Kriteriji kompleksnosti za k-means algoritam

Šeperić, Stela

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Stela Šeperić

KRITERIJI KOMPLEKSNOSTI ZA K-MEANS ALGORITAM

Diplomski rad

Voditelj rada:
doc.dr.sc. Pavle Goldstein

Zagreb, Srpanj 2014

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 mentoru doc.dr.sc. Pavlu Goldsteinu na zanimljivoj temi, strpljenju, savjetima i pomoći pri izradi diplomskog rada.
Hvala mojoj obitelji i prijateljima na podršci i razumijevanju tokom svih ovih godina studiranja.

Sadržaj

Uvod

Svrha ovog diplomskog rada je opisati metodu za određivanje k u k-means algoritmu. Određivanje k je poznati problem u statističkoj analizi podataka kao što su strojno učenje, bioinformatika, "data mining" i druga područja. Opisat ćemo metodu koja pretpostavlja da su podaci aproksimativno uniformno distribuirani. To je puno slabija pretpostavka od one o normalnosti podataka koju zahtjevaju neke druge metode. S druge strane, pretpostavka na distribuciju podataka daje puno bolje rezultate nego neke metode koje nemaju nikakve pretpostavke na podatke.

U prvom poglavlju definiramo osnovne pojmove u klaster analizi, koje ćemo kasnije koristiti. Nadalje opisujemo k-means algoritam. Posebna pozornost posvećena je matematičkim aspektima algoritma, tj. težištu skupa i konvergenciji algoritma.

U drugom poglavlju opisujemo problem određivanja broja klastera koji se često javlja pri klasteriranju podataka, te opisujemo dvije metode kojima riješavamo taj problem.

U trećem poglavlju iskazujemo i dokazujemo leme za diskretni i neprekidni slučaj koje nam opisuju ponašanje funkcije cilja u ovisnosti o broju klastera k. U nastavku opisujemo diskretnu metodu najmanjih kvadrata jer ćemo pomoću nje procjenjivati funkciju cilja.

Na kraju, u četvrtom poglavlju testiramo na primjerima našu metodu za određivanje k u k-means algoritmu na različitim skupovima podataka. Skupovi podataka generirani su u programskom jeziku Python, dok je procjena napravljena u programu R.

Poglavlje 1

K-means algoritam

1.1 Definicija

K-means clustering je procedura dijeljenja n točaka u k klastera u kojoj svaka točaka pripada klasteru s najbližim centrom. K-means clustering se često koristi za statističku analizu podataka kao što su strojno učenje, "data mining", bioinformatika i druga područja. Neka su podaci reprezentirani skupom vektora $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^n$. Za proizvoljan $k \in \mathbb{N}$ sa $\{C_1, \ldots, C_k\}$ označimo k klastera, te sa $c_1, \ldots, c_k$ pripadne centare. Cilj ove procedure je naći optimalnu k-particiju skupa X. To se postiže minimizacijom funkcije cilja f, koja je definirana u terminima klastera $C_1, \ldots, C_k$ i centara klastera $c_1, \ldots, c_k$.

Definicija 1.1.1. *Neka je c_i centar klastera C_i. Definiramo funkciju cilja sa*

$$f(C_1, \ldots, C_k, c_1, \ldots, c_k) = \sum_{i=1}^{k} \sum_{x \in C_i} d^2(x, c_i), \tag{1.1}$$

gdje je $d(\cdot, \cdot)$ Euklidska udaljenost

$$d(x, y) = \sqrt{\left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)}, \tag{1.2}$$

za $x = \{x_1, \ldots, x_n\}, y = \{y_1, \ldots, y_n\} \in \mathbb{R}^n$

Postoji nekoliko algoritama koji nastoje riješiti ovaj problem. No, svakako najpoznatiji k-means algoritam je Lloydov algoritam koji spada u grupu algoritama particioniranja.

1.2 Lloydov algoritam

Lloydov algoritam počinje tako da za zadani skup točaka X i fiksni broj klastera k, inicijalno, najčešće na slučajan način, odabire k centara $c_1, \ldots, c_k$. Tada nastavlja alternirajući između sljedeća dva koraka:

1. Pridruživanje točaka klasteru s najbližim centrom
$$C_i^{t+1} = \{x : d(x, c_i^{(t)}) \leq d(x, c_j^{(t)}), \forall j\} \tag{1.3}$$
2. Određivanje centara za novi raspored klastera
$$c_i^{(t+1)} = \frac{1}{|C_i^{(t)}|} \sum_{x \in C_i^{(t)}} x \tag{1.4}$$

Ovdje nam $C_i^{(t)}$ i $c_i^{(t)}$ označavaju i-ti klaster i i-ti centar u t-toj iteraciji, dok $|C_i^{(t)}|$ označava veličinu skupa $C_i^{(t)}$. Algoritam se zaustavlja kada se particije stabiliziraju ili nakon unaprijed određenog broja koraka. Lako se vidi da koraci (1.3) i (1.4) smanjuju vrijednost funkcije cilja f. Štoviše odabiri u (1.3) i (1.4) su optimalni odabiri u smislu da oni (lokalno) maksimalno poboljšavaju danu konfiguraciju. Dakle, ako označimo s $f^{(i)}$ vrijednost funkcije cilja u i-toj iteraciji, dobivamo padajući niz

$$f^{(1)} \geq f^{(2)} \geq \cdots \geq 0 \tag{1.5}$$

Iz toga slijedi da će algoritam dostići minimum od f u konačnom broju koraka, ali iz (1.5) također slijedi da će taj minimum vrlo često biti lokalan. Ove tvrdnje ćemo dokazati kasnije u potpoglavlju 1.4. Do ovog problema dolazi zbog osjetljivosti k-means algoritma na početne uvjete, kao što je početni odabir klastera ili centara. U praksi, ovaj problem se rješava tako da se algoritam pokrene više puta ali s različitim početnim uvjetima, te se tada odabire najbolje rješenje. Druga mogućnost je da se početni uvjeti pažljivije konstruiraju. Problem u Lloydovom algoritmu je pohlepan odabir u koracima (1.3) i (1.4).

1.3 Težište skupa

Definicija 1.3.1. *Neka je $X = \{x^1, \ldots, x^k\} \subseteq \mathbb{R}^n, k \in \mathbb{N}$, konačan, neprazan skup, pri čemu je*
$$x^1 = (x_1^1, , x_2^1, \ldots, x_n^1), \ldots, x^k = (x_1^k, , x_2^k, \ldots, x_n^k) \in \mathbb{R}^n.$$
Težište skupa X je točka $t_X = (t_1, t_2, \ldots, t_n) \in \mathbb{R}^n$ u kojoj funkcija
$$f(t) := \sum_{i=1}^{k} d^2(t, x^i)$$

postiže minimum, pri čemu je $t \in \mathbb{R}^n, x^1, \ldots, x^k \in X$ *i d metrika.*

Primjetimo da je za d iz (1.2), funkcija

$$f(t) = f(t_1, t_2, \ldots, t_n) := \sum_{i=1}^{k} \sum_{j=1}^{n} (x_j^i - t_j)^2.$$

.

Lema 1.3.2. *Neka je* $d : X \times X \to \mathbb{R}$ *euklidska udaljenost definirana formulom (1.2). Tada je*

$$t_X = \left(\frac{\sum_{i=1}^{k} x_1^i}{k}, \frac{\sum_{i=1}^{k} x_2^i}{k}, \ldots, \frac{\sum_{i=1}^{k} x_n^i}{k} \right) \in \mathbb{R}^n$$

točka minimuma funkcije f.

