličine:
   (i) $s_{p,j} = \sum_{\mathbf{x} \in C_{p,j}} \mathbf{x}, \forall j = 1, ..., k$
   (ii) $m_{p,j} = |C_{p,j}|, \forall j = 1, ..., k$
4: Informacije o trenutnom stanju svakog od procesora šalju se centralnom u obliku liste $\left\langle j, (s_{Pj}, m_{Pj}) \right\rangle$.
5: Centralni procesor računa globalne centre klastera formulom

$$\mu_j = \frac{1}{\sum_{p=1}^{P} m_{p,j}} \sum_{p=1}^{P} s_{p,j}$$

   i šalje tu informaciju natrag svakom od procesora.
6: Koraci (3)-(5) ponavljaju se do postizanja zadovoljavajuće razina stabilnosti ili do konačnog broja ponavljanja.

3.4 Varijante *k-means* algoritma

Online verzija *k-means* algoritma

Algoritam 6 *Online* verzija *k-means* algoritma

1: Nasumično inicijaliziraj $\mu_1, ..., \mu_k$.
2: Obzirom na vjerojatnosnu distribuciju $p(\mathbf{x})$ odaberi podatak $\mathbf{x} \in \mathcal{X}$.
3: Odaberi pobjednika, tj. centar koji je najbliži odabranom elementu, $\mu_{s(\mathbf{x})}$.
4: Modificiraj koordinate pobjedničkog centra formulom

$$\mu_{s(\mathbf{x})} \leftarrow \mu_{s(\mathbf{x})} + \alpha(\mathbf{x} - \mu_{s(\mathbf{x})})$$

pri čemu je $\alpha \in \langle 0, 1]$ *learning* koeficijent.
5: Ponavljaj korake (2)-(4) dok se ne zadovolji neki od predefiniranih uvjeta.

Learning koeficijent α može biti zadan kao konstanta ili kao funkcija vremena, tj. iteracije.

U prvom slučaju, za iteraciju $t \in \mathbb{N}$ vrijedi

$$\mu_j(t) = (1-\alpha)^t \mu_j(0) + \alpha \sum_{i=1}^{t} (1-\alpha)^{t-i} \mathbf{x}^{(j)}(t) \tag{3.10}$$

Iz ove formulacije zaključujemo da na trenutnu vrijednost prototipa najviše utječe trenutni podatak, dok utjecaj prethodnih eksponencijalno opada s vremenom.

MacQueen u svom radu iz 1967. [24] predlaže ograničavanje *learning* koeficijenta postepeno u iteracijama na sljedeći način

$$\alpha(t) = \frac{\alpha_0}{t} \tag{3.11}$$

pri čemu je $\alpha_0 \in \langle 0, 1]$ i tada vrijedi

$$\mu_j(t) = \mu_j(t-1) + \alpha(t)\left(\mathbf{x}^{(j)}(t) - \mu_j(t-1)\right) = \frac{1}{t}\left(\mathbf{x}^{(j)}(1) + ... + \mathbf{x}^{(j)}(t)\right) \tag{3.12}$$

Fritzke 1997. [7] predlaže zadavanje *learning* koeficijenta sljedećom funkcijom iteracije

$$\alpha(t) = \alpha_p \left(\frac{\alpha_k}{\alpha_p}\right)^{\frac{t}{t_{\max}}} \tag{3.13}$$

pri čemu su α_p i α_k inicijalna i posljednja vrijednost koeficijenta redom, uz pretpostavku da je $\alpha_p > \alpha_k$, a $t_{\max}$ maksimalan broj iteracija.

Bisekcijska varijanta *k-means* algoritma

U praksi se pokazalo da postoje slučajevi u kojima aglomerativne hijerarhijske metode daju bolje konačne rezultate od *k-means* algoritma. Iz tog razloga Steinbach, Karypis i Kumar 2000. [34] uvode bisekcijsku varijantu *k-means* algoritma.

Algoritam 7 Bisekcijski *k-means* algoritam

Ulazni parametri: Skup podataka $\mathcal{X}$ i broj klastera k
Izlazni parametri: Centri klastera $\{\mu_1, ..., \mu_k\}$.

1: Podijeli skup $\mathcal{X}$ na dva klastera.
2: Odaberi jedan od klastera, i označi ga s C. (Preporuka je odabrati onaj najvećeg kardinaliteta.)
3: Podijeli klaster C na dva klastera koristeći *k-means* algoritam.
4: Ponovi korak (3) konačan broj puta i izaberi segmentaciju s najboljom vrijednosti kriterija.
5: Ponavljaj korake (2)-(4) dok je ukupan broj klastera $< k$.

Iako uobičajeni *k-means* algoritmi konvergiraju lokalnom minimumu funkcije cilja, algoritmi u kojima se on primjenjuje lokalno (poput ovog) nemaju to svojstvo. Ipak, bisecijska varijanta kreira klastere podjednakih veličina i zadovoljavajuće kvalitete.

Harmonijski *k-means* algoritam

Zapišimo funkciju cilja (3.1) u sljedećem obliku

$$J(U, M) = \sum_{i=1}^{m} \min_{j=1,...,k} \|\mathbf{x}_i - \mu_j\|^2 \tag{3.14}$$

U radu iz 2000. Zhang [8] je pokazao svojstvo funkcije $\min(a_1, ..., a_k)$, definirane na produktu skupova pozitivnih realnih brojeva, da je se može vjerodostojno aproksimirati harmonijskim prosjekom

$$m_h(a_1, ..., a_k) = \frac{k}{\sum_{j=1}^{k} \frac{1}{a_j}} \tag{3.15}$$

Nadalje, razlike između ovih dviju funkcija moguće je pojačati ili reducirati uvođenjem eksponencijalnog parametra za sve članove sume u nazivniku, tj.

$$m_h(a_1, ..., a_k) = \frac{k}{\sum_{j=1}^{k} \left(\frac{1}{a_j}\right)^p} \tag{3.16}$$

U našem slučaju radi se o sljedećoj aproksimaciji funkcije cilja

$$J_h(U, M) = \sum_{i=1}^{m} \frac{k}{\sum_{j=1}^{k} ||\mathbf{x}_i - \mu_j||^{-p}} \tag{3.17}$$

pri čemu je $p \geqslant 2$, a u istom radu se preporuča korištenje parametra $p = 3.5$.

Incidencijsku matricu segmentacije definiramo sa

$$u_{ij} = \frac{||\mathbf{x}_i - \mu_j||^{-2-p}}{\sum_{j=1}^{k} ||\mathbf{x}_i - \mu_j||^{-2-p}} \in [0, 1] \tag{3.18}$$

Dakle, ova varijanta *k-means* algoritma spada u grupu *fuzzy* segmentacija.

Nadalje, uvodimo težinsku funkciju za svaku od točaka sa

$$w(\mathbf{x}_i) = \frac{\sum_{j=1}^{k} ||\mathbf{x}_i - \mu_j||^{-2-p}}{\left(\sum_{j=1}^{k} ||\mathbf{x}_i - \mu_j||^{-p}\right)^2} \tag{3.19}$$

Da bismo izbjegli probleme u slučaju kada je $\mathbf{x}_i \approx \mu_j$ izraz $||\mathbf{x}_i - \mu_j||$ mijenjamo izrazom $\max\{||\mathbf{x}_i - \mu_j||, \epsilon\}$ pri čemu je $\epsilon > 0$.

Konačno, nove centre klastera računamo na sljedeći način

$$\mu_j = \frac{\sum_{i=1}^{m} u_{ij} w(\mathbf{x}_i) \mathbf{x}_i}{\sum_{i=1}^{m} u_{ij} w(\mathbf{x}_i)} \tag{3.20}$$

k-medoids algoritam

Dosada obrađene varijante *k-means* algoritma koristile su Euklidsku metriku za računanje razlika među podacima. To znači da u slučaju kada imamo komponentu koja popirma kategorijske vrijednosti, moramo ju na neki način transformirati u komponentu koja poprima numeričke vrijednosti. Osim toga, *k-means* algoritam se zbog svoje funkcije cilja teško nosi s pojavom *outliera*. Da bismo pokušali izbjeći ove probleme, uvodimo novu funkciju cilja (vidi [26, str. 515-518]).

$$J_{\text{med}}(\mathbf{p}_1, ..., \mathbf{p}_k) = \sum_{i=1}^{m} \min_{j=1,...,k} d(\mathbf{x}_i, \mathbf{p}_j) \tag{3.21}$$

pri čemu su $\mathbf{p}_1, ..., \mathbf{p}_k \in \mathcal{X}$ prototipovi klastera, a $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ mjera razlikovanja. U ovom slučaju nije nužno da je mjera razlikovanja metrika, čak štoviše, ni simetrična. Dakle, dovoljna nam je matrica razlikovanja $D = [d_{ij}]$.

Obzirom da su prototipovi klastera elementi skupa $\mathcal{X}$, možemo odabrati skup $\mathcal{K} \subseteq \{1, ..., m\}$ čiji elementi predstavljaju indekse tih prototipova u skupu $\mathcal{X}$. Sada umjesto matrice M definiramo vektor $\mathbf{c} = (c_1, ..., c_m)$ sa

$$c_i = \arg\min_{j \in \mathcal{K}} d_{ij} \tag{3.22}$$

Algoritam 8 *k-medoids* algoritam

Ulazni parametri: Matrica razlikovanja D i broj klastera k

Izlazni parametri: Prototipovi klastera $\{\mathbf{p}_1, ..., \mathbf{p}_k\}$

1: Inicijaliziraj skup $\mathcal{K} \subseteq \{1, ..., m\}$

2: Modificiraj vektor $\mathbf{c}$ sa

$$c_i = \arg\min_{j \in \mathcal{K}} d_{ij}$$

3: Modificiraj skup $\mathcal{K} = \{j_1, ..., j_k\}$ sa

$$j_l \leftarrow \arg\min_{t: c_t = j_l} \sum_{t': c_{t'} = j_l} d_{t't}$$

4: Ponavljaj korake (2)-(3) do postizanja zadovoljavajuće razine stabilnosti skupa $\mathcal{K}$.

I na kraju, pravilom *the winner takes all* možemo definirati klastere $C_1, ..., C_k$

$$C_j = \{\mathbf{x}_i \in \mathcal{X} : j = \arg\min_{j'=1,...,k} d(\mathbf{x}_i, \mathbf{p}_{j'})\} \tag{3.23}$$

k-modes algoritam

Huang u svom radu iz 1998. [45] predstavlja *k-modes* algoritam koji olakšava pristup problemu segmentacije u slučaju kada komponente poprimaju kategorijske vrijednosti. Tada udaljenosti među podacima računamo sa

$$d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{n} \delta(x_{il}, x_{jl}) \tag{3.24}$$

pri čemu je

$$\delta(x_{il}, x_{jl}) = \begin{cases} 1 & \text{, ako je } x_{il} = x_{jl} \\ 0 & \text{, inače} \end{cases}$$

Neka je zadan skup C i objekt $\mathbf{m}$ (ne nužno element iz $\mathcal{X}$). Udaljenost između njih računamo sa

$$d(C, \mathbf{m}) = \sum_{\mathbf{x} \in C} d(\mathbf{x}, \mathbf{m})$$

Tada možemo definirati *mod* skupa C kao objekt u kojem gore navedena funkcija poprima minimum.

Lema 3.4.1. *Neka je Dom_l skup svih mogućih vrijednosti l-te komponente i c_{ls} broj objekata u skupu C čija je vrijednosti l-te komponente jednaka s. Tada* mod *skupa C, vektor $\mathbf{m}$, računamo sa*

$$m_l = arg \max_{s \in Dom_l} c_{ls}, \forall l = 1, ..., n \tag{3.25}$$

Dakle, u *k-modes* algoritmu pokušavamo pronaći takve $\mathbf{m}_1, ..., \mathbf{m}_k$ u kojima funkcija

$$J_{\text{mod}}(\mathbf{m}_1, ..., \mathbf{m}_k) = \sum_{i=1}^{m} \min_{j=1,...,k} d(\mathbf{x}_i, \mathbf{m}_j) \tag{3.26}$$

poprima minimalnu vrijednost.

Algoritam 9 *k-modes* algoritam

Ulazni parametri: Matrica podataka $\mathcal{X}$ i broj klastera k
Izlazni parametri: Prototipovi klastera $\{\mathbf{m}_1, ..., \mathbf{m}_k\}$

1: Inicijaliziraj k vrijednosti $\mathbf{m}_1, ..., \mathbf{m}_k$.
2: Koristeći (3.24) svaki podatak pridruži najbližem elementu $\mathbf{m}_j$, tj. klasteru C_j.
3: Modificiraj *mod* svakog klastera koristeći lemu (3.4.1).
4: Ponavljaj korake (2)-(3) do postizanja zadovoljavajuće razine stabilnosti prototipova $\{\mathbf{m}_1, ..., \mathbf{m}_k\}$.

Općenito, pretpostavimo da prvih n_1 komponenti poprima kontinuirane, a preostalih $n - n_1$ kategorijske vrijednosti. Tada funkciju cilja možemo zapisati u sljedećem obliku

$$J_p(U, M) = \sum_{j=1}^{k} \sum_{i=1}^{m} \left[u_{ij} \sum_{l=1}^{n_1} (x_{il} - \mu_{jl})^2 + \gamma u_{ij} \sum_{l=n_1+1}^{n} \delta(x_{il}, \mu_{jl}) \right] \tag{3.27}$$

gdje je $\gamma > 0$ koeficijent balansiranja, a vektori $\mu_1, ..., \mu_k$ prototipovi klastera dobiveni kombinacijom računanja (3.3) za kontinuirane komponente, i (3.4.1) za kategorijske komponente.

Optimizacija ovog kriterija provodi se iterativno, modificiranjem matrica M i U. Pri tome cijelo vrijeme koristeći mjeru razlikovanja definiranu sa

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n_1} (x_i - y_i)^2 + \gamma \sum_{i=n_1+1}^{n} \delta(x_i, y_i) \tag{3.28}$$

Reducirani *k-means* algoritam

U praksi se često susrećemo sa velikim brojem komponenti, od kojih dio njih nije relevantan za postizanje željenog cilja. U tom slučaju nastaje problem sa potencijalnim šumom koji nam takve komponente uvode kod standardnog *k-means* algoritma. Kao odgovor na ovaj problem, De Soete i Carroll 1994. [17] predlažu korištenje reduciranog *k-means* algoritma. On pokušava minimizirati funkciju cilja

$$J(U, M, A) = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} \|\mathbf{x}_i - A\mu_j\|^2 = \|X - UMA^T\|_F^2 \tag{3.29}$$

pri čemu je $A \in \mathbb{R}^{n \times q}$ matrica s ortonormiranim stupcima, $q < \min(n, k-1)$, M matrica čiji su reci centri klastera koji se nalaze u q-dimenzionalnom prostoru $\mathbb{R}^q$, a $\|\cdot\|_F$ Frobeniusova norma.

Algoritam 10 Reducirani *k-means* algoritam

Ulazni parametri: Matrica podataka X i broj klastera k
Izlazni parametri: Segmentacija $\{C_1, ..., C_k\}$

1: Inicijaliziraj matrice A, M i U.
2: Izračunaj SVD dekompoziciju $Q\Sigma P^T$ matrice $(UF)^T X$.
3: Modificiraj matricu A sa $A \leftarrow PQ^T$.
4: Modificiraj matricu U tako da za svaki podatak $\mathbf{x}_i$ odrediš kojem transformiranom centru $A\mu_j$ je najbliži.
5: Izračunaj novu aproksimaciju matrice M sa $M \leftarrow (UTU)^{-1}U^T XA$.
6: izračunaj novu vrijednost $J(U, M, A)$ pomoću formule (3.29). Ako joj se vrijednost značajno smanjila vrati se na korak (2).

Obzirom da je matrica U binarna, moguće je da će algoritam završiti u lokalnom minimumu, stoga se preporuča višestruko izvršavanje ovog algoritma. Još jedna od njegovih mana je činjenica da se vrijednosti k i q moraju zadati unaprijed.

Osim reduciranja *k-means* algoritma na prostor niže dimenzije, problem možemo riješiti i uvođenjem težinskog vektora po komponentama. Ovu varijantu algoritma, poznatiju pod nazivom *sparse clustering k-means*, uvode Witten i Tibshirani u svom radu iz 2010. [14]. Optimizacijska zadaća te varijante maksimizacija je kvadratnih grešaka među klasterima, tj. maksimizacija funkcije cilja

$$J(U, M, \mathbf{w}) = \sum_{i=1}^{m} \sum_{j=1}^{k} \sum_{l=1}^{n} w_l (x_{il} - \mu_l)^2 - \sum_{i=1}^{m} \sum_{j=1}^{k} \sum_{l=1}^{n} u_{ij} w_l (x_{il} - \mu_{jl})^2 \tag{3.30}$$

$$\text{uz } w_l \geqslant 0, \forall l = 1, ..., n$$
$$\sum_{l=1}^{n} w_l^2 \leqslant 1$$
$$\sum_{l=1}^{n} w_l < s$$

pri čemu je s unaprijed zadan parametar. Inicijalno se postavi $\mathbf{w} = \frac{1}{\sqrt{n}}\mathbf{e}$. Potom se izvrši *k-means* algoritam i nakon što dosegnemo (lokalni) optimum, zadržimo dobivene klastere te optimiziramo $\mathbf{w}$ preko gore navedenih uvjeta. Kao konačan rezultat imamo segmentaciju i važnost svake od komponenti za njeno postizanje. Obzriom na kompleksnost ovog algoritma gotovo je nemoguće odrediti globalni optimumu funkcije cilja.

