K-MEANS METODA ZA PARTICIONIRANJE PODATAKA

Seminarski rad

Ivančica Mirošević

Fakultet elektrotehnike, strojarstva i brodogradnje
Sveučilište u Splitu

Smjernice

- Algoritam
- Dualnost ciljne funkcije
- Složenost algoritma
- Algoritam prve varijacije
- Primjer

K-means

Neka je $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \cdots, m$.
Cilj k-meansa:
Za unaprijed zadani $k > 1$ odrediti particiju $\pi = \{C_1, C_2, ..., C_k\}$ skupa X koja minimizira vrijednost ciljne funkcije

$$J = J(\pi) = \sum_{i=1}^{k} \sum_{x \in C_i} ||\mathbf{x} - \mathbf{c}_i||_2^2,$$

gdje je

$$\mathbf{c}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x} \tag{1}$$

predstavnik (središte) skupine C_i, $i = 1, \cdots, k$.

Algoritam

1. **Inicijalizacija**. Zadaj $\{\mathbf{c}_i^0\}_{i=1}^k$. Postavi brojač $l = 0$;
2. **Pridruživanje.** Za svaki $\mathbf{x} \in X$ odredi $sk(\mathbf{x})$ takav da je

$$||\mathbf{x} - \mathbf{c}_{sk(\mathbf{x})}^l||_2 = \min_{j \in \{1,2,\cdots,k\}} ||\mathbf{x} - \mathbf{c}_j^l||_2.$$

   Definiraj skupine:

$$C_i^{(l+1)} = \{\mathbf{x} \in X : sk(\mathbf{x}) = i\}, i = 1, \cdots, k.$$

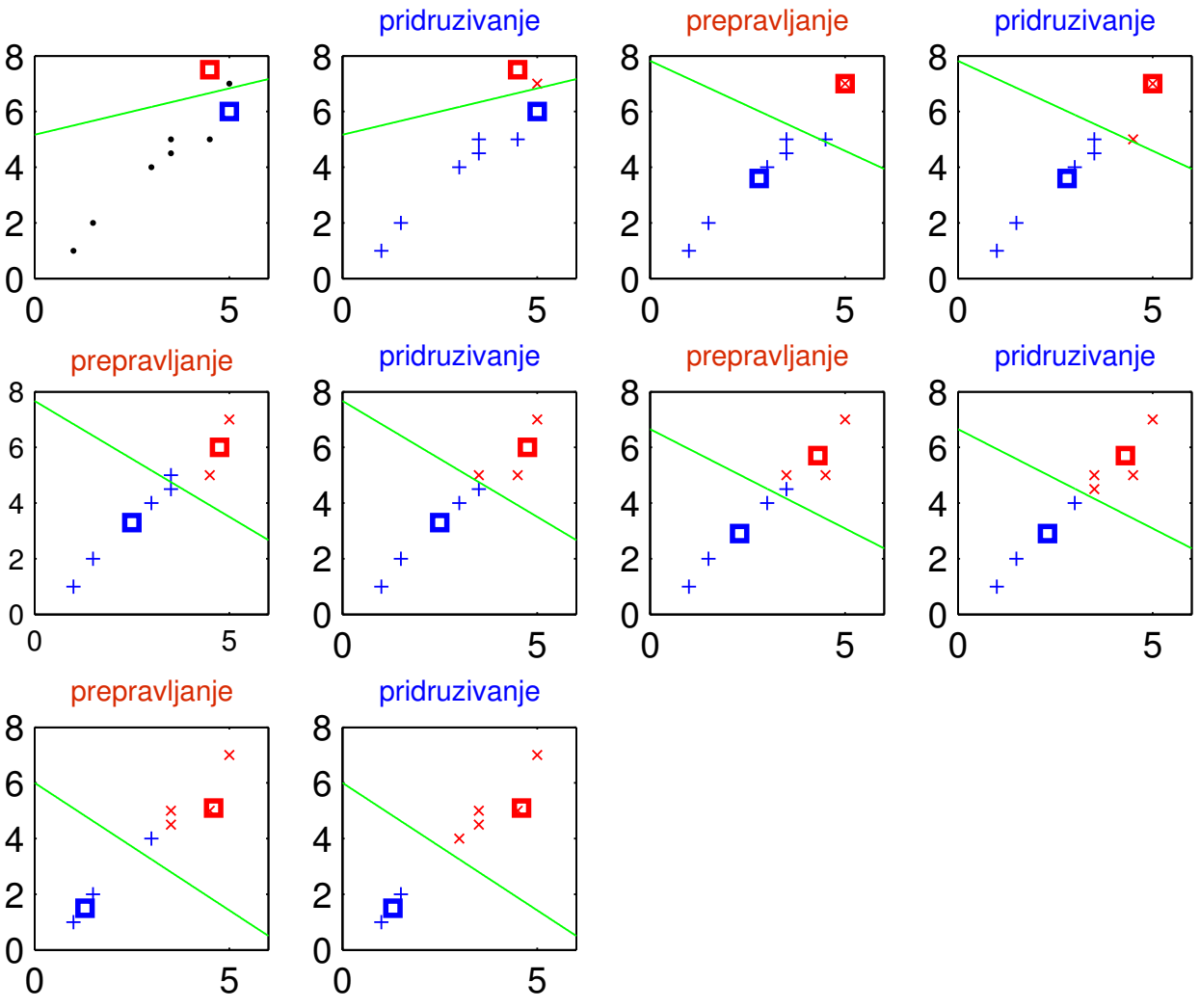
3. **Prepravljanje**. Izračunaj:

$$\mathbf{c}_i^{l+1} = \frac{\sum\limits_{\mathbf{x} \in C_i^{(l+1)}} \mathbf{x}}{|C_i^{(l+1)}|},$$

$$J^{(l+1)} = \sum_{i=1}^{k} \sum_{x \in C_i^{(l+1)}} ||\mathbf{x} - \mathbf{c}_i^{l+1}||_2^2.$$

   Stavi $l = l + 1$; Ponavljaj korake 2 i 3 dok se vrijednost J ne prestane smanjivati.

Primjer


pridruzivanje
prepravljanje
pridruzivanje
prepravljanje
pridruzivanje
prepravljanje
pridruzivanje
prepravljanje
pridruzivanje

Konvergencija algoritma

Teorem 1 *Vrijednost ciljne funkcije k-meansa monotono se smanjuje u svakoj iteraciji.*

Dualnost ciljne funkcije (1)

Ciljna funkcija k-meansa može se shvatiti kao $\mathrm{trag}\,(S_W)$ gdje je

$$S_W = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{c}_i)\,(\mathbf{x} - \mathbf{c}_i)^T .$$

Dulanost ciljne funkcije (2)

Definirajmo matrice S_B i S_T s

$$S_B = \sum_{i=1}^{k} |C_i| \left(\mathbf{c}_i - \mathbf{c}\right) \left(\mathbf{c}_i - \mathbf{c}\right)^T \quad \text{i}$$

$$S_T = \sum_{\mathbf{x} \in X} \left(\mathbf{x} - \mathbf{c}\right) \left(\mathbf{x} - \mathbf{c}\right)^T,$$

gdje je $\mathbf{c}$ srednja vrijednost cijelog skupa podataka. Vrijedi

$$S_T = S_W + S_B.$$

Dualnost ciljne funkcije (3)

$$S_T = \frac{1}{2m} \sum_{i=1}^{m} \sum_{j=1}^{m} (\mathbf{x}_i - \mathbf{x}_j) (\mathbf{x}_i - \mathbf{x}_j)^T ,$$

$$S_W = \sum_{i=1}^{k} \frac{1}{2|C_i|} \sum_{\mathbf{x} \in C_i} \sum_{\mathbf{y} \in C_i} (\mathbf{x} - \mathbf{y}) (\mathbf{x} - \mathbf{y})^T ,$$

i $$S_B = \frac{1}{2m} \sum_{i=1}^{k} \sum_{j=1}^{k} |C_i||C_j| (\mathbf{c}_i - \mathbf{c}_j) (\mathbf{c}_i - \mathbf{c}_j)^T .$$

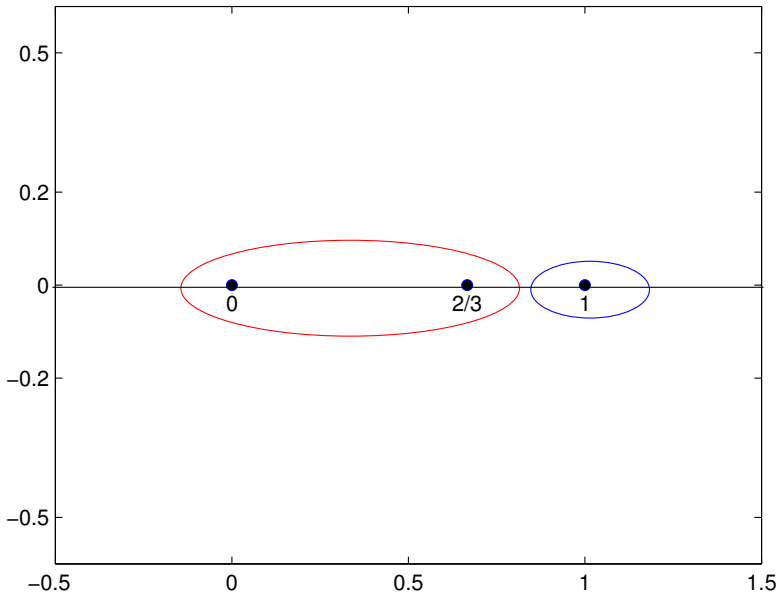
Složenost algoritma

- Pridruživanje: mk računanja udaljenosti, mk uspoređivanja za pronalaženje najbližeg predstavnika. Ukupna složenost: $O\left(mkn\right)$;
- Prepravljanje: mn zbrajanja i kn dijeljenja za računanje predstavnika, m zbrajanja udaljenosti izračunatih u prvom koraku za računanje vrijednosti ciljne funkcije. Složenost: $O\left(mn\right)$ ($k \leq m$).