Dokaz. Prvo računamo gradijent funkcije f:

$$\nabla f = \left(\frac{\partial f}{\partial t_1}, \frac{\partial f}{\partial t_2}, \ldots, \frac{\partial f}{\partial t_n} \right).$$

Neka je $l \in \{1, 2, \ldots, n\}$ proizvoljan. Lako se vidi da vrijedi:

$$\frac{\partial f}{\partial t_l}(t) = \sum_{i=1}^{k} (-2x_l^i + 2t_l) = -2 \sum_{i=1}^{k} x_l^i + 2kt_l.$$

Zbog proizvoljnosti od l, gornja jednakost vrijedi $\forall l \in \{1, 2, \ldots, n\}$. Izjednačavanjem gradijenta s nulom lako se dobiju koordinatne kritične točke t_X funkcije f:

$$t_l = \frac{\sum_{i=1}^{k} x_l^i}{k}.$$

Možemo na dva načina pokazati da je dobivena točka upravo točka minimuma funkcije f. Prvi način je da odredimo Hesseovu matricu H. Lako se vidi da su sve mješovite parcijalne derivacije jednake nuli, dok za ostale vrijedi

$$\frac{\partial^2 f}{\partial t_l \partial t_l}(t) = 2k, \forall l \in \{1, 2, \ldots, n\},$$

što je strogo veće od nule jer je k broj točaka u skupu X. Dakle, matrica H izgleda ovako:

$$H = \begin{bmatrix} 2k & 0 & \cdots & 0 \\ 0 & 2k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2k \end{bmatrix},$$

i ona je pozitivno definitna matrica. Dakle, dobivena kritična točka je točka minimuma funkcije f. Drugi način je da uočimo da gradijent funkcije f ima jedinstvenu nultočku, te da vrijedi

$$\lim_{\|t\|\to\infty} f(t) = +\infty,$$

iz čega možemo zaključiti da dotična nultočka mora biti točka minimuma. Dakle, funkcija f postiže minimum u točki

$$t_X = \left(\frac{\sum_{i=1}^{k} x_1^i}{k}, \frac{\sum_{i=1}^{k} x_2^i}{k}, \ldots, \frac{\sum_{i=1}^{k} x_n^i}{k} \right).$$

□

Napomena 1.3.3. *Uočimo da su koordinate težišta t_X aritmetičke sredine odgovarajućih koordinata točaka iz skupa X.*

1.4 Kovergencija algoritma

Želimo dokazati da k-means algoritam, odnosno funkcija cilja konvergira. Iskazujemo sljedeće dvije leme i teorem koji to potvrđuju.

Lema 1.4.1. *Neka su $C_1, \ldots, C_k$ klasteri dobiveni u nekoj iteraciji k-means algoritma, te neka su $c_1', \ldots, c_k'$ novi centri i $C_1', \ldots, C_k'$ novi klasteri. Tada vrijedi:*

(a) $c_i' = arg\min_t \sum_{x \in C_i} d^2(x,t), \forall i \in \{1, \ldots, k\}$

(b) $\forall x \in X, x \in C_i' \Leftrightarrow d^2(x, c_i') \leq d^2(x, c_j'), \forall j = 1, \ldots, k$

Dokaz. Dokaz tvrdnje (a) analogan je dokazu Leme 1.3.2. Tvrdnja (b) slijedi iz činjenice da smo klaster C_i' definirali kao $C_i' := \{x \in X : i = arg\min_{1 \leq j \leq k} d^2(x, c_j')\}$, tj. klaster C_i' sastoji se od onih točaka za koje je c_i' najbliži cenatar. □

Lema 1.4.2. *Neka su $c_1, \ldots, c_k$ centri dobiveni u nekoj iteraciji k-means algoritma i f vrijednost funkcije cilja u toj iteraciji, te neka su $c_1', \ldots, c_k'$ centri dobiveni u sljedećoj iteraciji i f' funkcija cilja u toj iteraciji. Tada vrijedi:*

$$f' = \sum_{i=1}^{k} \sum_{x \in C_i'} d^2(x, c_i') \leq f = \sum_{i=1}^{k} \sum_{x \in C_i} d^2(x, c_i)$$

Dokaz. Koristimo pomoćne tvrdnje dokazane u Lemi 1.4.1. Prema tvrdnji (b) vrijedi

$$\forall x \in X, x \in C_i' \Leftrightarrow d^2(x, c_i') \leq d^2(x, c_j'), \forall j = 1, \ldots, k$$

tj. za svaku točku, od svih centara kandidata, odabiremo najbliži centar. Kako ovo vrijedi za svaku točku c_j', onda specijalno vrijedi i za $c_j' = c_i$, a kako tvrdnja vrijedi za svaku točku x iz klastera C_i, imamo

$$\sum_{x \in C_i} d^2(x, c_i') \leq \sum_{x \in C_i} d^2(x, c_i), \forall j = 1, \ldots, k$$

Kako ovo vrijedi za svaki klaster $i = 1, \ldots, k$, dobivamo

$$f' = \sum_{i=1}^{k} \sum_{x \in C_i'} d^2(x, c_i') \leq f = \sum_{i=1}^{k} \sum_{x \in C_i} d^2(x, c_i),$$

što pokazuje da funkcija cilja pada. Prema tvrdnji (a) iz Leme 1.4.1 vrijedi

$$c_i' = arg \min_t \sum_{x \in C_i} d^2(x, t).$$

Dakle, za svako t vrijedi

$$\sum_{x \in C_i} d^2(x, c_i') \leq \sum_{x \in C_i} d^2(x, t),$$

pa specijalno to vrijedi i za $t = c_i$. Budući da to vrijedi $\forall i = 1, \ldots, k$ dobivamo

$$f' = \sum_{i=1}^{k} \sum_{x \in C_i'} d^2(x, c_i') \leq f = \sum_{i=1}^{k} \sum_{x \in C_i} d^2(x, c_i).$$

Prema tome slijedi $f' \leq f$. □

Teorem 1.4.3. *Funkcija cilja u k-means algoritmu konvergira u (lokalni) minimum.*

Dokaz. Kao što smo u uvodu opisali, k-means algoritam odvija se ovako:

1. odredimo k centara na slučajan način
2. odredimo k klastera
3. odredimo nove centre kao težišta tih klastera
4. odredimo nove klastere

Nakon toga ponavljaju se naizmjenično koraci iii. i iv. sve dok cijela iteracija ne prođe bez promjene klastera. Primijetimo da zbog Leme 1.4.1 (a) imamo pohlepan odabir u iii. i iv. Iz Leme 1.4.2 dobivamo da je u svakom koraku vrijednost funkcije cilja manja ili jednaka nego u prethodnom. Odnosno, dobivamo niz vrijednosti funkcije cilja takav da vrijedi (1.5). Dakle, vrijednost funkcije cilja se u svakom koraku algoritma smanjuje i funkcija je ograničena odozdo. Prema tome slijedi da funkcija cilja u k-means algoritmu kovergira, ali u (lokalni) minimum. □

Primjer 1.4.4. *Pokažimo na jednostavnom primjeru rad k-means algoritma. Neka je:*

$$X = \{x_1, x_2, x_3, x_4, x_5\} \subset \mathbb{R}^2$$

$$x_1 = (1, 2), x_2 = (3, 4), x_3 = (1, 1), x_4 = (2, 3), x_5 = (4, 2)$$

Neka je $k = 2$, željeni broj klastera, te neka su $c_1 = x_1$ i $c_2 = x_4$ inicijalni centri.