3.5 Validacija i odabir broja klastera

U većini prethodno opisanih algoritama od korisnika se zahtjeva odabir željenog broja klastera. Međutim, često nije jasno koja bi vrijednost najviše odgovarala danom skupu podataka. Ovo je samo jedan od primjera zbog kojeg želimo definirati mjere kojima mjerimo kvalitetu segmentacije. U praksi se često testira učinak raznih algoritama, njihovih parametara i odabir komponenti na kranji ishod segmentacije, stoga je potrebno definirati mjeru kojom uspoređujemo ishode različitih postupaka segmentacije. Navedimo neke od njih:

(a) Najprirodniji pristup je mjerenje odstupanja sljedećim izrazom

$$q(\mathcal{S}) = \frac{1}{k}\sum_{j=1}^{k}\frac{1}{m_j}\sum_{\mathbf{x}\in C_j} d(\mathbf{x}, \mu_j) \tag{3.31}$$

pri čemu je poželjno minimizirati navedeni izraz.

(b) Caliński i Harabasz 1974. [42] predlažu maksimizaciju sljedećeg kriterija

$$q_{\text{CH}}(\mathcal{S}) = \frac{tr(B)}{k-1}\frac{m-k}{tr(W)} \tag{3.32}$$

(c) Davies i Bouldin 1979. [13] predlažu minimizaciju sljedećeg kriterija

$$\text{DB}(\mathcal{S}) = \frac{1}{k}\sum_{j=1}^{k}\max_{l=1,...,k,l\neq j}\frac{\rho(C_j)+\rho(C_l)}{d(\mu_j,\mu_l)} \tag{3.33}$$

pri čemu je

$$\rho(C_j) = \frac{1}{m_j} \sum_{\mathbf{x} \in C_j} \|\mathbf{x} - \mu_j\|, \forall j = 1, ..., k$$

(d) Kaufman i Rousseeuw 1990. [30] uvode pojam *silhouette* koeficijenta. Neka je $\mathbf{x} \in \mathcal{X}$ i $C_j \in \mathcal{S}$ klaster kojem taj podatak pripada. Definiramo

$$a(\mathbf{x}) = \frac{1}{m_j - 1} \sum_{\mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$

$$b(\mathbf{x}) = \min_{l=1,...,k, l \neq j} \frac{1}{m_l} \sum_{\mathbf{y} \in C_l} d(\mathbf{x}, \mathbf{y})$$

$$s(\mathbf{x}) = \begin{cases} \frac{b(\mathbf{x}) - a(\mathbf{x})}{\max\{a(\mathbf{x}), b(\mathbf{y})\}} & \text{, ako je } m_j > 1 \\ 0 & \text{, inače} \end{cases}$$

Sada kao kriterij možemo koristiti maksimizaciju *silhouette* koeficijenta

$$\mathrm{SC}(\mathcal{S}) = \frac{1}{m} \sum_{i=1}^{m} s(\mathbf{x}_i) \tag{3.34}$$

(e) Dunn 1973. [21] predlaže maksimizaciju sljedećeg kriterija

$$D(\mathcal{S}) = \min_{j=1,...,k} \left[\min_{l=j+1,...,k} \frac{d(C_j, C_l)}{\max_{v=1,...,k} \mathrm{diam}(C_v)} \right] \tag{3.35}$$

pri čemu je

$$d(C_j, C_l) = \min\{d(\mathbf{x}, \mathbf{y} : \mathbf{x} \in C_j, \mathbf{y} \in C_l)\}$$
$$\mathrm{diam}(C_v) = \max_{\mathbf{x}, \mathbf{y} \in C_v} d(\mathbf{x}, \mathbf{y})$$

Često pomoću ovih mjera, osim što uspoređujemo različite segmentacije, biramo i parametar k za kojeg te mjere postižu najoptimalniju vrijednost. Jedan od prijedloga za odabir prikladnog parametra k je *GAP* statistika koju uvode Tibshirani, Walther i Hastie 2001. [38]. Radi se o formalnom zapisu *elbow* metode. Naime, definirajmo izraz

$$W(k) = \sum_{j=1}^{k} \frac{1}{m_j} \sum_{\mathbf{x}, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y}) \tag{3.36}$$

Elbow metoda bazira se na grafičkom prikazu vrijednosti ove funkcije u ovisnosti o k i biranju posljedenjeg u nizu k koji postiže značajno manju vrijednost od svog neposrednog prethodnika (grafički prikaz tada podsjeća na vizualizaciju ruke, a točka koju biramo predstavlja lakat).

Ideja *GAP* statistike je uspoređivanje vrijednosti $\log(W_k)$ sa njenim očekivanjem. Autori definiraju statistiku $\text{Gap}_m(k) = \mathbb{E}_m^\star [\log(W_k)] - \log(W_k)$ pri čemu je $\mathbb{E}_m$ očekivana vrijednost pri odabiru m-dimenzionalnog slučajnog uzorka iz referentne distribucije, a cilj je pronaći k koji maksimizira ovu statistiku.

Poglavlje 4

Pregled ostalih odabranih metoda segmentacije

4.1 Hijerarhijske metode segmentacije

Hijerarhijske metode segmentacije baziraju se na uzastopnim grupiranjima ili podjelama podataka i njihovih podskupova. Rezultat ovih metoda je stablasta struktura koju nazivamo dendogram.

Aglomerativne metode (*bottom-up*) inicijalno svaki podatak promatraju kao zaseban klaster, te ih postepeno grupiraju sve dok na kraju ne ostane jedan m-dimenzionalan klaster.

Algoritam 11 Aglomerativna hijerarhijska metoda

Ulazni parametri: Skup podataka $\mathcal{X}$
Izlazni parametri: Dendogram $\{C_1, ..., C_{2m-1}\}$

1: Kreiraj m jednočlanih klastera i izračunaj udaljenosti između svih parova. Spremi izračunate vrijednosti u matricu $D = [d_{ij}]_{m \times m}$.
2: Pronađi par klastera C_i i C_j koji su međusobno najbliži.
3: Kreiraj novi klaster $C_k = C_i \cup C_j$. U dendogramu kreiraj novi čvor koji predstavlja klaster C_k i poveži ga s klasterima C_i i C_j.
4: Izračunaj udaljenost klastera C_k sa svim preostalim klasterima, osim s C_i i C_j.
5: Izbaci iz matrice D retke i stupce koji predstavljaju klastere C_i i C_j te dodaj redak i stupac koji predstavlja klaster C_k.
6: Ponavljaj korake (2)-(5) sve dok je kvadratna matrica D dimenzije > 1.

Da bismo izvršili navedeni algoritam potrebno je odabrati metodu kojom računamo udaljenosti između klastera. Pod pretpostavkom da smo odabrali neku od mjera udaljenosti među podacima, udaljenosti klastera možemo računati na jedan od sljedećih načina:

(a) (*Single linkage* metoda ili metoda najbližih susjeda)

$$d(C_i, C_j) = \min\{d(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in C_i, \mathbf{y} \in C_j\} \tag{4.1}$$

Udaljenost između klastera jednaka je udaljenosti dvaju najbližih elemenata iz različitih klastera.

(b) (*Complete linkage* metoda ili metoda najudaljenijih susjeda)

$$d(C_i, C_j) = \max\{d(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in C_i, \mathbf{y} \in C_j\} \tag{4.2}$$

Udaljenost između klastera jednaka je udaljenosti dvaju najudaljenijih elemenata iz različitih klastera.

(c) (*Average linkage* metoda)

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

Udaljenost između klastera jednaka je prosječnoj udaljenosti elemenata iz različitih klastera.

(d) (Metoda centara)

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

Udaljenost između klastera jednaka je udaljenosti između njihovih centara.

Nasuport aglomerativnim metodama imamo metode podjele (*top-down*). Te metode inicijalno promatraju cijeli skup podataka kao jedan klaster te ga postepeno dijele sve dok ne ostane m jednočlanih klastera.

Pretpostavimo da smo izvršili jedan od prethodno navedenih algomerativnih hijerarhijskih algoritama. Tada je moguće kreirati matricu $D_T = [d_{ij}^{(T)}]_{m \times m}$ čiji elementi predstavljaju udaljenosti između klastera uzete u obzir kod grupiranja u kojem su se odgovarajući elementi prvi puta našla u zajedničkom klasteru. Nadalje, neka $D = [d_{ij}]_{m \times m}$, kao i prije, predstavlja matricu udaljenosti između podataka. Definirajmo vektore

$$E = (d_{12}, d_{13}, ..., d_{1m}, d_{23}, d_{24}, ..., d_{2m}, ..., d_{m-1m})^T$$
$$T = (d_{12}^{(T)}, d_{13}^{(T)}, ..., d_{1m}^{(T)}, d_{23}^{(T)}, d_{24}^{(T)}, ..., d_{2m}^{(T)}, ..., d_{m-1m}^{(T)})^T$$

Sada možemo definirati *cophenetic* koeficijent korelacije dendograma (vidi [37]) kao Pearsonov koeficijent korelacije između vektora E i T. Radi se o mjeri koja nam opisuje koliko vjerodostojno struktura dendograma čuva udaljenosti između podataka.

Najznačajniji problemi s kojima se susrećemo koristeći hijerarhijske metode su:

(a) Teško zadržavanje jasnoće pri analizi velikog skupa podataka.

(b) Nemogućnost prebacivanja podatka iz jednog klastera u drugi, iako je možda u početnim koracima pogrešno segmentiran.

(c) Konačan rezultat odražava strukturu koja je pretpostavljena odabirom algoritma.

4.2 *EM* algoritam

Umjesto da se koncentriramo na zadani skup podataka, segmentaciju je moguće izvršiti i promatrajući dani skup podataka kao slučajni uzorak iz neke vjerojatnosne distribucije. Najčešće se pretpostavlja da se radi o mješovitoj normalnoj distribuciji, tj. kombinaciji više n-dimenzionalnih normalnih distribucija. Njena gustoća je definirana sa

$$f(\mathbf{x}) = \sum_{j=1}^{k} \mathbb{P}(C_j) f(\mathbf{x}|C_j; \mu_j, \Sigma_j) \tag{4.3}$$

pri čemu je $\mathbb{P}(C_j)$ apriori vjerojatnost da podatak pripada j-toj normalnoj distribuciji, tj. j-tom klasteru, te $f(\mathbf{x}|C_j, \mu_j, \Sigma_j)$ funkcija gustoće n-dimenzionalne normalne distribucije s parametrima (μ_j, Σ_j) j-tog klastera,$\forall\, j = 1, ..., k$.

Podsjetimo se,

$$f(\mathbf{x}|C_j; \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{n/2}\sqrt{\det \Sigma_j}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_j)^T \Sigma_j^{-1} (\mathbf{x} - \mu_j)\right] \tag{4.4}$$

Koristeći ove pretpostavke Dempster, Laird i Rubin 1977. predstavljaju *EM* algoritam [6]. Za prototipove klastera uzimaju očekivanu vrijednost distribucije klastera, a umjesto matrice U, promatraju matricu $P = [p_{ij}] \in [0, 1]^{m\times k}$ pri čemu je $p_{ij} = \mathbb{P}(C_j|\mathbf{x}_i)$ uvjetna vjerojatnost. Funkciju cilja, koju je potrebno minimizirati, definiraju sa

$$J_{\mathrm{EM}}(P, M) = -\sum_{i=1}^{m} \log\left(\sum_{j=1}^{k} f(\mathbf{x}_i|C_j)\mathbb{P}(C_j)\right) \tag{4.5}$$

Sam naziv algoritma nam daje naznaku da postoje dva glavna koraka pomoću kojih pronalazimo rješenje ovog problema:

(i) Korak **E** (*Expectation*)
U ovom koraku procjenjujemo vjerojatnost da i-ti podatak pripada j-tom klasteru, uz pretpostavku da su nam poznati svi parametri mješovite normalne distribucije. Tu vrijednosti možemo izračunati koristeći Bayesov teorem

$$\mathbb{P}(C_j|\mathbf{x}_i) = \frac{f(\mathbf{x}_i|C_j)\mathbb{P}(C_j)}{f(\mathbf{x}_i)} = \frac{f(\mathbf{x}_i|C_j)\mathbb{P}(C_j)}{\sum_{l=1}^{k} \mathbb{P}(C_l) f(\mathbf{x}_i|C_l)} \tag{4.6}$$

(ii) Korak **M** (*Maximization*)
U ovom koraku pretpostavljamo da znamo kojem klasteru koji podatak pripada. Tada metodom maksimalne vjerodostojnosti procjenjujemo parametre svih distribucija.

$$\mathbb{P}(C_j) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{P}(C_j|\mathbf{x}_i) \tag{4.7}$$

$$\mu_j = \frac{1}{m\mathbb{P}(C_j)}\sum_{i=1}^{m}\mathbb{P}(C_j|\mathbf{x}_i)\mathbf{x}_i \tag{4.8}$$

$$\Sigma_j = \frac{1}{m\mathbb{P}(C_j)-1}\sum_{i=1}^{m}\mathbb{P}(C_j|\mathbf{x}_i)(\mathbf{x}_i-\mu_j)(\mathbf{x}_i-\mu_j)^T \tag{4.9}$$

Jezikom *k-means* algoritma korak **E** ekvivalent je modifikaciji matrice U, a korak **M** modifikaciji matrice M.

Algoritam 12 *EM* algoritam

Ulazni parametri: Matrica podataka X i broj klastera k
Izlazni parametri: Skup parametara $(\mu_1, \Sigma_1), ..., (\mu_k, \Sigma_k)$ mješovite normalne distribucije, i skup vjerojatnosti $\mathbb{P}(C_j|\mathbf{x}_i)$

1: Postavi $t \leftarrow 0$ i inicijaliziraj procjene parametara $\mathbb{P}(C_j), \mu_j^t, \Sigma_j^t$ za sve $j = 1, ..., k$.
2: (Korak **E**) Koristeći Bayesov teorem izračunaj vrijednosti $\mathbb{P}^{t+1}(C_j|\mathbf{x}_i)$.
3: (Korak **M**) Izračunaj vrijednosti $\mathbb{P}^{t+1}(C_j), \mu_j^{t+1}, \Sigma_j^{t+1}$.
4: Modificiraj $t \leftarrow t + 1$.
5: Ponavljaj korake (2)-(4) dok se vrijednosti parametara distribucije ne stabiliziraju.

EM algoritam je obzirom na kvalitetu usporediv sa *k-means* algoritmom. U oba slučaja izvedba ovisi o inicijalizaciji parametara. Ipak, *EM* algoritam nije pogodan za izvršenje nad podacima velikih dimenzija, obzirom da računanje kovarijacijske matrice zahtjeva izračun $\frac{1}{2}(n^2+n)$ elemenata. Još jedna mana ovog algoritma spora je konvergencija. Nadalje, ako modificiramo korak **E** na način da primjenimo pravilo *the winner takes all* i korak **M** tako da za procjenu kovarijacijske matrice koristimo $\Sigma_j = \epsilon\mathbb{I}$ onda se radi o *k-means* algoritmu kada $\epsilon \to 0$ (vidi [40, str. 99]).

4.3 Metode bazirane na gustoći

Klastere u podacima možemo interpretirati kao gusta područja u prostoru okružena prostorima niske gustoće. Ova interpretacija omogućuje uspješnu segmentaciju nepravilnih oblika, kao i otpornost na *outliere*.

Definicija 4.3.1. *Neka je zadana metrika* $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *i parametri* $\epsilon > 0, m^{(p)} \in \mathbb{N}$.

(a) Definiramo ϵ*-susjedstvo elementa* $\mathbf{x} \in \mathcal{X}$ *sa*

$$N_\epsilon(\mathbf{x}) = \{\mathbf{y} \in \mathcal{X} : d(\mathbf{x}, \mathbf{y}) \leqslant \epsilon\}$$

(b) Za točku $\mathbf{x} \in \mathcal{X}$ *kažemo da je unutarnja točka skupa ako je* $|N_\epsilon(\mathbf{x})| \geqslant m^{(p)}$.