Ukupna složenost: $O\left(mknt\right)$, t je broj iteracija.

Još uvijek nisu dokazane nikakve smislene međe za broj iteracija k-meansa u općem slučaju, odnosno za particioniranje skupa $X \subset \mathbb{R}^n$, $n > 1$, bez ikakvih dodatnih ograničenja.

Primjer zaustavljanja algoritma u lokalnom minimumu



$J\left(\{C_1^0, C_2^0\}\right) = \frac{2}{9}$ za $C_1^0 = \{0, \frac{2}{3}\}$ i $C_2^0 = \{1\}$.

$J\left(\{C_1^1, C_2^1\}\right) = \frac{1}{18}$ za $C_1^1 = \{0\}$ i $C_2^1 = \{\frac{2}{3}, 1\}$.

Prva varijacija particije

Definicija 1 *Prva varijacija particije* $\pi = \{C_1, \cdots, C_k\}$ *skupa* X *je particija* $\pi' = \{C'_1, \cdots, C'_k\}$ *koja se dobije pomicanjem jedne točke* $\mathbf{x} \in X$ *iz skupine* $C_i \in \pi$ *u skupinu* $C_j \in \pi$*. Skup svih prvih varijacija particije* $\pi = \{C_1, \cdots, C_k\}$ *označavamo s* $\mathcal{V}(\pi)$*.*

Definicija 2 *Sljedeća prva varijacija* π^* *particije* π *je prva varijacija particije* π *skupa* X *takva da je za svaku prvu varijaciju* π' *skupa* X

$$J(\pi^*) \leq J(\pi'),$$

Algoritam prve varijacije

1. Zadaj početnu particiju $\pi^{(0)} = \{C_1^{(0)}, \cdots, C_k^{(0)}\}$. Postavi brojač iteracija $l = 0$.

2. Generiraj sljedeću prvu varijaciju $\pi^{(l)*}$.

   Ako je $J\left(\pi^{(l)*}\right) - J\left(\pi^{(l)}\right) < 0$, postavi $\pi^{(l+1)} = \pi^{(l)*}$, povećaj l za 1, i vrati se na korak 2.

3. Stani.

Usporedba k-meansa i algoritma prve varijacije (1)

Neka je zadana biparticija $\pi = \{Z, Y\}$ skupa $X \subset \mathbb{R}^n$, $Z = \{\mathbf{z}_1, \cdots, \mathbf{z}_n\}$ i $Y = \{\mathbf{y}_1, \cdots, \mathbf{y}_m\}$. Želimo utvrditi treba li jedan vektor, npr. $\mathbf{z}_n$, premjestiti iz Z u Y. Neka su potencijalne skupine

$$Z^- = \{\mathbf{z}_1, \cdots, \mathbf{z}_{n-1}\} \quad \text{i} \quad Y^+ = \{\mathbf{y}_1, \cdots, \mathbf{y}_m, \mathbf{z}_n\}.$$

K-means provjerava vrijednost

$$\Delta_{km} = ||\mathbf{z}_n - \mathbf{c}\,(Y)\,||_2^2 - ||\mathbf{z}_n - \mathbf{c}\,(Z)\,||_2^2.$$

Ako je $\Delta_{km} < 0$, k-means pomiče $\mathbf{z}_n$ iz Z u Y. Inače $\mathbf{z}_n$ ostaje u Z.

Usporedba k-meansa i algoritma prve varijacije (2)

Stvarna promjena u vrijednosti ciljne funkcije je

$$\Delta_{pv} = \frac{m}{m+1}||\mathbf{z}_n - \mathbf{c}\,(Y)\,||_2^2 - \frac{n}{n-1}||\mathbf{z}_n - \mathbf{c}\,(Z)\,||_2^2$$

Razlika

$$\Delta_{km} - \Delta_{pv} = \frac{1}{m+1}||\mathbf{z}_n - \mathbf{c}\,(Y)\,||_2^2 + \frac{1}{n-1}||\mathbf{z}_n - \mathbf{c}\,(Z)\,||_2^2 \geq 0.$$

je zanemariva kada su skupine Z i Y velike.