1. $c_1 = x_1, c_2 = x_4$

$$\begin{array}{lll} d^2(x_2, c_1) = 8 & d^2(x_2, c_2) = 2 & x_2 \to C_2 \\ d^2(x_3, c_1) = 1 & d^2(x_3, c_2) = 5 & x_3 \to C_1 \\ d^2(x_5, c_1) = 9 & d^2(x_5, c_2) = 5 & x_5 \to C_2 \end{array}$$

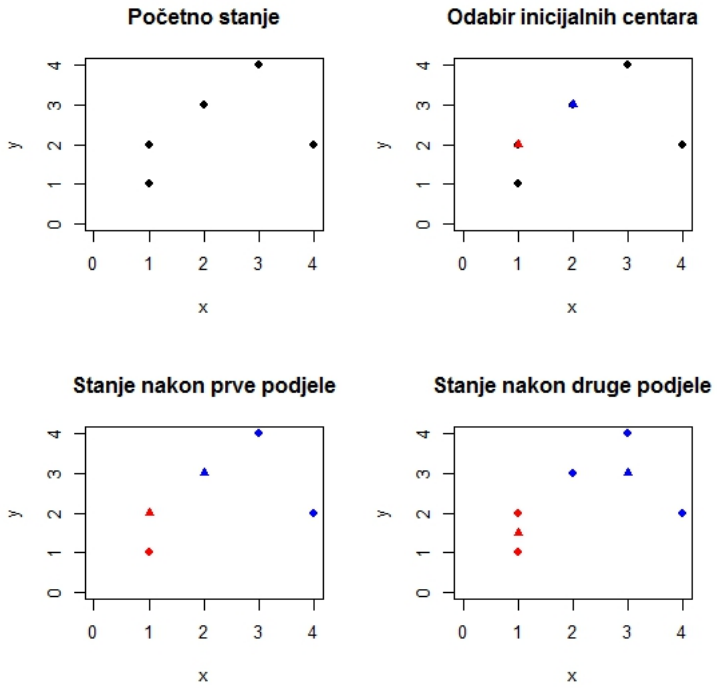
   $\Rightarrow C_1 = \{x_1, x_3\}, C_2 = \{x_2, x_4, x_5\}$

2. $c_1 = t_{C_1} = (1, \frac{3}{2}), c_2 = t_{C_2} = (3, 3)$

$$\begin{array}{lll} d^2(x_1, c_1) = \frac{1}{4} & d^2(x_1, c_2) = 5 & x_1 \to C_1 \\ d^2(x_2, c_1) = \frac{41}{4} & d^2(x_2, c_2) = 1 & x_2 \to C_2 \\ d^2(x_3, c_1) = \frac{1}{4} & d^2(x_3, c_2) = 8 & x_3 \to C_1 \\ d^2(x_4, c_1) = \frac{13}{4} & d^2(x_4, c_2) = 1 & x_4 \to C_2 \\ d^2(x_5, c_1) = \frac{37}{4} & d^2(x_5, c_2) = 2 & x_5 \to C_2 \end{array}$$

   $\Rightarrow C_1 = \{x_1, x_3\}, C_2 = \{x_2, x_4, x_5\}$

3. *c_1 i c_2 se više ne mijenjaju. Algoritam može stati.*



Slika 1.1: Primjer 1.6.

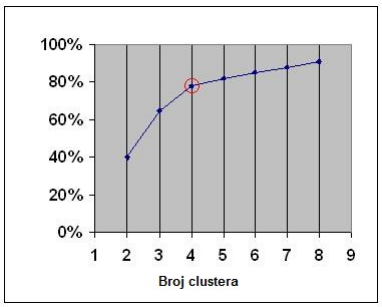
Poglavlje 2

Određivanje broja klastera

Određivanje broja klastera za neki skup podataka je česti problem u klasteriranju podataka. Nije jasno koji k je pravi, te je to često *ad hoc* odluka koja se temelji na prijašnjim saznanjima, pretpostavkama i praktičnom iskustvu. Odabir ovisi o distribuciji točaka (podataka). Nadalje, povećanje k uvijek će smanjiti vrijednost funkcije cilja. Ekstremni slučaj je kada imamo da je broj točaka jednak broju klastera ($m = k$), tj. svaka točka je svoj klaster. Tada je vrijednost funkcije cilja nula. Intuitivno, odabirom broja k želimo postići ravnotežu između točnosti pridruživanja točaka klasteru i minimizacije funkcije cilja kao funkcije od k. Pogledajmo dvije metode za odabir k.

2.1 Metoda lakta

Metoda lakta promatra postotak objašnjene varijance kao funkciju broja klastera. Odabiremo broj klastera k kada dodavanje još jednog klastera ne poboljšava podjelu podataka. Preciznije, ako napravimo graf u kojem prikazujemo ovisnost broja klastera o postotku objašnjene varijance, prvi klaster će puno doprinjeti jer će objasniti veliki postotak varijance. Ali u nekom trenutku ta dobit će se početi smanjivati, a grafički to znači da ćemo dobiti kut na grafu. U trenutku kada dobijemo taj kut, očitamo s grafa broj klastera. Ime metode dolazi od izgleda grafa, jer upravo taj kut podsjeća na lakat. Dodajmo još da se postotak objašnjene varijance dobiva kao omjer varijance između grupa (klastera) i ukupne varijance. Sljedeća slika je primjer jednog grafa za metodu lakta.



Slika 2.1: Lakat je označen crvenim krugom. Broj klastera bi trebao biti 4.

2.2 G-means

G-means algoritam temelji se na statističkom testu, kojim testiramo hipotezu da je podskup podataka normalno distribuiran. G-means pokreće k-means algoritam, pri čemu povećava k, dok ne dođemo u poziciju da ne odbacimo hipotezu da su podaci pridruženi svakom k-means centru normalno distribuirani. Dvije ključne prednosti ovog algoritma su da test hipoteza ne ograničava kovarijancu podataka i da se ne treba računati cijela matrica kovarijanci. Nadalje, jedini zahtjev je da odredimo razinu značajnosti α.
G-means algoritam kreće s malim brojem k-means centara, možemo čak postaviti $k = 1$. U svakoj iteraciji algoritam podijeli na dva dijela one centre za čije podatke utvrdimo da ne dolaze iz normalne distribucije. Između podjela, pokrećemo k-means algoritam na cijelom skupu podataka i na svim centrima da bi poboljšali trenutnu situciju. Da bi u potpunosti specificirali G-means algoritam, trebamo definirati hipoteze i statistički test koji koristimo. Hipoteze su:

- H0: Podaci oko centra dolaze iz normalne distribucije
- H1: Podaci oko centra ne dolaze iz normalne distribucije

Ako ne odbacimo hipotezu H0, tada smatramo da je jedan centar dovoljan za dane podatke. No, ako odbacimo hipotezu H0, tada želimo podijeliti klaster. Test koji koristimo temelji se na Anderson- Darling statistici. Neka su x_i vrijednosti koje su prilagođene na $N(0, 1)$, te neka je $x_{(i)}$ i-ta vrijednost. Neka je $z_i = F(x_{(i)})$, gdje je F kumulativna funkcija distribucije N(0,1). Tada imamo statistiku:

$$A^2(Z) = -\frac{1}{n}\sum_{i=1}^{n}(2i-1)[log(z_i) + log(1 - z_{n+1-i})] - n$$

Nadalje, kada imamo μ i σ procjenjene iz podataka, onda moramo prilagoditi statistiku na:

$$A_*^2(Z) = A^2(Z)\left(1 + \frac{4}{n} - \frac{25}{n^2}\right)$$

Za dani podskup podataka od skupa X u d dimenzija koji pripada centru c, test hipoteza teče ovako:

1. Odaberi razinu značajnosti α
2. Inicijaliziraj dva centra. Njih još zovemo "djeca" od c
3. Pokreni k-means na ta dva centra u X. Ovo se može pokrenuti do kraja ili do neke točke zaustavljanja. Neka su c_1, c_2 djeca centri izabrani k-means algoritmom.
4. Neka je $v = c_1 - c_2$ d-dimenzionalan vektor koji povezuje ta dva centra. Projiciraj X na v: $x'_i = \frac{\langle x_i, v\rangle}{\|v\|^2}$. X' je 1-dimenzinalna reprezentacija podataka projiciranih na v. Transformiraj X' tako da ima očekivanje 0 i varijancu 1.
5. Neka je $z_i = F(x_{(i)})$. Ako $A_*^2(Z)$ nije u skupu kritičnih vrijednosti, uz razinu značajnosti α, tada ne odbacujemo H0, zadržavamo originalne centre i odbacujemo $\{c_1, c_2\}$. Inače, odbacujemo H0 i prihvaćamo $\{c_1, c_2\}$ umjesto originalnih centara.