(c) Za točku $\mathbf{x} \in \mathcal{X}$ *kažemo da je rubna točka skupa ako je* $|N_\epsilon(\mathbf{x})| < m^{(p)}$ *i skup* $N_\epsilon(\mathbf{x})$ *sadrži barem jednu unutarnju točku tog skupa.*

(d) Ako točka $\mathbf{x} \in \mathcal{X}$ *nije ni unutarnja ni rubna točka, onda ju nazivamo* outlierom.

Definicija 4.3.2. *Za točku* $\mathbf{y} \in \mathcal{X}$ *kažemo da je direktno dohvatljiva obzirom na točku* $\mathbf{x} \in \mathcal{X}$ *ako je* $\mathbf{y} \in N_\epsilon(\mathbf{x})$ *i* $\mathbf{x}$ *je unutarnja točka skupa* $\mathcal{X}$.

Definicija 4.3.3. *Za točku* $\mathbf{y} \in \mathcal{X}$ *kažemo da je dohvatljiva obzirom na točku* $\mathbf{x} \in \mathcal{X}$ *ako postoji konačan niz točaka* $\mathbf{x}_1, ..., \mathbf{x}_n$ *t.d. je* $\mathbf{x} = \mathbf{x}_1$, $\mathbf{y} = \mathbf{x}_n$ *i* $\mathbf{x}_{i+1}$ *direktno je dohvatljiva obzirom na točku* $\mathbf{x}_i$, $\forall i = 1, ..., n-1$.

Definicija 4.3.4. *Točke* $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ *su povezane obzirom na gustoću ako postoji* $\mathbf{z} \in \mathcal{X}$ *takva da su* $\mathbf{x}$ *i* $\mathbf{y}$ *dohvatljive obzirom na* $\mathbf{z}$.

Najpoznatiji predstavnik ove grupe algoritama je *DBSCAN* algoritam. Predstavljaju ga Ester, Kriegel, Sander i Xu 1996. [33], a baziran je na sljedećim pravilima:

(i) Sve točke u jednom klasteru međusobno su povezane obzirom na gustoću.

(ii) Ako je unutarnja točka povezana obzirom na neku točku klastera, onda i ona pripada tom klasteru.

Algoritam 13 *DBSCAN* algoritam

Ulazni parametri: Skup podataka $\mathcal{X}$ i parametri $\epsilon, m^{(p)}$
Izlazni parametri: Segmentacija $\{C_1, ..., C_k\}$

1: Označi svaku točku skupa $\mathcal{X}$ kao unutarnju, rubnu ili *outlier*.
2: Ukloni *outliere*.
3: Poveži bridovima ϵ-susjedne unutarnje točke.
4: Formiraj klastere od navedenih točaka.
5: Pridruži rubne točke jednom od klastera koristeći pravilo (ii).

Odabir odgovarajućih parametara ϵ i $m^{(p)}$ ima jak utjecaj na konačan rezultat algoritma. Odabir prevelike vrijednosti ϵ može rezultirati sa $|N_\epsilon(\mathbf{x})| = m, \forall \mathbf{x} \in \mathcal{X}$. S druge strane odabir premale vrijednost ϵ rezultira sa $|N_\epsilon(\mathbf{x})| = 1, \forall \mathbf{x} \in \mathcal{X}$.

Stoga uvodimo funkciju $g : \mathcal{X} \to \mathbb{R}$ koja karakterizira gustoću u točki $\mathbf{x} \in \mathcal{X}$ (vidi [40, str. 53])

$$g(\mathbf{x}) = \sum_{\mathbf{x}' \in \mathcal{X}} f(\mathbf{x}, \mathbf{x}')$$

pri čemu je $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ proizvoljna funkcija, primjerice

$$f(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{d(\mathbf{x}, \mathbf{x}')}{2\sigma^2}\right) \text{ uz } \sigma > 0$$

Prednosti *DBSCAN* algoritma su sljedeće:

(a) Nije potrebno zadati parametar k.

(b) Klasteri mogu biti proizvoljnih oblika.

(c) Podaci mogu sadržavati *outliere*.

(d) Algoritam je minimalno osjetljiv na poredak podataka u skupu

Poglavlje 5

Spektralna segmentacija

U ovom poglavlju obradit ćemo segmentacijske metode baziranu na sličnostima između podataka. Konkretno, ne zahtjevamo više metrički prostor, čak niti funkciju sličnosti. Umjesto toga, moguće je segmentirati podatke samo na temelju relacija među podacima, najčešće reprezentiranih grafom.

5.1 Formalan zapis

Definicija 5.1.1. *Neka je $V \neq \emptyset$.*

(i) Uređeni par $G = (V, E)$, pri čemu je E podskup dvočlanih podmultiskupova skupa V, nazivamo (neusmjerenim) grafom. Skup V nazivamo skupom vrhova, a skup E skupom bridova. Ako su V i E konačni skupovi tada kažemo da je graf G konačan.

(ii) Ako je uz to zadana i funkcija $w : E \to \mathbb{R}$ onda za G kažemo da je težinski graf.

(iii) Za vrhove $u, v \in V$ kažemo da su povezani ako postoji konačan niz vrhova $u = v_1, v_2, ..., v_n = v$ takvi da za svaki $i = 2, ..., n$ vrijedi $\{v_{i-1}, v_i\} \in E$. Za graf G kažemo da je povezan ako su svi vrhovi međusobno povezani.

(iv) Za graf $G' = (V', E')$ kažemo da je podgraf grafa G ako je $V' \subseteq V$ i $E' \subseteq E$.

Definicija 5.1.2. *Neka je $G = (V, E)$ neusmjereni, težinski graf. Matricu $S = [s_{ij}] \in \mathbb{R}^{m \times m}$ definiranu sa*

$$s_{ij} = \begin{cases} w(v_i, v_j) & \text{, ako je } \{v_i, v_j\} \in E \\ 0 & \text{, inače} \end{cases} \tag{5.1}$$

nazivamo incidencijskom matricom grafa G.

Pretpostavimo da je, kao i prije, zadan m-dimenzionalan skup podataka $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$. Ovog puta, umjesto mjere razlikovanja, pretpostavimo da je zadana matrica $[w_{ij}] \in \mathbb{R}^{m\times m}$, pri čemu w_{ij} predstavlja relaciju između i-tog i j-tog podatka. Sada na jednostavan način možemo definirati pripadni neusmjereni, težinski graf $G = (V, E)$ u kojem skup vrhova $V = \{v_1, ..., v_m\}$ predstavlja podatke, skup bridova $E = \{\{v_i, v_j\} : i, j = 1, ..., m \text{ t.d. je } w_{ij} \neq 0\}$ relacije, a težinska funkcija $w : E \to \mathbb{R}$ definirana sa $w(\{v_i, v_j\}) = w_{ij}$ intezitet relacije između i-tog i j-tog podatka. Nadalje pretpostavljamo da je graf $G = (V, E)$ zadan s težinskom funkcijom w, tj. da se radi o neusmjerenom, težinskom grafu, čiji je skup vrhova dimenzije m.

Definicija 5.1.3. *Neka je dan graf $G = (V, E)$ i njegova incidencijska matrica S. Za svaki vrh $v_i \in V$ definiramo stupanj tog vrha $d_i = \sum_{j=1}^{m} s_{ij}$, a dijagonalnu matricu $D = diag(d_1, ..., d_m) = diag(S\,\mathbf{e})$ nazivamo matricom stupnjeva.*

Definicija 5.1.4. *Neka je dan graf $G = (V, E)$ i podskup skupa vrhova $B \subseteq V$. Volumen skupa B definiramo sa*

$$volB = \sum_{v_i \in B} d_i \tag{5.2}$$

Definicija 5.1.5. *Definiramo Laplaceovu matricu $L = [l_{ij}] \in \mathbb{R}^{m\times m}$ grafa G sa $L = D - S$, pri čemu su D i S matrice stupnjeva i incidencije redom.*

Navedimo neka od svojstava Laplaceovih matrica (dokazi se mogu pronaći u [43, str. 4-5])

Propozicija 5.1.6. *Laplaceova matrica L ima sljedeća svojstva:*

(i) $\mathbf{x}^T L\mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{m} w_{ij}(x_i - x_j)^2, \forall \mathbf{x} \in \mathbb{R}^m$.

(ii) L je simetrična, pozitivno semi-definitna matrica.

(iii) Najmanja svojstvena vrijednost Laplaceove matrice jednaka je 0*, a pripadni svojstveni vektor je* $\mathbf{e}$.

(iv) L ima m nenegativnih realnih svojstvenih vrijednosti $0 = \lambda_1 \leqslant \lambda_2 \leqslant ... \leqslant \lambda_m$.

Propozicija 5.1.7. *Kratnost svojstvene vrijednosti* 0 *Laplaceove matrice L odgovara broju komponenti povezanosti grafa G.*

5.2 Biparticioniranje grafa

Definicija 5.2.1. *Neka je zadan graf $G = (V, E)$ i $A, B \subseteq V$. Definiramo rez između skupova A i B sa*

$$rez(A, B) = \sum_{v_i \in A, v_j \in B} s_{ij} \tag{5.3}$$

Pretpostavimo da želimo particionirati vrhove grafa G na dva skupa, A i B. Označimo sa $\mathcal{P}_2(V)$ skup svih dvočlanih particija skupa V. Sada kao kriterij particioniranja možemo uzeti minimizaciju funkcije reza

$$\min_{\{A,B\} \in \mathcal{P}_2(V)} \text{rez}(A, B) \tag{5.4}$$

Ovaj problem poznatiji je pod nazivom *mincut* problem.

Neka je $\{A, B\} \in \mathcal{P}_2(V)$. Definirajmo vektor $\chi \in \mathbb{R}^m$ sa

$$\chi_i = \begin{cases} 1 & \text{, ako je } v_j \in A \\ -1 & \text{, ako je } v_j \in B \end{cases} \tag{5.5}$$

Tada je

$$\chi^T D \chi = \sum_{i=1}^{m} d_i \chi_i^2 = \sum_{i,j=1}^{m} s_{ij} = \text{rez}(A, A) + \text{rez}(B, B) + 2\text{rez}(A, B)$$

$$\chi^T S \chi = \sum_{i,j=1}^{m} s_{ij} \chi_i \chi_j = \text{rez}(A, A) + \text{rez}(B, B) - 2\text{rez}(A, B)$$

Dakle,

$$\text{rez}(A, B) = \frac{1}{4} \chi^T (D - S) \chi = \frac{1}{4} \chi^T L \chi \tag{5.6}$$

Fiedlerova vrijednost

Neka su $0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_m$ svojstvene vrijednosti Laplaceove matrice grafa G. Fiedler je 1973. [31] i 1974. [32] dokazao dva bitna svojstva druge po redu najmanje svojstvene vrijednosti:

(i) Ako je G povezan graf tada je λ_2 pozitivna vrijednost, štoviše ako se graf sastoji od k komponenti povezanosti onda je $0 = \lambda_1 = \cdots = \lambda_k$ i $\lambda_{k+1} > 0$.

(ii) Ako je $G' = (V, E')$ podgraf grafa $G = (V, E)$ dobiven uklanjanjem nekih bridova iz E tada je $\lambda_2(G') \leqslant \lambda_2(G)$.

Vrijednost λ_2 nazivamo Fiedlerovom vrijednosti, a pripadni svojstveni vektor ψ_2 Fiedlerovim vektorom. Osim što poznavajući vrijednost λ_2 možemo donijeti zaključak o povezanosti grafa, možemo i o intezitetu povezanosti. Naime, male Fiedlerove vrijednosti sugeriraju postojanje separabilnih podgrafova.

Navedimo još jednu bitnu primjenu Fiedlerovog vektora (vidi [32]). Ako je G povezan graf i $A \subseteq V$ skup koji sadrži sve vrhove $v_j \in V$ za koje je $(\psi_2)_j \geqslant 0$ tada je graf (A^C, E_{A^C}) povezan. Ako je pri tome dodatno $(\psi_2)_j > 0$ tada je i graf (A, E_A) povezan.

Pretpostavimo da u podacima imamo prisutnost k separabilnih klastera. Tada uz pripadno preslagivanje redaka i stupaca matrice S ona postiže skoro blok dijagonalnu strukturu, obzirom da je $s_{ij} > 0$ za vrhove v_i, v_j iz istog klastera i $s_{ij} \approx 0$ za vrhove iz različitih klastera. Štoviše, tada i pripadna Laplaceova matrica ima isto svojstvo. Poznato je da je spektar blok dijagonalnih matrica jednak uniji spektara njenih blokova (vidi [23]). Minimalna svojstvena vrijednost svakog bloka Laplaceove matrice približno je jednaka nuli, tj. $\lambda_1^{(j)} \approx 0, \forall j = 1, ..., k$. Štoviše, k najmanjih svojstvenih vrijednosti Laplaceove matrice grafa G bit će približno jednako nuli, a sljedeća će, u slučaju ispravnosti parametra k biti zanačajno veća.

Biparticioniranje pomoću Fiedlerovog vektora

Pomoću ovih saznanja moguće je pojednostavniti proces biparticioniranja grafa $G = (V, E)$ rješavanjem *mincut* problema. Umjesto da minimiziramo funkciju reza po svim dvočlanim particijama skupa V, koristimo informaciju o bliskosti vrhova iz Fiedlerovog vektora ψ_2. Naime, neka je zadan parametar $\tau \in \langle -1, 1\rangle$ kojeg nazivamo *threshold* i definirajmo funkciju $\chi : \langle -1, 1\rangle \to \{-1, 1\}^m$ sa

$$\chi(\tau) = \text{sign}(\psi_2 - \tau) \tag{5.7}$$

Ovime smo na početnu preporuku grupiranja vrhova u ovisnosti o predznaku pripadne komponente Fiedlerovog vektora uveli modifikaciju kojom donosimo odluku za rubne slučajeve, one za koje je ta vrijednost blizu nuli. Dakle, radi se o problemu optimizacije

$$\min_{\tau \in \langle -1,1\rangle} \chi^T(\tau) L \chi(\tau) \tag{5.8}$$

Nakon što pronađemo optimalni τ klastere $\{C_\tau, C_\tau^C\}$ definiramo sa

$$C_\tau = \{v_i \in V : \chi_i(\tau) \geqslant 0\} \tag{5.9}$$

Algoritam 14 Bisekcijski spektralni algoritam

Ulazni parametri: Skup podataka $\mathcal{X}$ i matrica relacija W
Izlazni parametri: Segmentacija $\{C_1, C_2\}$

1: Iz matrice relacija konstruiraj matricu incidencije S pripadnog grafa G i izračunaj matricu stupnjeva $D = \text{diag}(S\,\mathbf{e})$ te pripadnu Laplaceovu matricu $L = D - S$.
2: Odredi svojstvene vrijednosti i svojstvene vektore matrice L.
3: Odredi rješenje problema minimizacije (5.8).
4: Kreiraj biparticiju $\{C_1, C_2\}$=$\{C_\tau, C_\tau^C\}$ pravilom $C_\tau = \{v_i \in V : \chi_i(\tau) \geqslant 0\}$.

5.3 k-particioniranje grafa

Tolliver i Miller u svom radu iz 2006. [15] predlažu *top-down* generalizaciju algoritma 14 kojom je moguće particionirati graf G na k klastera. Grafove iterativno biparticioniramo u svakom koraku birajući graf G' s najmanjom Fiedlerovom vrijednosti $\lambda_2(G')$ kao sljedeći na kojeg primjenjujemo postupak. Mane ovog pristupa su to što ishod ovisi o prvim koracima te orijentirajući se na samo jednu svojstvenu vrijednost gubi neke potencijalno korisne informacije pohranjene u preostalim svojstvenim vrijednostima.

Definicija 5.3.1. *Neka je zadan graf $G = (V, E)$ i particija skupa V, $\mathcal{S} = \{C_1, ..., C_k\}$. Njen rez definiramo sa*

$$rez(\mathcal{S}) = \sum_{j=1}^{k} rez(C_j, C_j^C) \tag{5.10}$$

Radi se o generaliziranom rezu skupova. Analogno, kao u slučaju biparticioniranja, želimo minimizirati navedenu vrijednost po skupu svih k-članih particija skupa V. Međutim, u praksi se pokazalo da minimizacija ovog izraza ne uzima u obzir veličine klastera, pa često završimo sa nebalansiranim klasterima ([18]).