Napomenimo još da k-means algoritam implicitno pretpostavlja da su podaci u svakom klasteru sferično raspoređeni oko centra, a to zapravo nije slučaj u praksi sa stvarnim podacima. U takvim situacijama, G-means algoritam dolazi do izražaja jer on pretpostavlja da podaci u svakom clusteru prate višedimenzionalnu normalnu distribuciju. Iako G-means ima veću pogrešku druge vrste (Type II error) kada je broj podataka mali, to ga sprečava da napravi overfit u smislu da izabere previše centara kada podaci nisu sferični.

Poglavlje 3

Lema

3.1 Lema

U prethodnom poglavlju opisali smo dvije metode za određivanje k u k-means algoritmu. Dok je metoda lakta bila prilično jednostavna i nije zahtijevala gotovo nikakve pretpostavke, G-means metoda je bila konceptualno složenija. Ona je zapravo zahtijevala normalnu distribuiranost podataka u klasterima. To je jedna vrlo jaka pretpostavka, iako se za taj algoritam zahtijeva samo da se unaprijed odredi razina značajnosti α. Stoga ćemo u ovom poglavlju opisati metodu koja nema tako jake pretpostavke kao G-means, ali nije ni trivijalna kao metoda lakta. Ta metoda je zapravo temelj ovog diplomskog rada. Pretpostavimo da imamo optimiziranu funkciju cilja $f(k)$, te da su podaci ekvidistantni. Trebamo pokazati da se funkcija cilja u k-means algoritmu ponaša otprilike kao $\frac{C_1}{k^2} + \frac{C_2}{k}$, pri čemu je k broj klastera, a C_1 i C_2 su konstante.

Lema 3.1.1. *Pretpostavimo da imamo k klastera, da su podaci ekvidistantni, te da u svakom klasteru imamo $\frac{m}{k} + 1$ točaka (podataka). Za funkciju cilja vrijedi:*

$$f(k) \approx \frac{m}{12}\left(\frac{C_1}{k^2} + \frac{C_2}{k}\right)$$

pri čemu su C_1 i C_2 konstante.

Dokaz. Uzmimo da u svakom klasteru imamo $\frac{m}{k}+1$ točaka iz $\mathbb{R}$, te da je udaljenost između njih $\frac{1}{n}$. Pretpostavimo da je do na translaciju $C_i = \{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{m}{kn}\}$ neki klaster. Radi lakšeg računanja, označimo sa $z = \frac{m}{kn}$. Sada imamo:

$$\begin{aligned} \sum_{x \in C_i} x &= \sum_{i=0}^{z} \frac{i}{n} = \frac{1}{n}\sum_{i=0}^{z} i = \frac{1}{n}\frac{z(z+1)}{2} \\ \bar{x} &= \frac{1}{z+1}\frac{1}{n}\frac{z(z+1)}{2} = \frac{z}{2n} \end{aligned}$$

Sada računamo funkciju cilja.

$$\begin{aligned}
\sum_{x\in C_i}(x-\bar{x})^2 &= \sum_{i=0}^{z}\left(\frac{i}{n}-\frac{z}{2n}\right)^2 \\
&= \sum_{i=0}^{z}\left(\frac{2i-z}{2n}\right)^2 \\
&= \frac{1}{4n^2}\sum_{i=0}^{z}(2i-z)^2 \\
&= \frac{1}{4n^2}\left[4\sum_{i=0}^{z}i^2-4z\sum_{i=0}^{z}i+z^2\sum_{i=0}^{z}1\right] \\
&= \frac{1}{4n^2}\left[4\frac{z(z+1)(2z+1)}{6}-4z\frac{z(z+1)}{2}+z^2(z+1)\right] \\
&= \frac{z(z+1)(2z+1)}{6n^2}-\frac{z^2(z+1)}{2n^2}+\frac{z^2(z+1)}{4n^2} \\
&= \frac{2z(z+1)(2z+1)-6z^2(z+1)+3z^2(z+1)}{12n^2} \\
&= \frac{z(z+1)[2(2z+1)-6z+3z]}{12n^2} \\
&= \frac{z(z+1)(z+2)}{12n^2} \\
&= \frac{z^3+3z^2+2z}{12n^2} \\
&= \frac{\left(\frac{m}{kn}\right)^3+3\left(\frac{m}{kn}\right)^2+2\left(\frac{m}{kn}\right)}{12n^2} \\
&= \frac{m}{kn}\frac{1}{12n^2}\frac{m^2+3mkn+2k^2n^2}{k^2n^2} \\
&= \frac{1}{k}\frac{m}{12}\left(\frac{m^2}{n^5k^2}+\frac{3mkn}{n^5k^2}+\frac{2k^2n^2}{n^5k^2}\right) \\
&= \frac{1}{k}\frac{m}{12}\left(\frac{m^2}{n^5k^2}+\frac{3m}{n^4k}+\frac{2}{n^3}\right) \\
&= \frac{1}{k}\frac{m}{12}\left(\frac{C_1}{k^2}+\frac{C_2}{k}\right)
\end{aligned}$$

Dakle, dobili smo da je vrijednost funkcije cilja unutar jednog klastera $\frac{1}{k}\frac{m}{12}\left(\frac{C_1}{k^2}+\frac{C_2}{k}\right)$.

Budući da smo pretpostavili da imamo k takvih klastera slijedi :

$$k \cdot \frac{1}{k}\frac{m}{12}\left(\frac{C_1}{k^2} + \frac{C_2}{k}\right) = \frac{m}{12}\left(\frac{C_1}{k^2} + \frac{C_2}{k}\right)$$

što je upravo trebalo dokazati. Dakle, funkcija cilja se ponaša kao $\frac{m}{12}\left(\frac{C_1}{k^2} + \frac{C_2}{k}\right)$ □

U neprekidnom slučaju lemu iskazujemo i dokazujemo na sljedeći način.

Lema 3.1.2. *Neka je $X \subseteq (0,1)$. Pretpostavimo da imamo k klastera, da su podaci u klasterima uniformno distribuirani, $C_i \sim U(\frac{i-1}{k}, \frac{i}{k})$, te da u svakom klasteru imamo $\frac{m}{k}$ točaka (podataka) i $k \ll m$. Za funkciju cilja vrijedi:*

$$f(k) \approx \frac{m}{12}\frac{1}{k^2}$$

Dokaz. Budući da je $C_i \sim U(\frac{i-1}{k}, \frac{i}{k})$, imamo da je $Var(C_i) = \frac{\left(\frac{1}{k}\right)^2}{12}$. Kada računamo funkciju cilja unutar jednog klastera, dobivamo

$$f\mid_{C_i} = \sum_{x \in C_i}(x - \bar{x})^2$$

Uočimo da je uzoračka varijanca dana sa

$$s^2(C_i) = \frac{1}{\left(\frac{m}{k} - 1\right)}\sum_{x \in C_i}(x - \bar{x})^2$$

Budući da je $\frac{1}{\frac{m}{k}-1} \approx \frac{1}{\frac{m}{k}}$, dobivamo $s^2(C_i) = \frac{1}{\left(\frac{m}{k}\right)}\sum_{x \in C_i}(x - \bar{x})^2$. Ako se uzoračka i teorijska varijanca poklapaju,$s^2(C_i) = Var(C_i)$, imamo

$$\left(\frac{k}{m}\right)\sum_{x \in C_i}(x - \bar{x})^2 = \frac{1}{12k^2}$$

$$\sum_{x \in C_i}(x - \bar{x})^2 = \frac{m}{k}\frac{1}{12k^2} = \frac{m}{12k^3}$$

Budući da smo pretpostavili da imamo k takvih klastera slijedi:

$$f(X,k) \approx \sum_{i=1}^{k}\sum_{x \in C_i}(x - \bar{x})^2 = k \cdot \frac{m}{12k^3} = \frac{m}{12k^2}$$

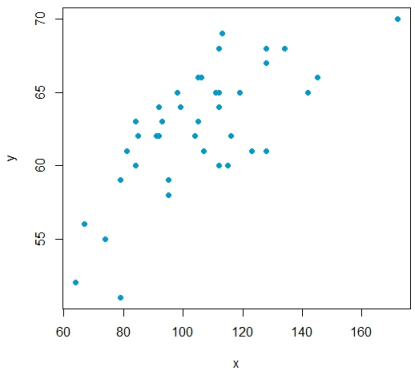
što je upravo trebalo dokazati. Dakle, funkcija cilja se ponaša kao $\frac{m}{12k^2}$. □

Napomena 3.1.3. *Napomenimo da trebamo uključiti i dimenziju prostora u kojem se nalaze podaci. Stoga imamo da se funkcija cilja ponaša kao $\frac{d}{12}\frac{m}{k^2}$. Budući da su svi primjeri napravljeni u dvodimenzionalnom slučaju ($d = 2$), imamo $\frac{m}{6}\frac{1}{k^2}$.*

Napomena 3.1.4. *Uočimo da smo kombinatorno i statistički dobili slične rezultate.*

3.2 Diskretna metoda najmanjih kvadrata

Sada ćemo ukratko opisati diskretnu metodu najmanjih kvadrata, jer pomoću nje procjenjujemo $f(k)$ za razne k. Često u praksi imamo puno podataka kao na slici dolje.