Kako bi se izbjegla prethodno navedena situaciju, uvode se razna poboljšanja tog kriterija u formi

$$J(C_1, ..., C_k) = \sum_{j=1}^{k} \frac{\text{cut}(C_j, C_j^C)}{f(C_j)} \tag{5.11}$$

Najpoznatija među njima:

(i) Ding, He, Zha, Gu i Simon u radu iz 2001. [9] predlažu korištenje min-max reza

$$\text{Mcut}(C_1, ..., C_k) = \sum_{j=1}^{k} \frac{\text{cut}(C_j, C_j^C)}{\text{cut}(C_j, C_j)} \tag{5.12}$$

(ii) Shi i Malik u radu iz 2002. [27] predlažu korištenje normaliziranog reza

$$\text{Ncut}(C_1, ..., C_k) = \sum_{j=1}^{k} \frac{\text{cut}(C_j, C_j^C)}{\text{vol}C_j} \tag{5.13}$$

(iii) Hagen i Kahng u radu iz 2006. [29] predlažu korištenje razmjernog reza

$$\text{Rcut}(C_1, ..., C_k) = \sum_{j=1}^{k} \frac{\text{cut}(C_j, C_j^C)}{|C_j|} \tag{5.14}$$

Spektralna relaksacija reza

Definirajmo vektore $\mathbf{h}_1, ..., \mathbf{h}_k \in \mathbb{R}^m$, $\mathbf{h}_j = (h_{1j}, ..., h_{mj})$, sa

$$h_{ij} = \begin{cases} \sqrt{|C_j|}^{-1} & \text{, ako je } i \in C_j \\ 0 & \text{, inače} \end{cases} \tag{5.15}$$

i pripadnu matricu $H = \left[\mathbf{h}_1 \cdots \mathbf{h}_k\right]$

Obzirom da je $\|\mathbf{h}_i\| = \mathbf{h}_i^T \mathbf{h}_i = 1$ i $\mathbf{h}_i^T \mathbf{h}_j = 0, \forall i \neq j$ tada vrijedi $H^T H = \mathbb{I}$. Štoviše, vrijedi

$$\mathbf{h}_j^T L \mathbf{h}_j = (H^T L H)_{jj} = \frac{1}{2} \sum_{p=1}^{m} \sum_{q=1}^{m} s_{pq}(h_{pj} - h_{qj})^2 = \frac{\text{cut}(C_j, C_j^C)}{|C_j|} \tag{5.16}$$

tj.

$$\text{Rcut}(C_1, ..., C_k) = \sum_{i=1}^{k} \mathbf{h}_i^T L \mathbf{h}_i = \sum_{i=1}^{k} (H^T L H)_{ii} = tr(H^T L H) \tag{5.17}$$

Drugim riječima cilj je

$$\min_{H^T H = \mathbb{I}} tr(H^T L H) \tag{5.18}$$

Navedimo bez dokaza teorem kojeg možemo iskoristiti za rješenje navedenog problema.

Teorem 5.3.2. *(Ky Fan) Neka je $M \in \mathbb{R}^{m \times m}$ simetrična matrica sa svojstvenim vrijednostima $0 \leqslant \lambda_1 \leqslant ... \leqslant \lambda_m \in \mathbb{R}$ i pripadnim svojstvenim vektorima $\psi_1, ..., \psi_m$. Neka je $U \in \mathbb{R}^{m \times k}$,$1 \leqslant k \leqslant m$ unitarna matrica. Tada je rješenje problema*

$$Y = arg \min_{U^T U = \mathbb{I}} tr(U^T M U) \tag{5.19}$$

matrica $Y = (\psi_1, ..., \psi_k)Q$ pri čemu je $Q \in \mathbb{C}^{k \times k}$ unitarna matrica.

Prema teoremu 5.3.2 minimalna vrijednost ove funkcije postiže se u matrici $Y = (\psi_1, ..., \psi_k)R$. U praksi se najčešće pretpostavlja da je $R = \mathbb{I}$. Retke matrice Y možemo promatrati kao spektralne koordinate vrhova grafa sličnosti.

Algoritam 15 Algoritam spektralne k segmentacije

Ulazni parametri: Skup podataka $\mathcal{X}$ i matrica relacija W
Izlazni parametri: Segmentacija $\{C_1, ..., C_k\}$

1: Iz matrice relacija konstruiraj matricu incidencije S pripadnog grafa G i izračunaj matricu stupnjeva $D = \text{diag}(S\,\mathbf{e})$ te pripadnu Laplaceovu matricu $L = D - S$.
2: Odredi svojstvene vrijednosti i svojstvene vektore matrice L.
3: Definiraj $Y = \begin{bmatrix}\psi_1 \cdots \psi_k\end{bmatrix}$ pri čemu su $\psi_1, ..., \psi_k$ svojstveni vektori pridruženi svojstvenim vrijednostima $0 \leqslant \lambda_1 \leqslant \cdots \leqslant \lambda_k \leqslant \cdots$.
4: Promatrajući matricu Y kao m-dimenzionalan skup podataka sa k komponenti, segmentiraj njene retke u k klastera.

Preostaje odrediti segmentaciju matrice/skupa podataka Y. U ovom koraku najčešće se koristi *k-means* algoritam ili *QR* dekompozicija s pivotiranjem koju su u svom radu iz 2002. predložili Zha, He, Ding, Simon i Gu [19].

Algoritam 16 Korištenje *QR* dekompozicije za segmentiranje matrice Y

Ulazni parametri: $Y \in \mathbb{R}^{m\times k}$
Izlazni parametri: Vektor $\mathbf{c} = (c_1, ..., c_m)$ koji označava pripadnost svakog od vrha jednom od k klastera

1: Odredi ortogonalnu matricu $Q \in \mathbb{R}^{k\times k}$, gornjetrokutastu matricu $R \in \mathbb{R}^{k\times m}$ i permutacijsku matricu $P \in \mathbb{R}^{m\times m}$ takve da je $Y^T P = QR$.
2: Izačunaj $U = [u_{ji}]_{k\times m}$ formulom $U = R(1:k, 1:k)^{-1} RP^T$.
3: Za svaki $i = 1, ..., m$ pridruži $c_i \leftarrow \arg\max_{j=1,...,k} |u_{ji}|$.

Poglavlje 6

Primjena

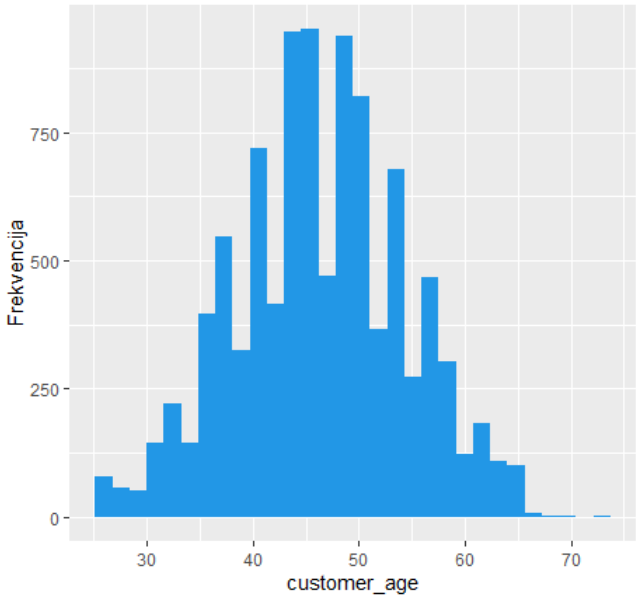
U ovom poglavlju dajemo primjer segmentacije korisnika kreditne kartice jedne banke. Korišteni skup podataka dostupan je na [2]. Za analizu koristimo programsko okruženje **RStudio** i programski jezik **R**. Od korištenih paketa istaknimo **tidyverse** [3], grupu paketa dizajniranu za korištenje u *data science* projektima i grupu paketa **tidymodels** [4] dizajniranu za korištenje pri modeliranju i metodama strojnog učenja.

6.1 Pregled podataka

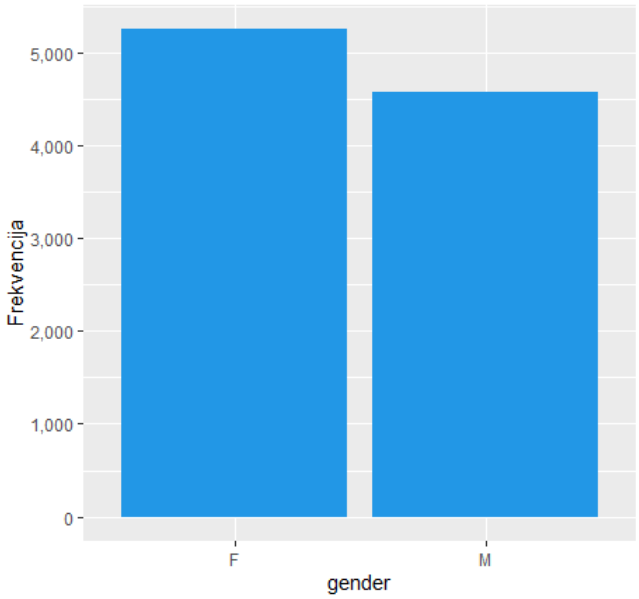
```
> data %>% dplyr::glimpse()
Rows: 9,831
Columns: 21
$ clientnum                <dbl> 768805383, 818770008, 713982108, 769911858, 709106358, 713061558, 810347208, 818906208, 710
$ attrition_flag           <chr> "Existing Customer", "Existing Customer", "Existing Customer", "Existing Customer", "Existi
$ customer_age             <dbl> 45, 49, 51, 40, 40, 44, 51, 32, 37, 48, 42, 65, 56, 35, 57, 44, 48, 41, 61, 45, 47, 62, 41,
$ gender                   <chr> "M", "F", "M", "F", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "F", "M", "M", "M", "
$ dependent_count          <dbl> 3, 5, 3, 4, 3, 2, 4, 0, 3, 2, 5, 1, 1, 3, 2, 4, 4, 3, 1, 2, 1, 0, 3, 4, 2, 3, 1, 1, 3, 4, 3
$ education_level          <chr> "High School", "Graduate", "Graduate", "High School", "Uneducated", "Graduate", "Unknown",
$ marital_status           <chr> "Married", "Single", "Married", "Unknown", "Married", "Married", "Married", "Unknown", "Sin
$ income_category          <chr> "$60K - $80K", "Less than $40K", "$80K - $120K", "Less than $40K", "$60K - $80K", "$40K - $
$ card_category            <chr> "Blue", "Blue", "Blue", "Blue", "Blue", "Blue", "Gold", "Silver", "Blue", "Blue", "Blue", "
$ months_on_book           <dbl> 39, 44, 36, 34, 21, 36, 46, 27, 36, 36, 31, 54, 36, 30, 48, 37, 36, 34, 56, 37, 42, 49, 33,
$ total_relationship_count <dbl> 5, 6, 4, 3, 5, 3, 6, 2, 5, 6, 5, 6, 3, 5, 5, 5, 6, 4, 2, 6, 5, 2, 4, 3, 4, 6, 4, 3, 5, 6, 3
$ months_inactive_12_mon   <dbl> 1, 1, 1, 4, 1, 1, 1, 2, 2, 3, 3, 2, 6, 1, 2, 1, 2, 4, 2, 1, 2, 3, 2, 3, 2, 1, 1, 3, 2, 0, 2
$ contacts_count_12_mon    <dbl> 3, 2, 0, 1, 0, 2, 3, 2, 0, 3, 2, 3, 0, 3, 2, 2, 3, 1, 3, 2, 0, 3, 1, 2, 3, 2, 2, 2, 2, 0, 3
$ credit_limit             <dbl> 12691.0, 8256.0, 3418.0, 3313.0, 4716.0, 4010.0, 34516.0, 29081.0, 22352.0, 11656.0, 6748.0
$ total_revolving_bal      <dbl> 777, 864, 0, 2517, 0, 1247, 2264, 1396, 2517, 1677, 1467, 1587, 0, 1666, 680, 972, 2362, 12
$ avg_open_to_buy          <dbl> 11914.0, 7392.0, 3418.0, 796.0, 4716.0, 2763.0, 32252.0, 27685.0, 19835.0, 9979.0, 5281.0,
$ total_amt_chng_q4_q1     <dbl> 1.335, 1.541, 2.594, 1.405, 2.175, 1.376, 1.975, 2.204, 3.355, 1.524, 0.831, 1.433, 3.397,
$ total_trans_amt          <dbl> 1144, 1291, 1887, 1171, 816, 1088, 1330, 1538, 1350, 1441, 1201, 1314, 1539, 1311, 1570, 13
$ total_trans_ct           <dbl> 42, 33, 20, 20, 28, 24, 31, 36, 24, 32, 42, 26, 17, 33, 29, 27, 27, 21, 30, 21, 27, 16, 18,
$ total_ct_chng_q4_q1      <dbl> 1.625, 3.714, 2.333, 2.333, 2.500, 0.846, 0.722, 0.714, 1.182, 0.882, 0.680, 1.364, 3.250,
$ avg_utilization_ratio    <dbl> 0.061, 0.105, 0.000, 0.760, 0.000, 0.311, 0.066, 0.048, 0.113, 0.144, 0.217, 0.174, 0.000,
```

Slika 6.1: Pregled podataka

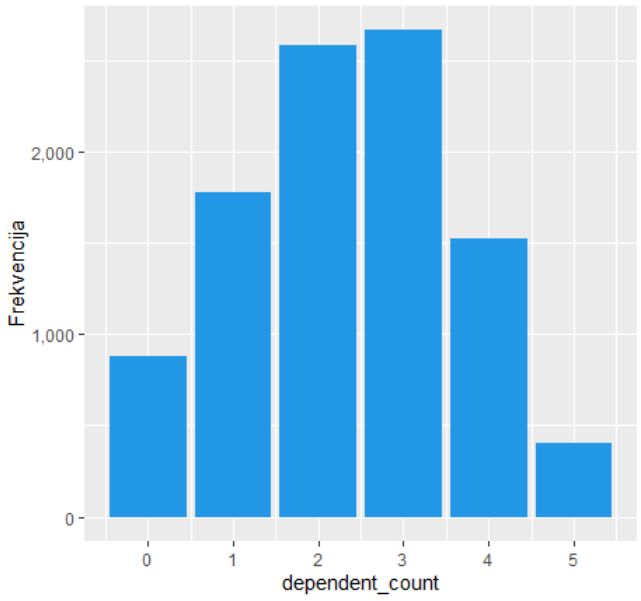
1. **customer_age** (dob)



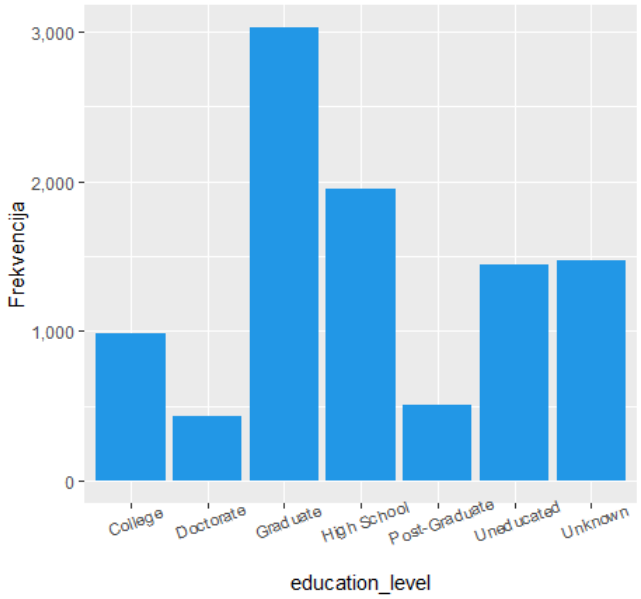
2. **gender** (spol)



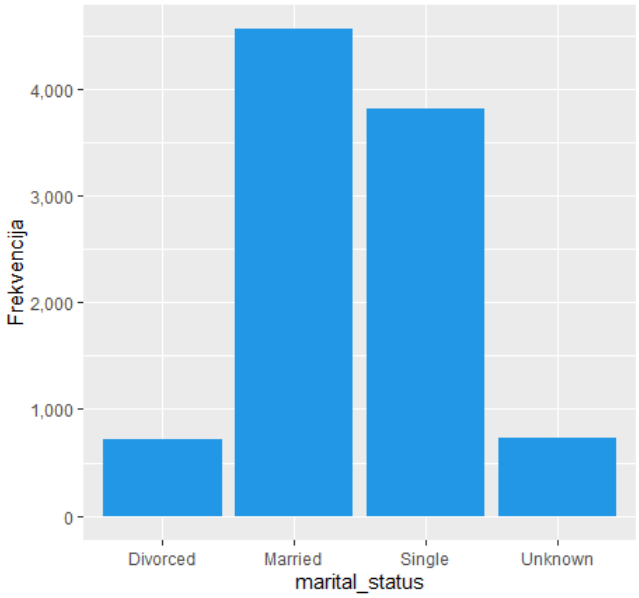
3. **dependent_count** (broj uzdržavanih osoba)