Slika 3.1: Podaci

Podatke želimo prilagoditi nekom (jednostavnom) linearnom modelu. Odnosno, želimo naći linearni model koji najbolje opisuje naše podatke. U ovom primjeru pravac bi mogao dobro opisivati dane podatke. Cilj je provući pravac tako da suma kvadrata razlika između stvarnih vrijednosti i vrijednosti predviđenih pravcem bude najmanja. Budući da imamo model $y_k = \alpha + \beta x_k$, tražimo α i β takve da prilagođene vrijednosti od y_k dane s $\widehat{y_k} = \alpha + \beta x_k$ budu što bliže opaženim vrijednostima y_k. Od tuda i dolazi ime metoda najmanjih kvadrata. Formalno zapisano, neka je y zadan na diskretnom skupu točaka $x_0, \ldots, x_n$, te neka je $\hat{y} = \alpha + \beta x$ aproksimacijska funkcija. Točaka $x_0, \ldots, x_n$ ima mnogo više nego nepoznatih parametara α, β aproksimacijske funkcije $\hat{y}$, tj. $n \gg 2$. Aproksimacijska funkcija određuje se iz uvjeta da je 2-norma vektora pogrešaka u čvorovima aproksimacije najmanja moguća, tj. minimizira se izraz:

$$\begin{aligned} S &= \sum_{k=0}^{n} (y_k - \hat{y}_k)^2 \\ &= \sum_{k=0}^{n} (y_k - \alpha - \beta x_k)^2 \longrightarrow min \end{aligned}$$

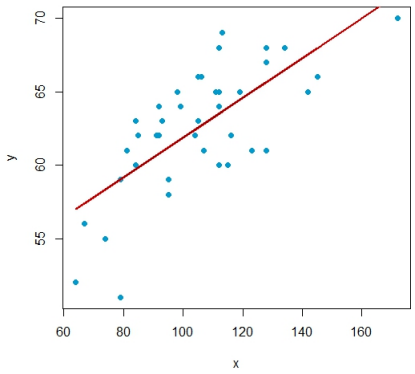
Uočimo da:

- uvijek je $S \geq 0$, bez obzira kakvi su parametri, jer se radi o zbroju kvadrata

- Funkcija S minimizira se kao funkcija dvije varijable α i β
- S je dovoljno glatka funkcija, jer je funkcija u parametrima α i β, pa je nužni uvjet ekstrema

$$\frac{\partial S}{\partial \alpha} = 0, \frac{\partial S}{\partial \beta} = 0$$

Sada u gornjem primjeru, aproksimacijski pravac za dane podatke izgleda ovako:



Slika 3.2: Aproksimacijski pravac

Odnosno, ovaj pravac predstavlja linearni model koji nabolje opisuje zadane podatke.

3.3 Primjena MNK na problem

U Lemi 3.1.1 teoretski smo pokazali da se funkcija cilja ponaša kao $\frac{C_1}{k^2} + \frac{C_2}{k}$. Sada to trebamo testirati u praksi. Dakle, aproksimiramo dane podatke pravcem. Uzmimo da naš model izgleda ovako:

$$f(k) = \frac{\alpha}{k} + \frac{\beta}{k^2}, \tag{3.1}$$

te ćemo procijeniti njegove α i β . Procjene računamo u programu R. Za početak trebamo transformirati (3.1). To radimo tako da pomnožimo (3.1) s k^2. Time dobivamo

$$f(k) \cdot k^2 = \alpha \cdot k + \beta$$

Stavimo li da je $z(k) = f(k) \cdot k^2$, imamo $z(k) = \alpha \cdot k + \beta$. Pomoću funkcije `lm` u R-u dobivamo $\hat{\alpha}$ i $\hat{\beta}$, koje uvrštavamo u (3.1) i time dobivamo $\widehat{f(k)} = \frac{\hat{\alpha}}{k} + \frac{\hat{\beta}}{k^2}$. Nakon toga

računamo $f_{min}(k)$ tako da stavimo da je $f_{min}(k) = f(k) - \widehat{f(k)}$. Za najmanji $f_{min}(k)$ očitamo k, i taj k je optimalan broj klastera za podjelu testiranih podataka.
Budući da smo u Lemi 3.1.2 dobili da se funkcija cilja ponaša kao $\frac{m}{12k^2}$, trebali bi testirati takav model. No, mi testiramo ovakav model

$$f(k) = \alpha + \frac{\beta}{k}, \tag{3.2}$$

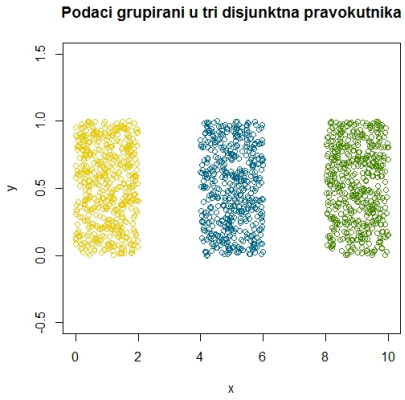
jer se MNK bolje ponaša na ovakvom modelu i jer su greške manje. Transformirajući taj model tako da (3.2) pomnožimo s k dobivamo $f(k) \cdot k = \alpha k + \beta$. Stavimo li da je $z(k) = f(k) \cdot k$, imamo $z(k) = \alpha k + \beta$. Sada na isti način kako je gore opisano, procjenjujemo α i β te računamo $f_{min}(k)$.

Poglavlje 4

Primjeri

U ovom poglavlju testiramo tvrdnje Leme 3.1.1 i Leme 3.1.2 na različitim skupovima podataka. Podaci su uniformno distribuirani.

Primjer 4.0.1. *Za početak uzmimo ovakve podatke.*



Slika 4.1: Podaci grupirani u tri disjunktna pravokutnika

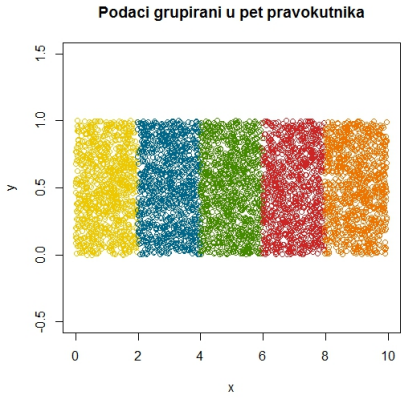
Budući da su podaci grupirani u tri disjunktna pravokutnika, očekujemo da ćemo dobiti da je optimalan broj klastera $k = 3$*. Testiramo ovaj skup podataka za* $k = 2, \ldots, 20$*, te za zvaki* k *računamo pripadnu funkciju cilja* $f(k)$*, i nakon toga* $f_{min}(k)$ *za model (3.1) i* $f2_{min}(k)$ *za model (3.2). Dobivamo sljedeće rezultate:*

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
2	4574.905	3286.607	2681.654
3	608.705	-516.847	-674.891
4	493.835	-450.337	-484.933
5	467.364	-335.977	-328.507
6	453.046	-243.074	-220.894
7	442.937	-170.065	-143.909
8	434.611	-112.481	-86.915
9	430.089	-63.622	-40.632
10	426.822	-22.853	-3.256
11	424.996	12.234	28.172

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
12	423.366	41.971	54.254
13	421.702	67.278	76.038
14	420.706	89.714	95.141
15	419.133	108.683	110.986
16	418.333	126.036	125.428
17	292.119	15.979	12.663
18	267.815	6.143	0.312
19	150.457	-98.181	-106.349
20	173.013	-63.825	-74.168

Procjenjeni parametri za model (3.1) su $\hat{\alpha} = 4976.777$ *i* $\hat{\beta} = -4800.362$*, a za model (3.2) su* $\hat{\alpha} = 64.28415$ *i* $\hat{\beta} = 3657.935$*. Iz gornje tablice se vidi da se najmanji* $f(k)$ *za oba modela postiže upravo za* $k = 3$*. Dakle,* $k = 3$ *je optimalan broj klastera.*

Primjer 4.0.2. *Pogledajmo sada ovakav skup podataka.*



Slika 4.2: Podaci grupirani pet pravokutnika

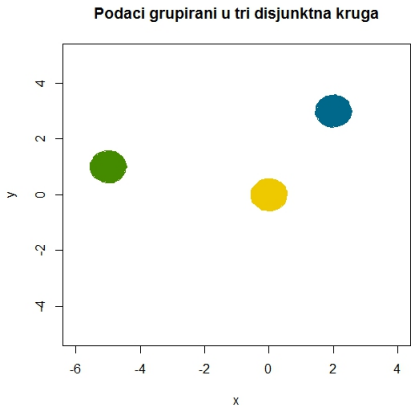
Podaci su grupirani u pet pravokutnika s vrlo malim preklapanjem između pravokutnika. Sada više nije jasno koji bi k bio optimalan. Testirajući ovaj skup podataka ponovo za $k = 2, \ldots, 20$*, dobivamo sljedeće pripadne funkcije cilja:*

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
2	10804.731	4515.575	4366.976
3	4988.672	1250.7011	800.474
4	2984.082	351.153	-79.337
5	2043.615	19.136	-344.937
6	1555.805	-85.781	-382.836
7	1246.124	-133.105	-371.152
8	1053.971	-134.582	-322.282
9	918.246	-125.612	-270.544
10	824.818	-105.558	-214.001
11	742.912	-96.119	-173.204

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
12	712.396	-51.546	-101.466
13	680.738	-20.403	-46.604
14	629.993	-17.854	-23.188
15	597.469	-4.589	8.560
16	572.925	10.627	40.255
17	541.539	14.089	58.492
18	518.983	22.320	80.044
19	502.775	33.513	103.302
20	455.845	11.123	91.892

Procjenjeni parametri za model (3.1) su $\hat{\alpha} = 8485.119$ *i* $\hat{\beta} = 8186.386$*, a za model (3.2) su* $\hat{\alpha} = -310.9139$ *i* $\hat{\beta} = 13497.34$*. Iz gornje tablice se vidi da se najmanji* $f(k)$ *u modelu (3.1) postiže za* $k = 8$*, dok se najmanji* $f(k)$ *u modelu (3.2) postiže za* $k = 6$*. Dakle, modelom (3.2) smo se više približili stvarnom k. Iako smo ciljali na* $k = 5$ *jer smo tako generirali podatke, taj k nismo postigli. Problem leži u tome što je k-means algoritam konceptualno zamišljen tako da su klasteri kojima pridružujemo podatke sferičnog oblika. Odnosno k-means algoritam u dvodimenzionalnom slučaju nastoji "pokriti" skup podataka klasterima u obliku krugova. Dakle, u ovom primjeru algoritam ima problema s točkama koje se nalaze na rubovima pravokutnika. Stoga u sljedećim primjerima generiramo skupove podataka u obliku krugova i testiramo na njima tvrdnju Leme 3.1.1.*

Primjer 4.0.3. *Skup podataka koji sad promatramo jesu disjunktni krugovi, koji imaju jednaki radijus i svaki krug sadrži tisuću točaka.*



Slika 4.3: Podaci grupirani u tri disjunktna kruga

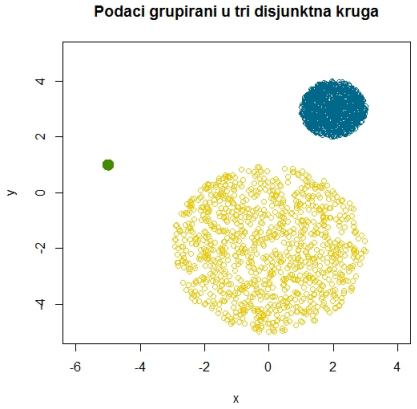
Očekujemo da će nam algoritam dati da je $k = 3$ optimalan broj klastera. Testirajući ovaj skup podataka, dobivamo sljedeće rezultate.

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
2	6926.245	4793.762	4991.455
3	374.094	-674.517	-848.924
4	328.460	-318.107	-538.673
5	281.128	-168.978	-372.474
6	247.117	-90.666	-264.131
7	236.931	-29.758	-172.635
8	204.015	-14.349	-129.290
9	157.848	-25.891	-116.143
10	130.925	-26.979	-95.615
11	182.506	44.505	-5.210

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
12	108.533	-13.728	-46.830
13	105.698	-3.847	-22.289
14	97.433	-1.651	-7.089
15	92.694	2.346	8.507
16	79.386	-3.566	12.993
17	76.893	0.272	26.201
18	73.364	2.218	36.628
19	71.232	4.864	46.983
20	67.272	5.107	54.263

Procjenjeni parametri za model (3.1) su $\hat{\alpha} = 907.5758$ i $\hat{\beta} = 6714.782$, a za model (3.2) su $\hat{\alpha} = -200.5213$ i $\hat{\beta} = 4270.622$. Iz gornje tablice se vidi da se najmanji $f(k)$ za oba modela postiže upravo za $k = 3$ kao što smo i očekivali. Dakle, optimalan broj klastera za podijeliti ovaj skup podataka je $k = 3$.

Primjer 4.0.4. *U ovom primjeru također testiramo skup podataka koji su grupirani u tri disjunktna kruga, no ovaj put radijusi krugova se razlikuju, ali broj točaka po pojedinom krugu je i dalje jednak. Dakle, imamo tisuću točaka u svakom krugu. Očekujemo da ćemo dobiti da su tri klastera optimalna za podjelu ovih podataka. Skup podataka izgleda ovako:*



Slika 4.4: Podaci grupirani u tri disjunktna kruga

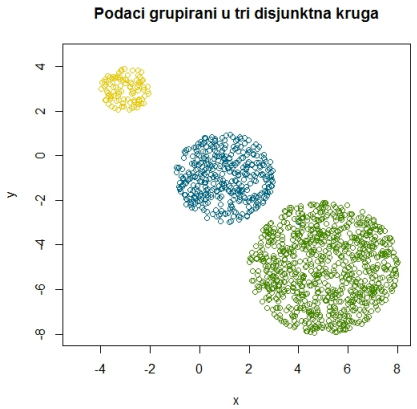
Rezultati testiranja ovakvog skupa podataka su:

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
2	19104.184	11489.058	10717.238
3	4938.470	314.621	-514.157
4	3202.426	-95.620	-783.041
5	2171.062	-385.853	-934.110
6	1724.270	-361.202	-794.039
7	1425.623	-334.195	-673.497
8	1281.568	-240.076	-503.161
9	1161.230	-178.762	-378.972
10	1084.052	-112.883	-260.529
11	1026.317	-55.068	-158.211

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
12	959.852	-26.270	-91.297
13	913.229	6.981	-25.062
14	879.605	41.288	38.049
15	855.506	75.665	97.788
16	829.288	100.310	144.927
17	634.567	-49.765	14.934
18	606.468	-38.366	44.371
19	772.740	163.098	262.121
20	476.348	-101.738	12.061

Procjenjeni parametri za model (3.1) su $\hat{\alpha} = 11154.13$ *i* $\hat{\beta} = 8152.246$*, a za model (3.2) su* $\hat{\alpha} = -416.0089$ *i* $\hat{\beta} = 17605.91$*. Iz gornje tablice se vidi da se najmanji* $f(k)$ *za oba modela postiže za* $k = 5$ *što baš i nije u skladu s očekivanjem. To se dogodilo jer gustoća točaka nije jednaka u svakom krugu, što nas motivira da u sljedećem primjeru testiramo skup podataka koji ima jednaku gustoću.*

Primjer 4.0.5. *U ovom primjeru testiramo podatke koji su podijeljeni u tri disjunktna kruga, pri čemu su radijusi različiti, a gustoća, odnosno broj točaka je proporcionalan kvadratu radijusa (*$m \approx r^2 \cdot 100$*). Očekujemo da će tri klastera biti optimalna za podjelu podataka. Takav skup podataka izgleda ovako:*



Slika 4.5: Podaci grupirani u tri disjunktna kruga

Testirajući ove podatke, dobivamao sljedeće rezultate.