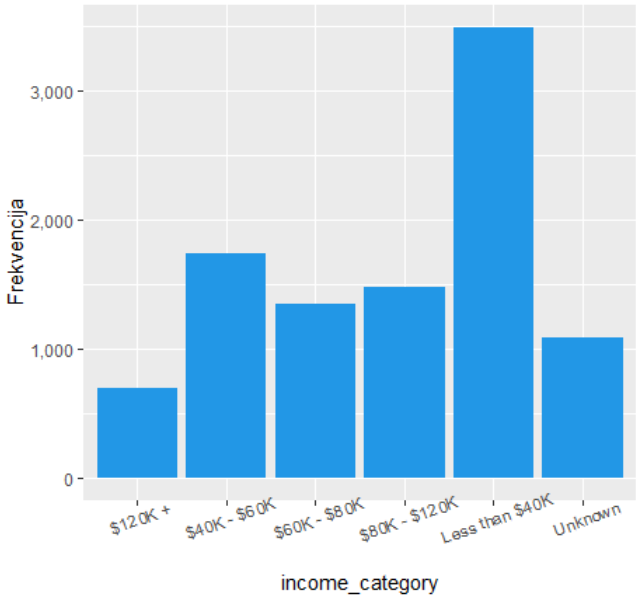
4. **education_level** (razina obrazovanja)



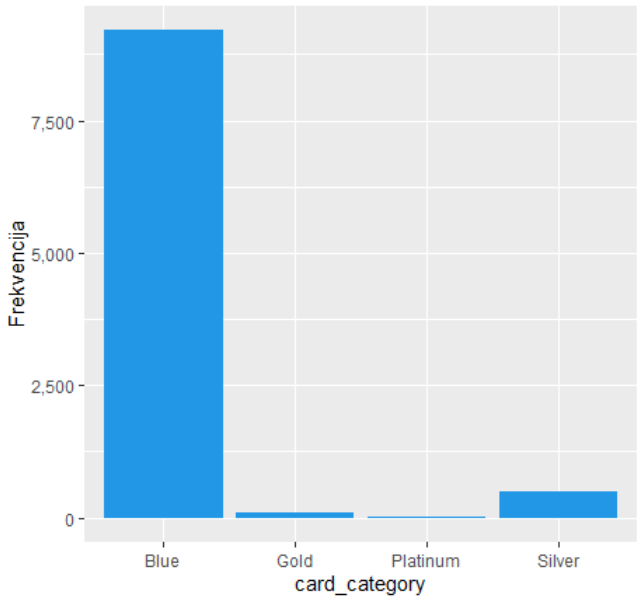
5. **marital_status** (bračni status)



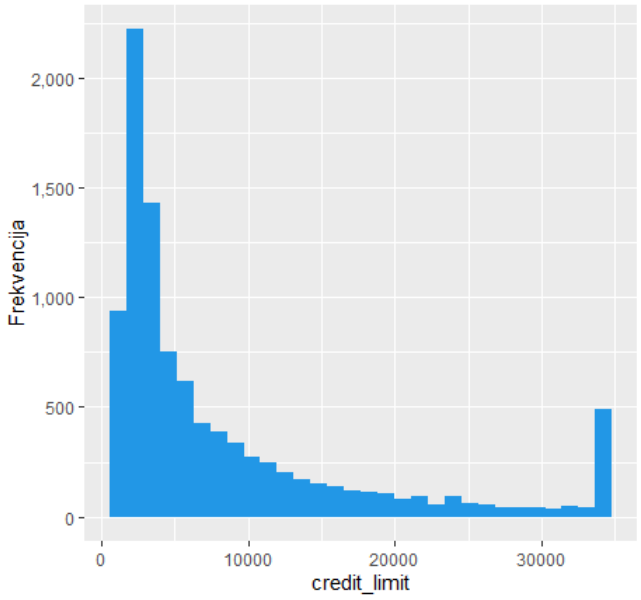
6. **income_category** (iznos prihoda)



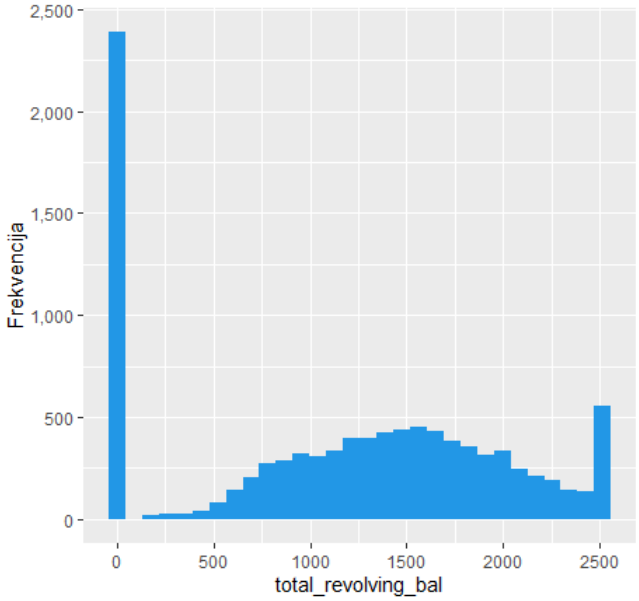
7. **card_category** (tip kreditne kartice)



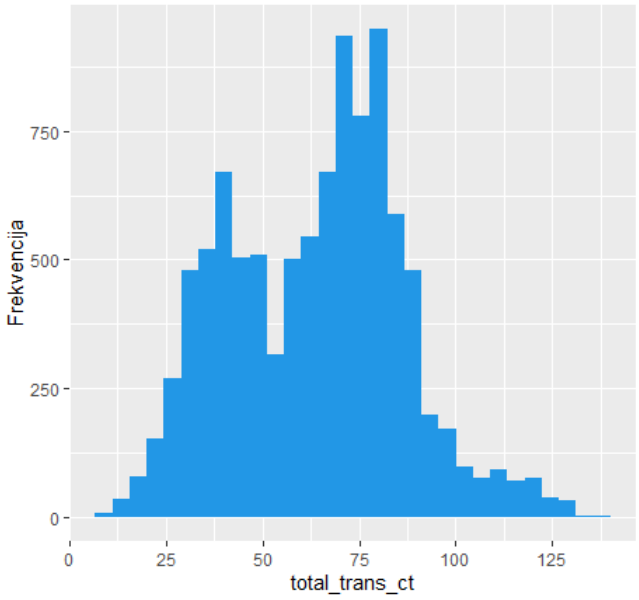
8. **credit_limit** (limit kreditne kartice)



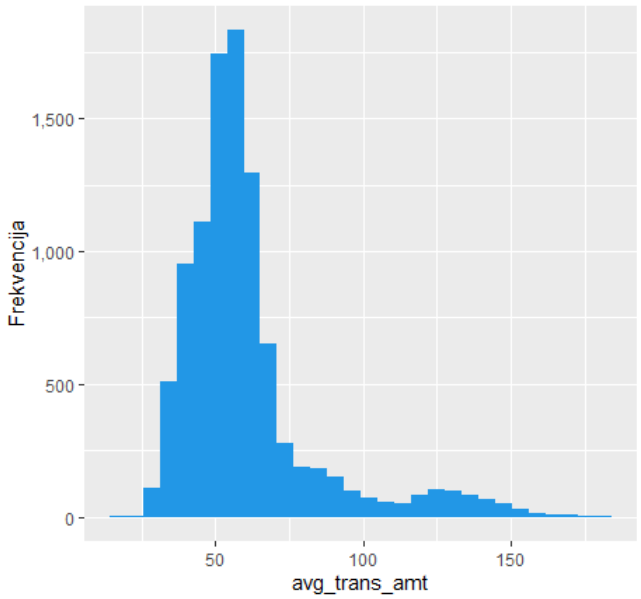
9. **total_revolving_bal** (ukupan iznos revolvinga)



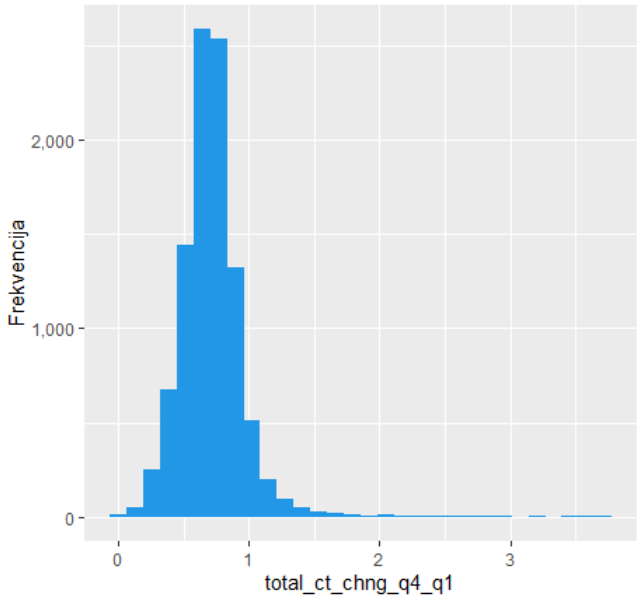
10. **total_trans_ct** (ukupan broj transakcija)



11. **avg_trans_amt** (prosječan iznos transakcije)



12. **total_ct_chng_q4_q1** (kvartalna promjena broja transakcija)



6.2 Priprema podataka

U sljedećem koraku transformirajmo kategorijske varijable (komponente):

1. Varijable **education_level**, **income_category** i **card_category** imaju svojstvo ordinalne varijable, stoga ih transformiramo u numeričku varijablu na jednostavan način. Za prvu varijablu postupak je sljedeći. Najnižu vrijednosti, *Uneducated*, zamijenimo sa 1, a najvišu vrijednost, *Doctorate*, sa 6. Vrijednost *Unknown* zamijenimo prosjekom svih novih vrijednosti u skupu za koje nam je poznata razina obrazovanja.

2. Varijable **marital_status** i **gender** transformiramo koristeći *dummy encoding* proceduru. Naime, pod pretpostavkom da varijabla poprima m različitih vrijednosti, kreiramo $m-1$ indikatorsku varijablu. Na taj način, svaka kategorija, osim jedne, predstavljena je svojom varijablom, a svaki podatak može imati najviše jednu jedinicu u novih $m-1$ varijabli. U slučaju bračnog statusa kreiramo dvije nove varijable, **marital_status_single** i **marital_status_married**. Dakle, referentna vrijednost nam je *Divorced* i podaci koji primaju tu vrijednost u novim varijablama će imati obje 0. Podatke sa statusom *Unknown* ponovno mijenjamo vrijednostima prosjeka u novim varijablama.

Nakon toga, sve varijable standardiziramo (1.18), kako bi uklonili utjecaj različitih raspona po varijablama na konačan ishod.

```
> data %>% glimpse()
Rows: 9,831
Columns: 14
$ clientnum              <chr> "768805383", "818770008", "713982108", "769911858", "709106358", "713061558"...
$ customer_age           <dbl> -0.17030724, 0.32948096, 0.57937506, -0.79504249, -0.79504249, -0.29525429, ...
$ gender_female          <dbl> -1.0715718, 0.9331137, -1.0715718, 0.9331137, -1.0715718, -1.0715718, -1.071...
$ dependent_count        <dbl> 0.5053885, 2.0476744, 0.5053885, 1.2765314, 0.5053885, -0.2657545, 1.2765314...
$ education_level        <dbl> -0.81410479, 0.72181016, 0.72181016, -0.81410479, -1.58206226, 0.72181016, 0...
$ marital_status_married <dbl> 1.036314, -1.042482, 1.036314, 0.000000, 1.036314, 1.036314, 1.036314, 0.000...
$ marital_status_single  <dbl> -0.8829737, 1.2235228, -0.8829737, 0.0000000, -0.8829737, -0.8829737, -0.882...
$ income_category        <dbl> 0.5233469, -1.0427522, 1.3063965, -1.0427522, 0.5233469, -0.2597026, 2.08944...
$ card_category          <dbl> -0.2411389, -0.2411389, -0.2411389, -0.2411389, -0.2411389, -0.2411389, 5.94...
$ credit_limit           <dbl> 0.469453641, -0.023894926, -0.562073138, -0.573753318, -0.417683862, -0.4962...
$ total_revolving_bal    <dbl> -0.475111290, -0.368219095, -1.429769169, 1.662732607, -1.429769169, 0.10235...
$ total_trans_ct         <dbl> -0.9597436, -1.3557072, -1.9276545, -1.9276545, -1.5756869, -1.7516707, -1.4...
$ avg_trans_amt          <dbl> -1.42069461, -0.91198365, 1.45233624, -0.08024587, -1.33915260, -0.64604550,...
$ total_ct_chng_q4_q1    <dbl> 3.80003767, 12.48403219, 6.74320096, 6.74320096, 7.43742168, 0.56172664, 0.0...
```

Slika 6.2: Pregled transformiranih podataka

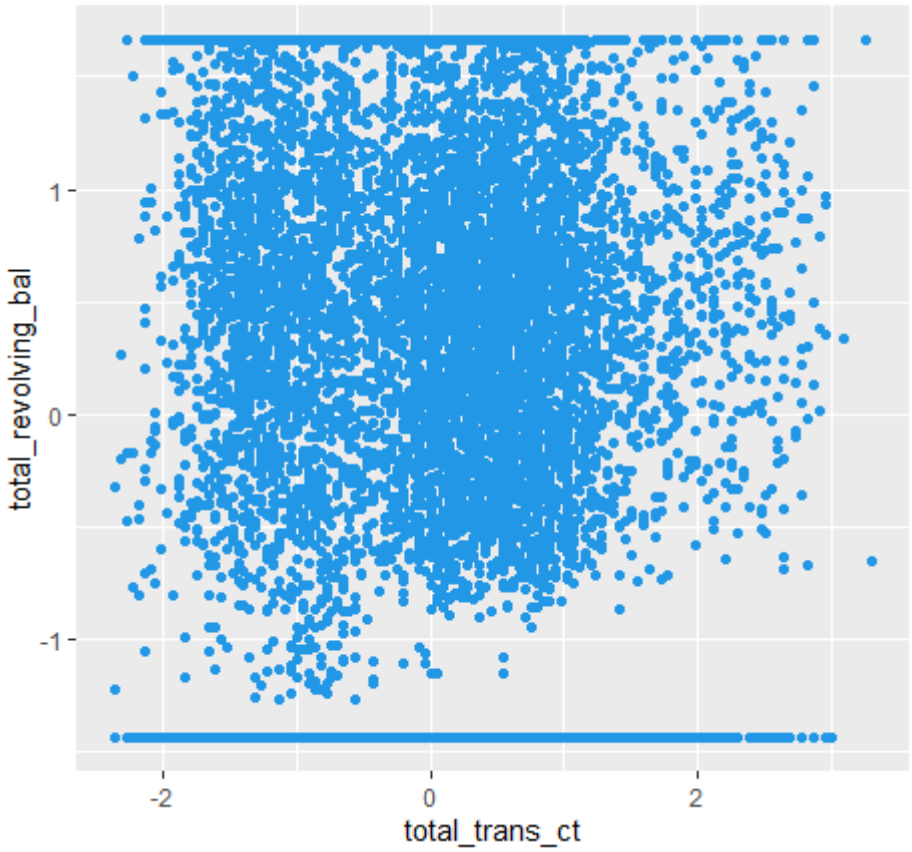
6.3 Primjena algoritma

Primijenimo *k-means* algoritam na ovom skupu podataka. Koristimo funkciju *kmeans* dostupnu pomoću paketa **stats**, sa sljedećim parametrima:

1. *algorithm* = Hartigan-Wong (inicijalizacijska metoda (ii))
2. *iter.max* = 100 (maksimalan broj iteracija)
3. *nstart* = 25 (broj ponavljanja)

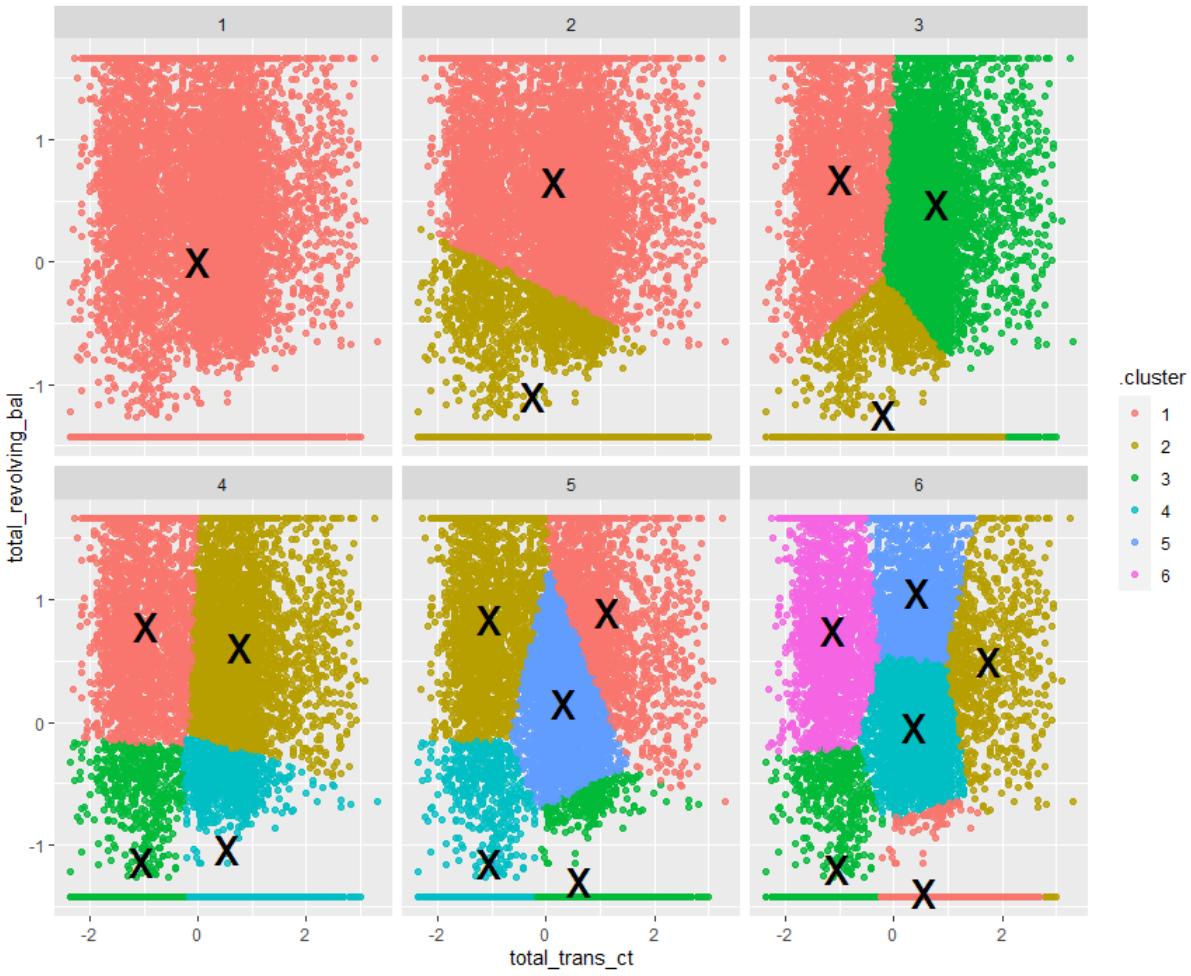
Dvije varijable

U prvom slučaju pokušajmo segmentirati podatke koristeći samo dvije varijable, **total_trans_ct** i **total_revolving_bal**.