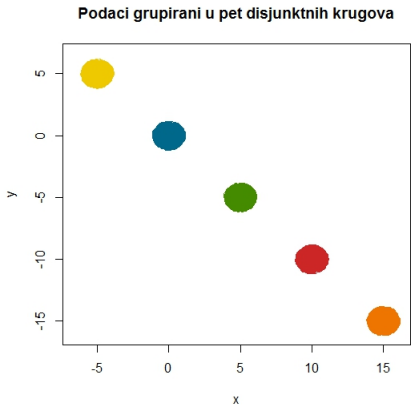
k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
2	7460.242	-908.253	578.182
3	5849.003	1085.525	1347.153
4	3406.659	139.871	94.914
5	2371.387	-95.249	-226.294
6	1918.016	-55.961	-203.623
7	1604.238	-37.813	-177.371
8	1373.477	-30.551	-153.109
9	1253.554	28.181	-74.682
10	1114.677	28.153	-54.878
11	939.593	-36.024	-100.132

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
12	873.632	-11.416	-57.902
13	799.906	-9.823	-40.082
14	751.891	5.759	-9.628
15	666.830	-24.898	-26.683
16	639.130	-5.543	5.121
17	607.468	3.891	25.964
18	578.116	10.736	43.283
19	541.622	6.364	48.547
20	510.027	3.463	54.534

Procjenjeni parametri za model (3.1) su $\hat{\alpha} = 9397.31$ *i* $\hat{\beta} = 14679.36$*, a za model (3.2) su* $\hat{\alpha} = -258.5701$ *i* $\hat{\beta} = 14281.26$*. Iz gornje tablice se vidi da se najmanji* $f(k)$ *za oba modela postiže za* $k = 2$*. To smo dobili zato jer za k-means algoritam je bolje da skupove podataka koji su blizu jedan drugome stavi u jedan klaster, nego da ih podijeli u dva klastera. Da smo više razmaknuli drugi i treći krug, dobili bi da je* $k = 3$ *optimalan broj klastera.*

Sada ćemo promatrati primjere s pet disjunktnih krugova. Testiranja provodimo za $k = 2, \ldots, 20$.

Primjer 4.0.6. *Prvo promatramo pet disjunktnih krugova, koji imaju jednake radijuse i svaki sadrži tisuću točaka. Očekujemo da ćemo dobiti* $k = 5$ *za optimalan broj klastera. Skup podataka izgleda ovako.*



Slika 4.6: Podaci grupirani u pet disjunktnih krugova

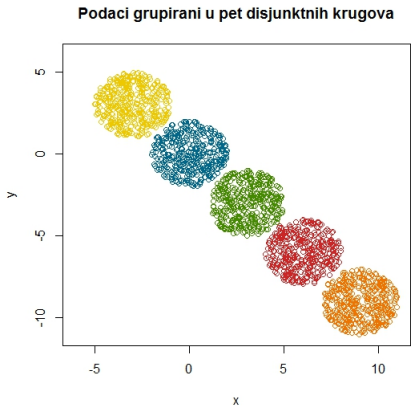
Testirajući podatke, dobivamo sljedeće rezultate:

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
2	22070.314	24182.023	10997.243
3	10287.985	7585.893	2795.248
4	6205.667	2637.886	503.097
5	2506.406	-1087.599	-2122.061
6	2319.558	-1086.436	-1592.843
7	2197.882	-973.168	-1203.042
8	2142.024	-797.775	-875.293
9	2116.816	-610.502	-602.140
10	2099.673	-437.111	-380.594
11	2082.313	-284.993	-202.663

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
12	2072.910	-143.825	-49.324
13	1893.305	-189.389	-91.223
14	2058.400	95.435	191.904
15	1873.266	17.680	109.065
16	1868.234	109.357	193.542
17	1688.648	17.235	92.934
18	1556.086	-35.900	30.574
19	1675.660	156.077	212.961
20	1547.909	94.571	141.742

Procjenjeni parametri za model (3.1) su $\hat{\alpha} = 32765.66$ *i* $\hat{\beta} = -73978.16$*, a za model (3.2) su* $\hat{\alpha} = 332.0662$ *i* $\hat{\beta} = 21482.01$*. Iz gornje tablice se vidi da se najmanji* $f(k)$ *u oba modela postiže za* $k = 5$*, što je u skladu s očekivanjem.*

Primjer 4.0.7. *Sada promatramo pet disjunktnih krugova, koji imaju jednake radijuse ali je broj točaka proporcionalan kvadratu radijusa. Očekujemo da ćemo dobiti* $k = 5$ *za optimalan broj clustera. Skup podataka izgleda ovako.*



Slika 4.7: Podaci grupirani u pet disjunktnih krugova

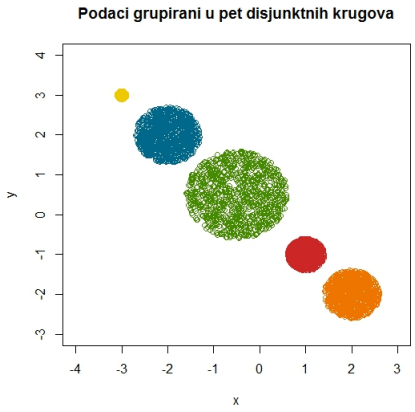
Testirajući podatke, dobivamo sljedeće rezultate:

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
2	21208.005	7176.556	6834.065
3	10468.684	1456.458	977.124
4	6711.643	80.750	-338.726
5	4003.478	-1239.662	-1582.178
6	3688.311	-646.765	-920.869
7	3333.114	-361.722	-578.582
8	3037.814	-181.423	-350.771
9	2855.711	3.668	-126.008
10	2621.982	61.984	-34.246
11	2427.891	105.709	37.973

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
12	2243.440	118.660	75.449
13	2042.769	84.470	62.562
14	1935.581	119.577	116.331
15	1653.416	-39.566	-26.336
16	1445.938	-139.628	-111.754
17	1466.940	-24.026	16.946
18	1440.423	33.406	86.163
19	1413.534	81.517	144.928
20	1209.000	-55.605	17.485

Procjenjeni parametri za model (3.1) su $\hat{\alpha} = 24984.24$ *i* $\hat{\beta} = 6157.314$*, a za model (3.2) su* $\hat{\alpha} = -273.1993$ *i* $\hat{\beta} = 29294.28$*. Iz gornje tablice se vidi da se najmanji* $f(k)$ *za oba modela postiže za* $k = 5$*, a to smo i očekivali.*

Primjer 4.0.8. *Sada promatramo pet disjunktnih krugova, koji imaju različite radijuse i svaki sadrži tisuću točaka. Očekujemo da ćemo dobiti* $k = 5$ *za optimalan broj klastera. Skup podataka izgleda ovako.*



Slika 4.8: Podaci grupirani u pet disjunktnih krugova

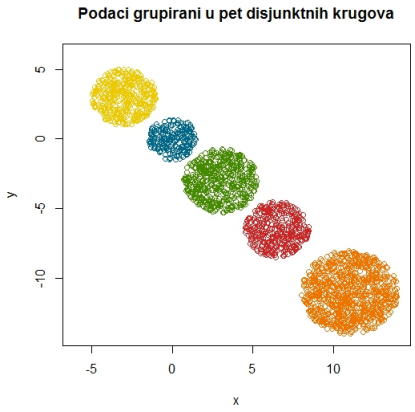
Testirajući podatke, dobivamo sljedeće rezultate:

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
2	7306.180	509.899	1970.713
3	3101.099	-611.896	-367.742
4	2094.459	-383.589	-441.069
5	1865.305	30.080	-110.236
6	1715.303	267.734	113.086
7	1648.126	457.426	312.570
8	1614.239	605.237	478.678
9	630.312	-243.860	-349.696
10	610.505	-159.892	-245.061
11	589.861	-98.333	-163.889

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
12	583.863	-37.688	-85.041
13	489.520	-76.960	-107.590
14	482.244	-37.995	-53.329
15	422.755	-58.129	-59.486
16	416.189	-30.805	-19.386
17	409.848	-7.669	15.447
18	404.343	12.693	46.543
19	400.491	31.720	75.438
20	396.406	48.010	100.826

Procjenjeni parametri za model (3.1) su $\hat{\alpha} = 6231.836$ *i* $\hat{\beta} = 14721.45$*, a za model (3.2) su* $\hat{\alpha} = -264.407$ *i* $\hat{\beta} = 11199.75$*. Iz gornje tablice se vidi da se najmanji* $f(k)$ *za model (3.1) postiže za* $k = 3$*, dok se najmanji* $f(k)$ *za model (3.2) postiže za* $k = 4$ *što nije u skladu s očekivanjem. Modelom (3.2) smo se približili očekivanom* $k = 5$*.*

Primjer 4.0.9. *Sada promatramo pet disjunktnih krugova, koji imaju različite radijuse i broj točaka proporcionalan je kvadratu radijusa. Očekujemo da ćemo dobiti* $k = 5$ *za optimalan broj klastera. Skup podataka izgleda ovako.*



Slika 4.9: Podaci grupirani u pet disjunktnih krugova

Testirajući podatke, dobivamo sljedeće rezultate:

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
2	34499.411	17719.209	11763.795
3	15677.173	2455.048	432.648
4	9429.616	-1250.222	-2069.362
5	7192.816	-1717.413	-2058.834
6	6827.020	-801.704	-926.413
7	6306.450	-357.067	-376.826
8	4855.715	-1056.639	-1024.945
9	4565.008	-746.954	-691.394
10	4360.848	-460.624	-396.148
11	4271.902	-141.532	-76.489

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
12	4231.507	162.729	223.619
13	4037.287	263.427	317.518
14	3995.910	477.233	523.100
15	3947.562	651.834	688.784
16	3928.221	828.935	856.720
17	2511.549	-413.348	-394.707
18	2484.119	-284.936	-275.253
19	2453.121	-175.831	-174.828
20	2408.658	-93.668	-101.011

Procjenjeni parametri za model (3.1) su $\hat{\alpha} = 51878.31$ *i* $\hat{\beta} = -36635.82$*, a za model (3.2) su* $\hat{\alpha} = 262.3417$ *i* $\hat{\beta} = 44946.55$*. Iz gornje tablice se vidi da se najmanji* $f(k)$ *za model (3.1) postiže za* $k = 5$*, što smo i očekivali, dok se najmanji* $f(k)$ *za model (3.2) postiže za* $k = 4$*. To nije u skladu s očekivanjem, te je u ovom slučaju model (3.2) lošiji.*

Primjer 4.0.10. *Pogledajmo sada složeniji primjer u kojem imamo 10 000 podataka i podaci su 15-dimenzionalni. Očekujemo* $k = 25$ *jer su podaci tako generirani. Testiranje provodimo za* $k = 15, 16, \ldots, 35$*. Dobiveni rezultati su:*

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
15	700089832.6	19637607	116962692
16	675092777.0	115497553	164172273
17	500151686.0	38521915	52942921
18	450206287.0	68830885	59630178
19	450111689.9	135097128	110206906
20	300139435.5	40452227	5838843
21	200153959.4	-13064884	-52885698
22	150180890.2	-23753276	-65348825
23	100166590.8	-40360371	-81114917
24	50154028.0	-61813662	-99733290
25	149302.1	-87287523	-120855361

k	$f(k)$	$f_{min}(k)$	$f2_{min}(k)$
26	*148998.4*	*-66127100*	*-94194753*
27	*148577.9*	*-47803780*	*-69509143*
28	*148210.8*	*-31882268*	*-46586769*
29	*147861.6*	*-18004959*	*-25245255*
30	*147534.2*	*-5875964*	*-5326510*
31	*147599.1*	*4751371*	*13307526*
32	*146776.9*	*14081931*	*30776051*
33	*146548.8*	*22290692*	*47186423*
34	*146407.0*	*29523868*	*62631551*
35	*145837.2*	*35905608*	*77193664*

Procjenjeni parametri za model (3.1) su $\hat{\alpha} = -9845373478$ *i* $\hat{\beta} = 300782352950$*, a za model (3.2) su* $\hat{\alpha} = -572179053$ *i* $\hat{\beta} = 17329592908$*. Iz gornje tablice se vidi da se najmanji* $f(k)$ *u oba modela postiže za* $k = 25$*, što smo i očekivali.*

Bibliografija

[1] G. Hamerly, C. Elkan , *Learning k in k-means*, 2003.

[2] I. Čermak, *K-means clustering u prostoru Minkowskog (diplomski rad)*, Prirodoslovno-matematički fakultet, 2011.

[3] *Wikipedia, the free encyclopedia*, http://en.wikipedia.org

Sažetak

U ovom radu obrađena je metoda za određivanje k u k-means algoritmu, što je vrlo česti problem u klasteriranju podataka. Pokazano je da funkcija cilja u k-means algoritmu konvergira. Nadalje, u potpoglavlju (1.3) pokazano je da su koordinate težišta skupa jednake aritmetičkim sredinama odgovarajućih koordinata elemenata tog skupa, te se u slučaju standardnog k-means algoritma novi centri određuju na takav način. U lemama smo dokazali da ako su podaci ekvidistantni, odnosno uniformno distribuirani, onda se optimizirana funkcija cilja ponaša odprilike kao $C \cdot \frac{1}{k^2}$. Pretpostavka o ekvidistantnosti, odnosno uniformnoj distribuiranosti podataka, nije tako jaka kao pretpostavka o normalnoj distribuiranosti podataka u G-means algoritmu. Ovime smo pokazali da uz relativno nezahtjevne prepostavke, lijepo možemo opisati ponašanje funkcije cilja u ovisnosti o broju klastera k. Na kraju, rezultati testiranja vrlo dobro potvrđuju naša teorijska saznanja o ponašanju funkcije cilja.

Summary

In this thesis, we have studied method for determining k in k-means algorithm, which is a well known problem. It is shown that the objective function coverges. Furthermore, a couple of lemmas proved that, if data follow a uniform distribution, then the objective function behaves as $C \cdot \frac{1}{k^2}$. Assumption about data is not as strong as assumption in G-means algorithm which requires normal distribution of data. We have shown that with relatively limited assumptions, we can neatly describe behaviour of the objective function with respect to k - the number of clusters. Finally, a series of tests on simulated data sets confirmed our theoretical results about behaviour of the objective function.

Životopis

Rođena sam u Zagrebu 1989. godine. Od 1995. do 2003. pohađala sam osnovnu školu u Jastrebarskom. Nakon toga, 2003. godine upisala sam V. gimnaziju u Zagrebu. Srednjoškolsko obrazovanje završila sam 2007. godine, te sam potom upisala preddiplomski studij Matematike na Prirodoslovno- matematičkom fakultetu u Zagrebu. Preddiplomski studij završila sam 2012. godine, te sam iste godine upisala diplomski studij Matematičke statistike na istom fakultetu. U svrhu dodatnog obrazovanja pohađala sam tečajeve programskog jezika SAS u Sveučilišnom računskom centru u Zagrebu.