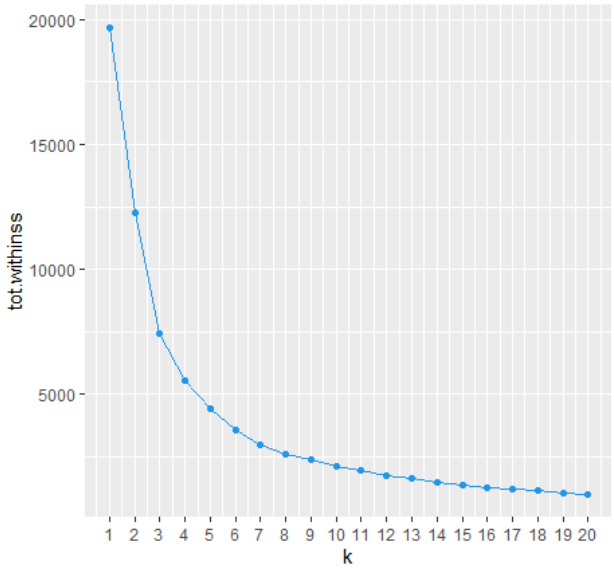


Slika 6.3: Scatter plot za dvije varijable

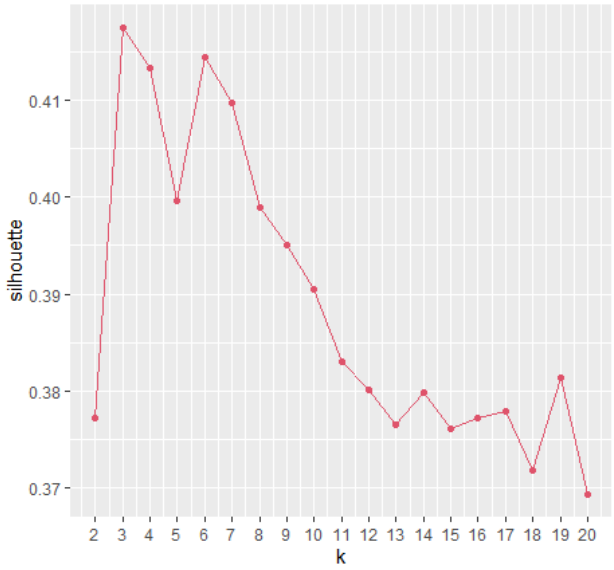
Iz ove vizualizacije nije jasno koliko bi klaster bilo prigodno odabrati, stoga provedimo algoritam za vrijednosti $k = 1, ..., 20$, te prvih 6 prikažimo grafički.



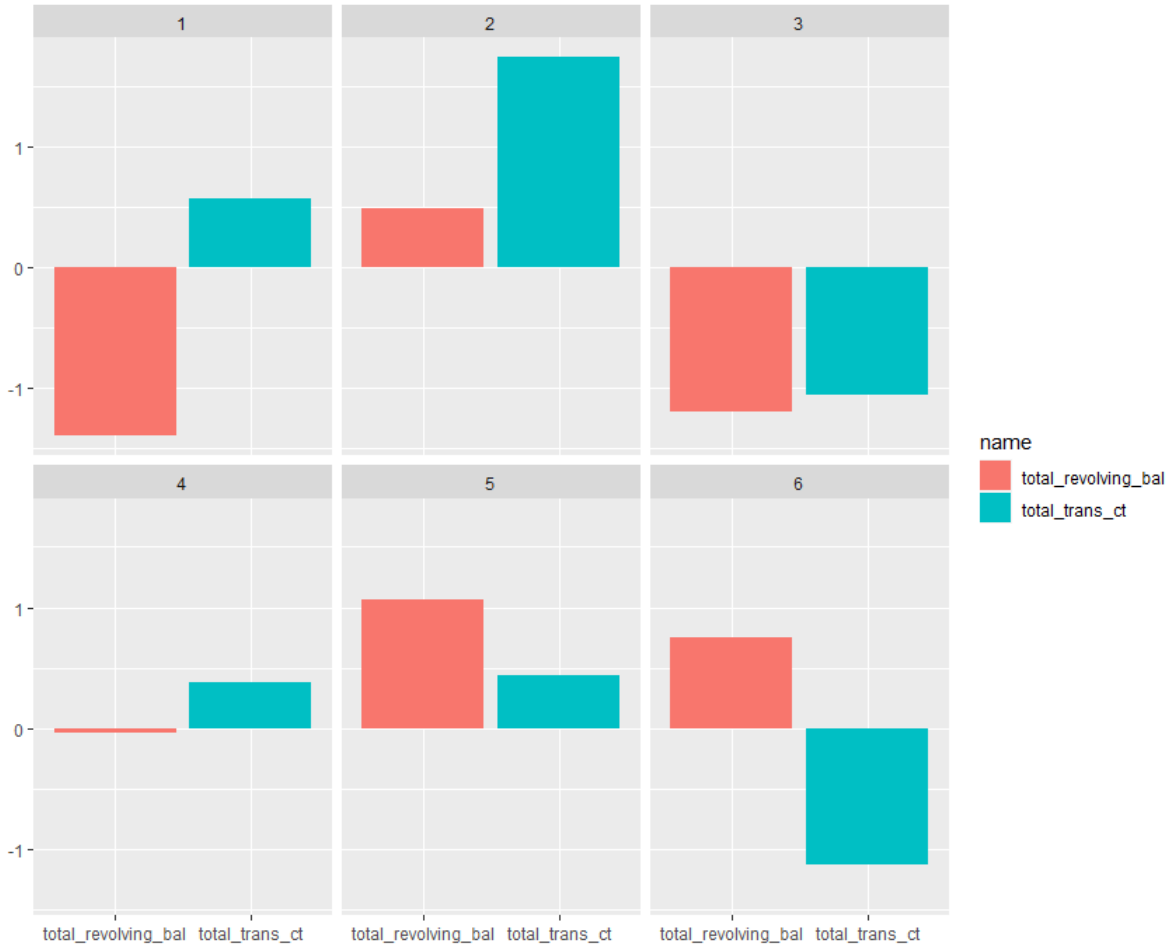
Vizualno svih 6 segmentacija ima smisla. Provjerimo kako se ponaša funkcija cilja u odnosu na k.



Primjenjući *elbow* metodu, vidimo da se lakat pojavljuje negdje oko vrijednosti $k = 5$. Za donošenje odluke, promotrimo vrijednosti *silhouette* koeficijenta.



Vrijednost $k = 5$ ima manju vrijednost navedenog koeficijenta od svojih susjeda, dakle izgleda kako je prikladnije koristiti $k = 4$ ili $k = 6$. Pogledajmo centre klastera za vrijednost $k = 6$.

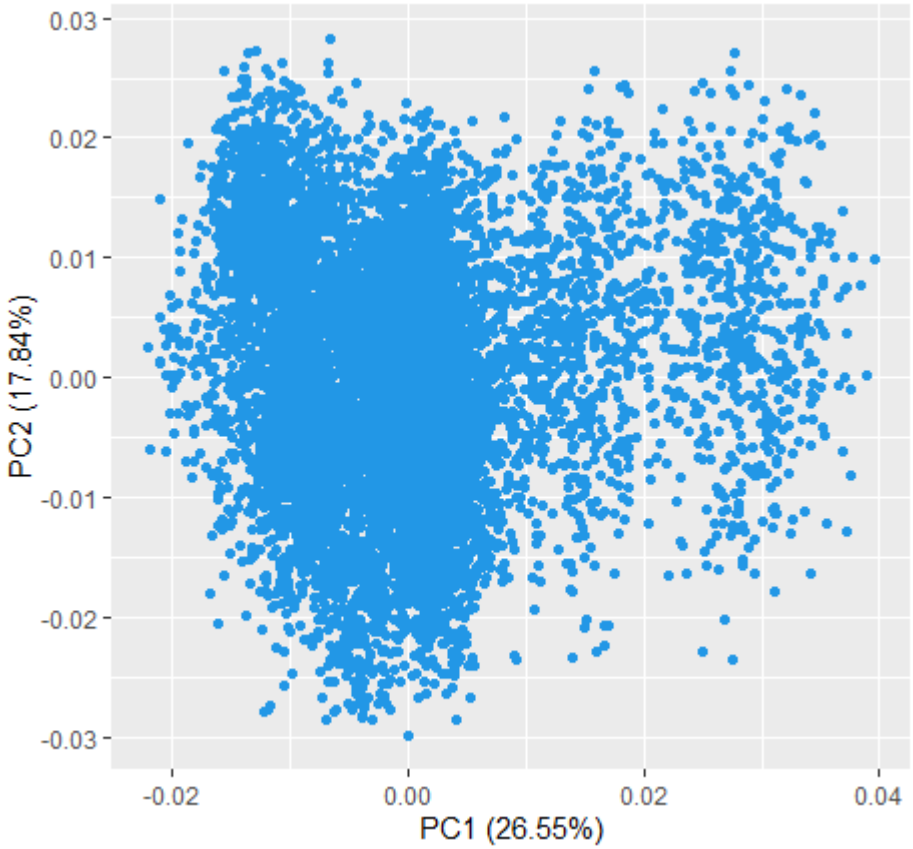


Iz ovog prikaza možemo zaključiti sljedeće. Prvi i treći klasteri sadrže klijente sa manjim iznosom revolvinga, a razlikuju se po ukupnom broju transakcija. Četvrti se ne razlikuje po iznosu revolvinga od prosjeka populacije, ali sadrži klijente sa nešto većim brojem transakcija. Šesti klaster sadrži klijente sa većim iznosom revolvinga, i značajno manjim ukupnim brojem transkacija. Drugi i peti klasteri sadrže klijente sa većim iznosom revolvinga i većim ukupnim brojem transkacija, pri čemu se kod drugog vidi značajn utjecaj onih sa visokim brojem transakcija, a kod petog značajan utjecaj klijenata sa višim iznosom revolvinga.

Nadalje, veličine klastera su redom 1398, 818, 1498, 2309, 1786 i 2022, što daje zadovoljavajuću razinu balansiranosti.

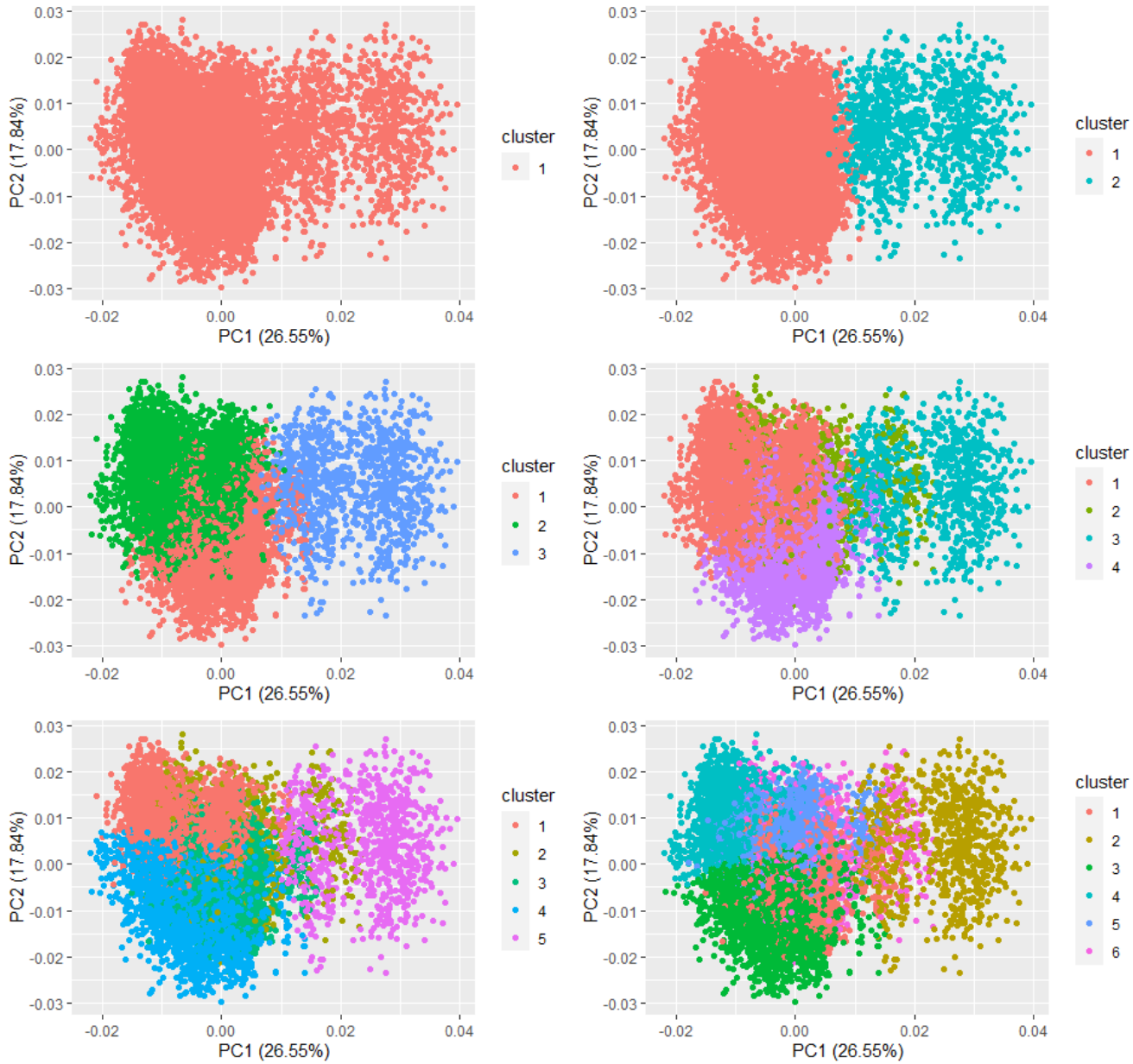
Šest varijabli

Ponovimo cijeli postupak dodajući još 4 varijable, **credit_limit**, **customer_age**, **marital_status_married** i **avg_trans_amt**. Obzriom da koristimo više od 2 varijable, za vizualizaciju koristimo analizu glavnih komponenti (eng. *principal components analysis*). Naših 6 komponenti svodimo na dvije linearne kombinacije istih, pokušavajući zadržati što više informacija o koreliranosti. Detalji ovog pristupa mogu se pronaći u [5].

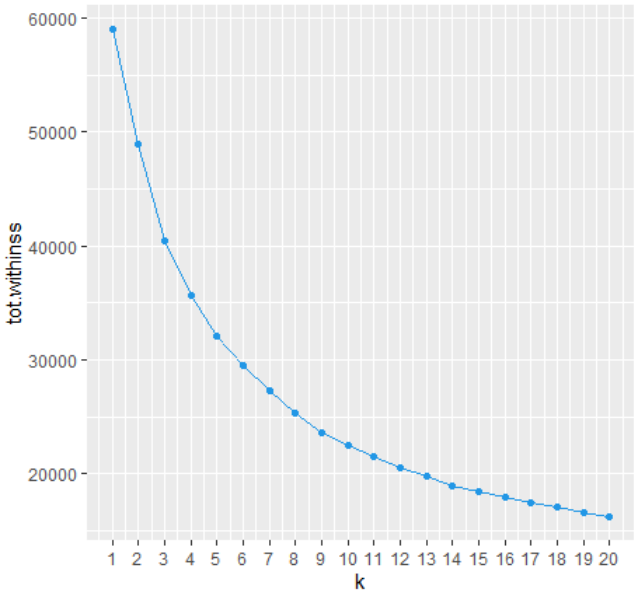


Slika 6.4: Scatter plot za 6 varijabli (PCA)

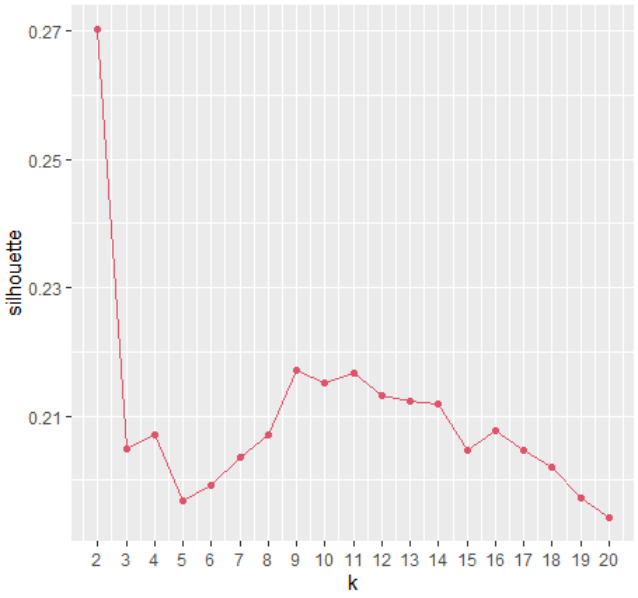
Kao i u slučaju dvije varijable, vizualizacija nam ne daje jasan odgovor na pitanje koliko bi klaster bilo prigodno odabrati.



Vizualno izgleda da već za vrijednost $k = 4$ nije moguće pronaći jasnu segmentaciju. Ipak, obzirom na reduciranost ovog prikaza, pogledajmo vrijednost funkcije cilja u ovisnosti o k.

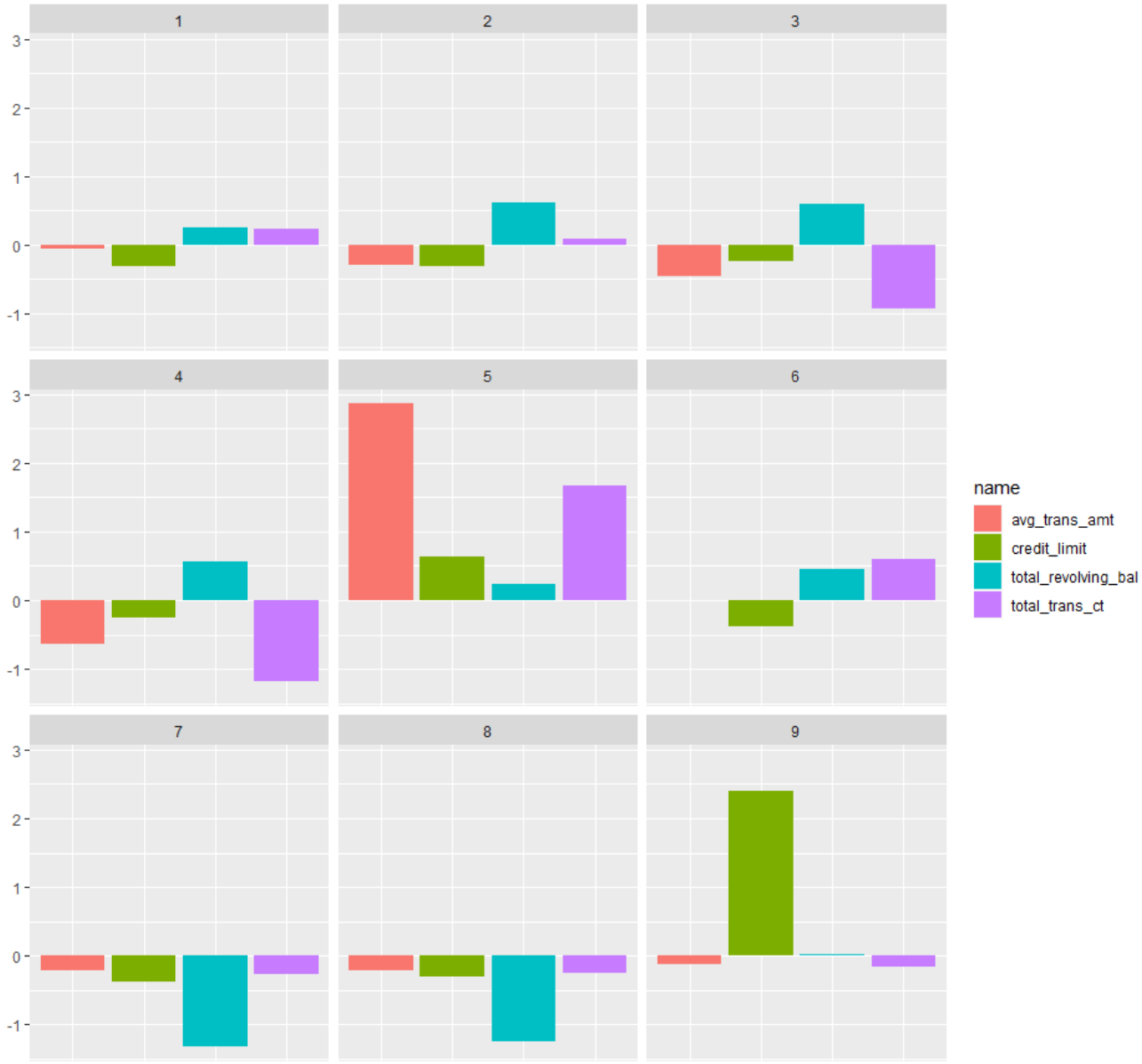


Ovog puta pozicija lakta nije jednostavno uočljiva, iako se naslučuje da bi prikladno bilo odabrati neku od vrijednosti $k = 5, 6, 7, 8, 9$. Promotrimo *shilouette* koeficijent.



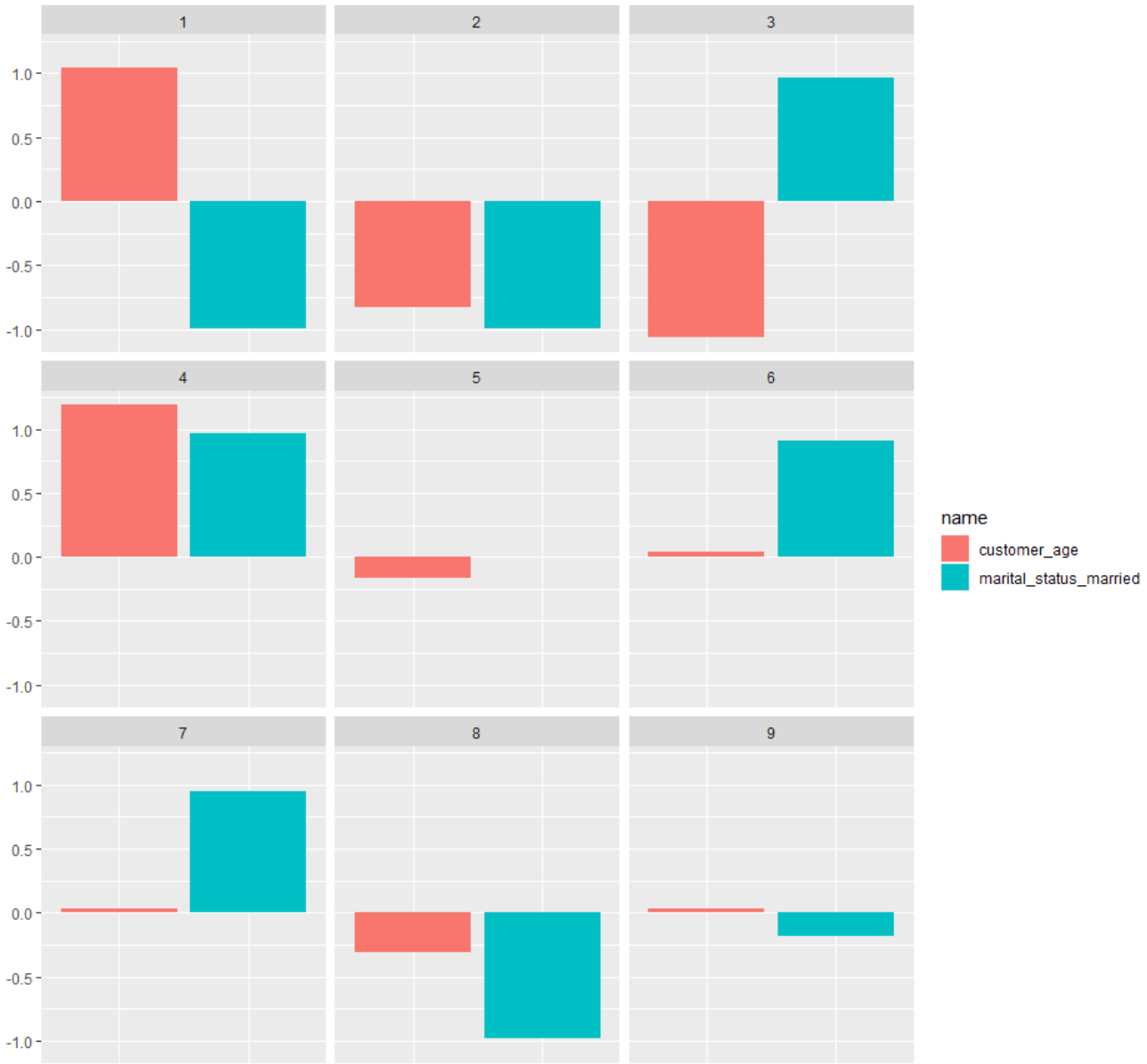
Vidimo da nam navedeni koeficijent indicira vrijednost $k = 9$ kao prikladan odabir.

Promotrimo centre u tom slučaju. Obzirom da sada radimo sa 6 varijabli, podijelimo ih na one koje daju informacije o ponašanju klijenta i one koje daju neke demografske podatke.



Promatrajući prvu grupu varijabli, možemo zaključiti sljedeće. Značajno više razine prosječnog iznosa transakcije javljaju se u petom klasteru, a u četvrtom su one nešto niže

od prosjeka populacije. Limit kreditne kartice kao značajn faktor pojavljuje se u devetom klasteru. Sedmi i osmi klaster karakteriziraju niske razine revolving iznosa, a ukupan broj transakcija je niži u trećem i četvrtom klasteru, te viši u petom.



U drugoj grupi varijabli vidimo značajnije razlike. Promatramo li dob klijenta, prvi

i četvrti klaster sadrži stariju populaciju, a drugi i treći mlađu. Po bračnom statusu prvi, drugi i osmi kalster sadrže neoženjene klijente, a treći, četvrti, šesti i sedmi značajniju populaciju oženjenih.

Veličine klastera su redom 1294, 1399, 788, 866, 712, 1517, 1115, 1228 i 912, što daje zadovoljavajuću razinu balansiranosti. Pokušajmo sada, zbog jednostavnosti i jasnije slike, opisati klastere pomoću nekih karakterističnih svojstava njihovih centara.

1. Stariji, neoženjeni klijenti.
2. Mlađi, neoženjeni klijenti sa nešto višim iznosom revolvinga.
3. Mlađi, oženjeni klijenti sa nešto višim iznosom revolvinga i nižim ukupnim brojem transakcija.
4. Stariji, oženjeni klijenti sa nešto nižim prosječnim transakcijama, manjim brojem istih i nešto višim iznosom revolvinga.
5. Klijenti sa značajno višim prosječnim transakcijama, većim brojem istih i nešto većim limitom na kreditnoj kartici.
6. Oženjeni klijenti, sa nešto većim brojem transakcija i nešto višim iznosom revolvinga.
7. Oženjeni klijenti sa nižim iznosom revolvinga.
8. Neoženjeni klijenti sa nižim iznosom revolvinga.
9. Klijenti sa značajno većim limitom na kreditnoj kartici.

Dodatak A

Kodovi u R-u

```
# Ucitavanje paketa
library(tidyverse)
library(ggfortify)

# Ucitavanje podataka
data <- readr::read_csv("BankChurners.csv")
data <- data %>% dplyr::select(1:21) %>%
  stats::setNames(., colnames(.) %>% janitor::make_clean_names()) %>%
  stats::na.omit() %>%
  dplyr::mutate(
    avg_trans_amt = total_trans_amt/total_trans_ct,
    clientnum = as.character(clientnum))

# Ispis pregleda podataka
data %>% dplyr::glimpse()

# Vizualizacije po varijablama

## dob
data %>% ggplot2::ggplot(aes(x = customer_age)) +
  ggplot2::geom_histogram(fill = 4) +
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)}) +
  ggplot2::scale_x_continuous(n.breaks = 6)

## spol
data %>% ggplot2::ggplot(aes(x = gender)) +
  ggplot2::geom_bar(fill = 4) +
```

```
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)})

## broj uzdrzavanih osoba
data %>%
  ggplot2::ggplot(aes(x = dependent_count)) +
  ggplot2::geom_bar(fill = 4) +
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)}) +
  ggplot2::scale_x_continuous(n.breaks = 6)

## razina obrazovanja
data %>%
  ggplot2::ggplot(aes(x = education_level)) +
  ggplot2::geom_bar(fill = 4) +
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)}) +
  ggplot2::theme(axis.text.x = element_text(angle = 20))

## bracni status
data %>%
  ggplot2::ggplot(aes(x = marital_status)) +
  ggplot2::geom_bar(fill = 4) +
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)})

## iznos prihoda
data %>% ggplot2::ggplot(aes(x = income_category)) +
  ggplot2::geom_bar(fill = 4) +
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)}) +
  ggplot2::theme(axis.text.x = element_text(angle = 20))

## tip kreditne kartice
data %>% ggplot2::ggplot(aes(x = card_category)) +
  ggplot2::geom_bar(fill = 4) +
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)})

## limit kreditne kartice
data %>% ggplot2::ggplot(aes(x = credit_limit)) +
  ggplot2::geom_histogram(fill = 4) +
```

```
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)}) +
  ggplot2::scale_x_continuous(n.breaks = 6)

## ukupan iznos revolvinga
data %>% ggplot2::ggplot(aes(x = total_revolving_bal)) +
  ggplot2::geom_histogram(fill = 4) +
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)}) +
  ggplot2::scale_x_continuous(n.breaks = 6)

## ukupan broj transakcija
data %>% ggplot2::ggplot(aes(x = total_trans_ct)) +
  ggplot2::geom_histogram(fill = 4) +
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)}) +
  ggplot2::scale_x_continuous(n.breaks = 6)

## prosjecan iznos transakcije
data %>% ggplot2::ggplot(aes(x = avg_trans_amt)) +
  ggplot2::geom_histogram(fill = 4) +
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)}) +
  ggplot2::scale_x_continuous(n.breaks = 6)

## kvartalna promjena broja transakcija
data %>% ggplot2::ggplot(aes(x = total_ct_chng_q4_q1)) +
  ggplot2::geom_histogram(fill = 4) +
  ggplot2::scale_y_continuous(
    "Frekvencija", labels = function(x) {formattable::comma(x, 0)}) +
  ggplot2::scale_x_continuous(n.breaks = 6)

# Transformacija kategorijskih varijabli

## razina obrazovanja
data <- data %>% dplyr::mutate(
  education_level = dplyr::case_when(
    education_level == "Uneducated" ~ 1,
    education_level == "High␣School" ~ 2,
    education_level == "College" ~ 3,
    education_level == "Graduate" ~ 4,
    education_level == "Post-Graduate" ~ 5,
```

```
    education_level == "Doctorate" ~ 6)) %>%
  dplyr::mutate(
    education_level = dplyr::if_else(
      is.na(education_level), mean(education_level, na.rm = TRUE),
      education_level))

## iznos prihoda
data <- data %>% dplyr::mutate(
  income_category = dplyr::case_when(
    income_category == "Less␣than␣$40K" ~ 1,
    income_category == "$40K␣-␣$60K" ~ 2,
    income_category == "$60K␣-␣$80K" ~ 3,
    income_category == "$80K␣-␣$120K" ~ 4,
    income_category == "$120K␣+" ~ 5)) %>%
  dplyr::mutate(
    income_category = dplyr::if_else(
      is.na(income_category), mean(income_category, na.rm = TRUE),
      income_category))

## tip kreditne kartice
data <- data %>% dplyr::mutate(
  card_category = dplyr::case_when(
    card_category == "Blue" ~ 1,
    card_category == "Silver" ~ 2,
    card_category == "Gold" ~ 3,
    card_category == "Platinum" ~ 4))

## spol
data <- data %>%
    dplyr::mutate(gender_female = dplyr::if_else(gender == "F", 1, 0))

## bracni status
data <- data %>%
  dplyr::mutate(
    marital_status_single = dplyr::case_when(
      marital_status == "Single" ~ 1,
      marital_status != "Unknown" ~ 0),
    marital_status_married = dplyr::case_when(
      marital_status == "Married" ~ 1,
      marital_status != "Unknown" ~ 0)) %>%
  dplyr::mutate_at(
    vars(starts_with("marital_status_")),
```

```
    ~dplyr::if_else(is.na(.), mean(., na.rm = TRUE), .))

# Odabir varijabli
data <- data %>%
  dplyr::select(
    clientnum, customer_age, gender_female, dependent_count, education_level
    marital_status_married, marital_status_single, income_category,
    card_category, credit_limit, total_revolving_bal, total_trans_ct,
    avg_trans_amt, total_ct_chng_q4_q1)

# Standardizacija podataka
standardization <- data %>%
  dplyr::summarise_if(
    ~is.numeric(.),
    list(mean = ~mean(., na.rm = TRUE), sd = ~sd(., na.rm = TRUE)))
data <- data %>%
  dplyr::mutate_if(
    ~is.numeric(.), ~(.-mean(., na.rm = TRUE))/sd(., na.rm = TRUE))

# Scatter plot dvije varijable
data %>% ggplot2::ggplot(aes(x = total_trans_ct, y = total_revolving_bal)) +
  ggplot2::geom_point(col = 4)

# Podaci sa samo dvije varijable
data_2vars <- data %>% dplyr::select(total_trans_ct, total_revolving_bal)

# k-means algoritam za k=1,...,20
set.seed(6091)
segmentations <- dplyr::tibble(k = 1:20) %>%
  dplyr::mutate(
    segmentation = purrr::map(
      k,
      ~stats::kmeans(
        data_2vars, centers = ., iter.max = 100, nstart = 25,
        algorithm = "Hartigan-Wong")),
    tidied = purrr::map(segmentation, ~broom::tidy(.)),
    glanced = purrr::map(segmentation, ~broom::glance(.)),
    augmented = purrr::map(segmentation, ~broom::augment(., data_2vars)),
    silhouette = unlist(
      purrr::map(
        segmentation,
        ~clusterCrit::intCriteria(
```

```
          data_2vars %>% as.matrix(), .$cluster, crit = "Silhouette"))))

# Vizualizacija za k=1,...,6
segmentations %>%
  tidyr::unnest(cols = c(augmented)) %>%
  dplyr::filter(k <= 6) %>%
  ggplot2::ggplot(aes(x = total_trans_ct, y = total_revolving_bal)) +
  ggplot2::geom_point(aes(color = .cluster), alpha = .8) +
  ggplot2::facet_wrap(~k) +
  ggplot2::geom_point(
    data = segmentations %>% tidyr::unnest(cols = c(tidied)) %>%
      dplyr::filter(k <= 6),
    size = 10, shape = "x")

# Elbow metoda
segmentations %>% tidyr::unnest(cols = c(glanced)) %>%
  ggplot2::ggplot(aes(x = k, y = tot.withinss)) +
  ggplot2::geom_line(col = 4) +
  ggplot2::geom_point(col = 4) +
  ggplot2::scale_x_continuous(n.breaks = 20)

# Silhouette koeficijent
segmentations %>% dplyr::filter(k != 1) %>%
  ggplot2::ggplot(aes(x = k, y = silhouette)) +
  ggplot2::geom_line(col = 2) +
  ggplot2::geom_point(col = 2) +
  ggplot2::scale_x_continuous(n.breaks = 20)

# Velicine klastera za k=6
segmentations %>% dplyr::filter(k == 6) %>%
  dplyr::mutate(size = purrr::map(segmentation, ~.$size)) %>%
  dplyr::select(size) %>% unlist()

# Centri klastera za k=6
segmentations %>% dplyr::filter(k == 6) %>%
  tidyr::unnest(tidied) %>%
  dplyr::select(cluster, total_trans_ct, total_revolving_bal) %>%
  tidyr::pivot_longer(c(total_trans_ct, total_revolving_bal)) %>%
  ggplot2::ggplot(aes(x = name, y = value, fill = name)) +
  ggplot2::geom_bar(stat = "identity", position = "dodge") +
  ggplot2::facet_wrap(~cluster) +
  ggplot2::xlab("") +
```

```
  ggplot2::ylab("")

# Podaci sa 6 varijabli
data_6vars <- data %>%
  dplyr::select(
    total_trans_ct, total_revolving_bal, customer_age, marital_status_marrie
    avg_trans_amt, credit_limit)

# Scatter plot 6 varijabli pomocu PCA
data_6vars %>% stats::prcomp() %>%
  ggplot2::autoplot(col = 4)

# k-means algoritam za k=1,...,20
set.seed(2228)
segmentations <- dplyr::tibble(k = 1:20) %>%
  dplyr::mutate(
    segmentation = purrr::map(
      k,
      ~stats::kmeans(
        data_6vars, centers = ., iter.max = 100, nstart = 25,
        algorithm = "Hartigan-Wong")),
    tidied = purrr::map(segmentation, ~broom::tidy(.)),
    glanced = purrr::map(segmentation, ~broom::glance(.)),
    augmented = purrr::map(segmentation, ~broom::augment(., data_6vars)),
    silhouette = unlist(
      purrr::map(
        segmentation, ~clusterCrit::intCriteria(data_6vars %>% as.matrix(),
        .$cluster, crit = "Silhouette"))))

# Vizualizacija za k=1,...,6
segmentations %>% dplyr::filter(k <= 6) %>%
  dplyr::mutate(
    plot = unlist(
      purrr::map(
        segmentation,
        ~{ggplot2::autoplot(segmentation, data = data_6vars)}))) %>%
  dplyr::filter(k == 1) %>%
  dplyr::pull(plot)

# Elbow metoda
segmentations %>% tidyr::unnest(cols = c(glanced)) %>%
  ggplot2::ggplot(aes(x = k, y = tot.withinss)) +
```

```
  ggplot2::geom_line(col = 4) +
  ggplot2::geom_point(col = 4) +
  ggplot2::scale_x_continuous(n.breaks = 20)

# Silhouette koeficijent
segmentations %>% dplyr::filter(k != 1) %>%
  ggplot2::ggplot(aes(x = k, y = silhouette)) +
  ggplot2::geom_line(col = 2) +
  ggplot2::geom_point(col = 2) +
  ggplot2::scale_x_continuous(n.breaks = 20)

# Velicine klastera za k=9
segmentations %>% dplyr::filter(k == 9) %>%
  dplyr::mutate(size = purrr::map(segmentation, ~.$size)) %>%
  dplyr::select(size) %>% unlist()

# Bihevioralni centri klastera za k=9
segmentations %>% dplyr::filter(k == 9) %>%
  tidyr::unnest(tidied) %>%
  dplyr::select(
    cluster, total_trans_ct, total_revolving_bal, avg_trans_amt,
    credit_limit) %>%
  tidyr::pivot_longer(
    c(
      total_trans_ct, total_revolving_bal, avg_trans_amt, credit_limit)) %>%
  ggplot2::ggplot(aes(x = name, y = value, fill = name)) +
  ggplot2::geom_bar(stat = "identity", position = "dodge") +
  ggplot2::facet_wrap(~cluster) +
  ggplot2::xlab("") +
  ggplot2::ylab("") +
  ggplot2::theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank())

# Demografski centri klastera za k=9
segmentations %>% dplyr::filter(k == 9) %>%
  tidyr::unnest(tidied) %>%
  dplyr::select(cluster, customer_age, marital_status_married) %>%
  tidyr::pivot_longer(c(customer_age, marital_status_married)) %>%
  ggplot2::ggplot(aes(x = name, y = value, fill = name)) +
  ggplot2::geom_bar(stat = "identity", position = "dodge") +
  ggplot2::facet_wrap(~cluster) +
```

```
ggplot2::xlab("") +
ggplot2::ylab("") +
ggplot2::theme(
  axis.text.x = element_blank(),
  axis.ticks.x = element_blank())
```

Bibliografija

[1] https://mathworld.wolfram.com/StirlingNumberoftheSecondKind.html.

[2] https://www.kaggle.com/sakshigoyal7/credit-card-customers.

[3] https://www.tidyverse.org/.

[4] https://www.tidymodels.org/.

[5] https://urn.nsk.hr/urn:nbn:hr:217:174751.

[6] A. Dempster, N. Laird, D. Rubin: *Maximum Likelihood From Incomplete Data Via The EM algorithm*. Journal of the Royal Statistical Society. Series B (Methodological), 39:1–38, siječanj 1977.

[7] B, Fritzke: *Some Competitive Learning Methods*. svibanj 1997.

[8] B, Zhang: *Generalized k-harmonic Means-Boosting in Un supervised Learning*. listopad 2000.

[9] C. Ding, X. He, H. Zha M. Gu H. Simon: *A Min-max Cut Algorithm for Graph Partitioning and Data Clustering*. Proceedings - IEEE International Conference on Data Mining, ICDM, 107-114, studeni 2001.

[10] D, Steinley: *Local Optima in K-Means Clustering: What You Don't Know May Hurt You*. Psychological methods, 8:294–304, listopad 2003.

[11] D. Aloise, A. Deshpande, P. Hansen P. Popat: *NP-hardness of Euclidean sum-of-squares clustering*. Machine Learning, 75:245–248, svibanj 2009.

[12] D. Arthur, S. Vassilvitskii: *K-Means++: The Advantages of Careful Seeding*. Svezak 8, stranice 1027–1035, siječanj 2007.

[13] D. Davies, D. Bouldin: *A Cluster Separation Measure*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-1:224 – 227, svibanj 1979.

[14] D. Witten, R. Tibshirani: *A Framework for Feature Selection in Clustering*. Journal of the American Statistical Association, 105:713–726, lipanj 2010.

[15] D.A. Tolliver, G. Miller: *Graph Partitioning by Spectral Rounding: Applications in Image Segmentation and Clustering*. Svezak 1, stranice 1053– 1060, srpanj 2006.

[16] E, Forgy: *Cluster analysis of multivariate data: eIciency versus interpretability of classi7cations*. Biometrics, 21, siječanj 1965.

[17] G. De Soete, J. Carroll: *K-means clustering in a low-dimensional Euclidean space*, stranice 212–219. Springer, 1994.

[18] H. Jia, S. Ding, X. Xu N. ru: *The latest research progress on spectral clustering*. Neural Computing and Applications, 24, lipanj 2014.

[19] H. Zha, X. He, C. Ding H. Simon M. Gu: *Spectral Relaxation for K-means Clustering*. Adv. Neural Inf. Process. Syst., 14, travanj 2002.

[20] H.P. Friedman, J. Rubin: *On Some Invariant Criteria for Grouping Data*. Journal of The American Statistical Association, 62:1159–1178, prosinac 1967.

[21] J, Dunn: *A fuzzy relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters*. Cybernetics and Systems, 3:32–57, studeni 1973.

[22] J, Gower: *A General Coefficient of Similarity and Some of Its Properties*. Biometrics, 27:857–871, prosinac 1971.

[23] J, Koliha: *Block diagonalization*. Mathematica Bohemica, 126, siječanj 2001.

[24] J, MacQueen: *Some Methods for Classification and Analysis of MultiVariate Observations*. Svezak 1, stranice 281–297, siječanj 1967.

[25] J, Ward: *Hierarchical Grouping to Optimize an Objective Function*. Journal of the American Statistical Association, 58:236–244, ožujak 1963.

[26] J. Friedman, T. Hastie, R. Tibshirani: *The Elements of Statistical Learning*. Springer, 2009.

[27] J. Shi, J. Malik: *Normalized Cuts and Image Segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22, svibanj 2002.

[28] J.A. Hartigan, M.A. Wong: *Algorithm AS 136: A K-Means Clustering Algorithm*. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1):100–108, 1979.

[29] L. Hagen, A. Kahng: *New spectral methods for ratio cut partitioning and clustering*. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 11:1074 – 1085, listopad 1992.

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

[31] M, Fiedler: *Algebraic Connectivity of Graphs*. Czechoslovak Mathematical Journal, 23:298–305, siječanj 1973.

[32] M, Fiedler: *Property of eigenvectors of nonnegative symmetric matrices and its application to graph theory*. Czechoslovak Mathematical Journal, 25, studeni 1974.

[33] M. Ester, H.P. Kriegel, J. Sander X. Xu: *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. Svezak 96, stranice 226–231, siječanj 1996.

[34] M. Steinbach, G. Karypis, V. Kumar: *A Comparison of Document Clustering Techniques*. Proceedings of the International KDD Workshop on Text Mining, lipanj 2000.

[35] P. Hansen, B. Jaumard: *Cluster Analysis and Mathematical Programming*. Math. Program., 79:191–215, listopad 1997.

[36] R. Duda, P. Hart, D.G. Stork: *Pattern Classification*. Wiley, 2001.

[37] R. Sokal, F. Rohlf: *Sokal RR, Rohlf FJ. The comparison of dendrograms by objective methods. Taxon 11: 33-40*. Taxon, 11:33–40, veljača 1962.

[38] R. Tibshirani, G. Walther, T. Hastie: *Estimating the Number of Clusters in a Data Set Via the Gap Statistic*. Journal of the Royal Statistical Society Series B, 63:411–423, veljača 2001.

[39] S. Chen, B. ma, K. Zhang: *On the Similarity Metric and the Distance Metric*. Theoretical Computer Science, 410:2365–2376, ožujak 2009.

[40] S. Wierzchon, M. Kłopotek: *Modern Algorithms of Cluster Analysis*. Springer, 2018.

[41] T, Gonzalez: *Clustering to minimize the maximum intercluster distance*. Theoretical Computer Science, 38:293–306, prosinac 1985.

[42] T. Caliński, J.A. Harabasz: *A Dendrite Method for Cluster Analysis*. Communications in Statistics - Theory and Methods, 3:1–27, siječanj 1974.

[43] U, Luxburg: *A tutorial on spectral clustering*. Statistics and Computing, siječanj 2007.

[44] W. Zhao, H. Ma, Q. He: *Parallel K-Means Clustering Based on MapReduce*. Svezak 5931, stranice 674–679, siječanj 1970.

[45] Z, Huang: *Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values*. Data Min. Knowl. Discov., 2:283–304, rujan 1998.

Sažetak

U ovom radu obrađene su neke od metoda segmentacije podataka kao klase nenadziranog učenja, zajedno sa svojim matematičkim temeljima, te je prikazana njihova primjena na podacima o korisnicima kreditnih kartica. Rad započinje formalnim matematički zapisom problema, nakon čega se svaki od obrađenih algoritama detaljno opisuje. Posebna pažnja dana je *k-means* algoritmu i njegovim varijantama. Na kraju rada opisana je primjena navedenog algoritma koristeći programski jezik **R**. U navedenom primjeru prikazana je mogućnost grupiranja klijenata jedne banke na temelju demografskih karakteristika i zajedničkih obrazaca ponašanja, kao i mogućnost opisivanja velikog skupa podataka karakterističnim svojstvima nekoliko grupa dobivenih u procesu.

Summary

This paper discusses some of the data segmentation methods as a class of unsupervised learning, along with their mathematical foundations, and presents their application to credit card user data. The paper begins with a formal mathematical notation of the problem, after which each of the processed algorithms is described in detail. Special attention is given to the *k-means* algorithm and its variants. At the end of the paper, the application of the mentioned algorithm is described using the programming language **R**. The example shows the possibility of grouping bank's clients based on demographic characteristics and common patterns of behavior, as well as the possibility of describing a large data set with the characteristic properties of several groups obtained in the process.

Životopis

Rođen sam 15. veljače 1996. u Splitu. Osnovnoškolsko obrazovanje stiječem u OŠ "Žrnovnica", te srednjoškolsko u III. gimnaziji u Splitu. Akademske godine 2014./2015. upisujem preddiplomski studij matematike na Prirodoslovno-matematičkom fakultetu u Splitu. Isti završavam 2018. godine, nakon čega u akademskoj godini 2018./2019. upisujem diplomski studij matematičke statistike na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta u Zagrebu. Tijekom diplomskog studija razvijam poseban interes u Data Science području. Istim se imam priliku baviti, i dodatno razvijati znanja i vještine, radom u tvrtki Cantab PI.