Međutim, $\Delta_{km} - \Delta_{pv}$ može postati bitna kod malih skupina.

Primjer

Stavimo li

$$Z = \{0, \tfrac{2}{3}\},\ Y = \{1\} \text{ i } z_n = \tfrac{2}{3},$$

tada je

$$\Delta_{km} = 0,$$

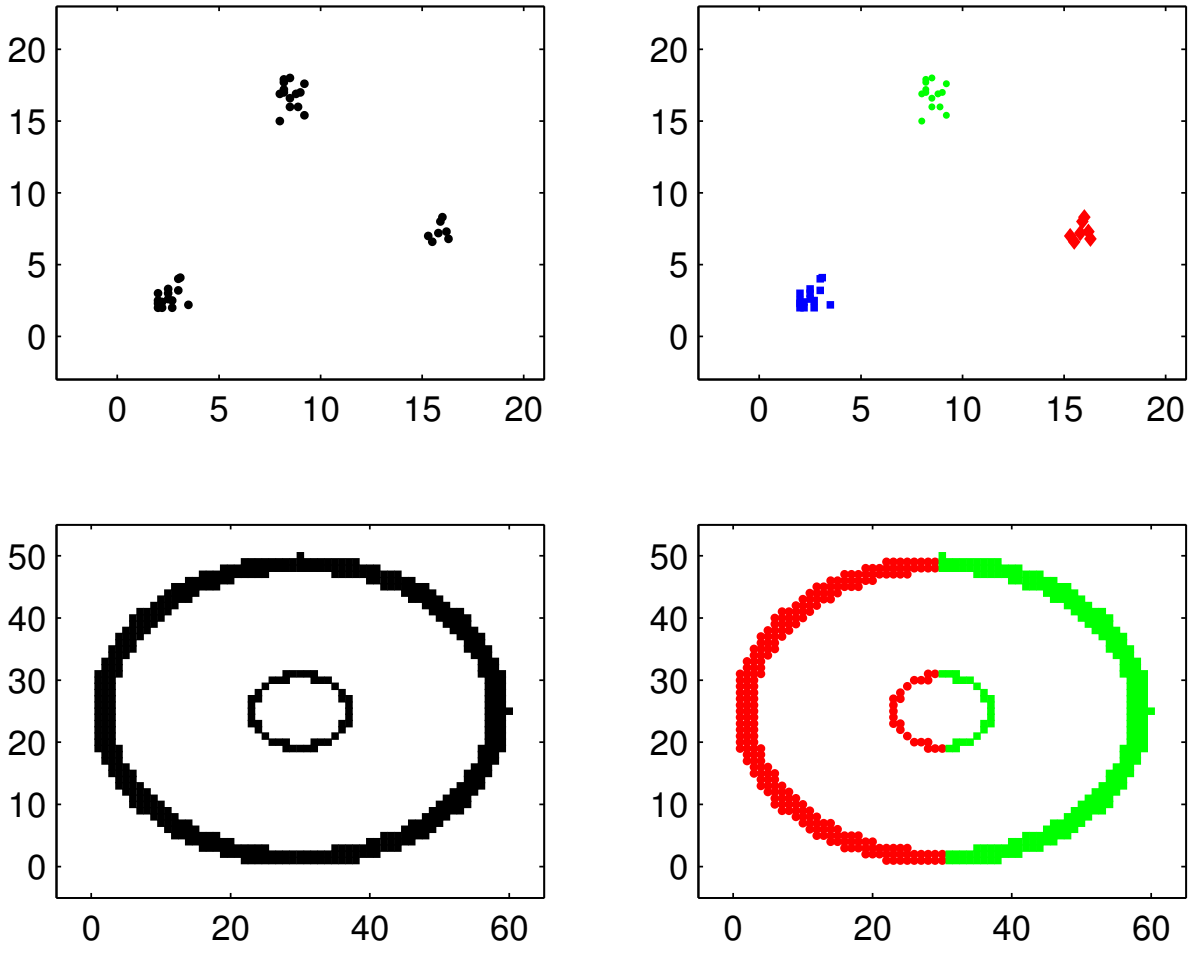
$$\Delta_{pv} = -\frac{3}{18}$$

K-means propušta optimalnu particiju $\{Z^+, Y^-\}$, za razliku od algoritma prve varijacije.

K-means algoritam poboljšan prvom varijacijom

1. Zadaj početnu particiju $\pi^{(0)} = \{C_1^{(0)}, \cdots, C_k^{(0)}\}$. Postavi brojač iteracija $l = 0$.
2. Generiraj sljedeću k-means particiju $\pi^{(l)'}$. Ako je $J\left(\pi^{(l)'}\right) - J\left(\pi^{(l)}\right) < 0$, postavi $\pi^{(l+1)} = \pi^{(l)'}$, povećaj l za 1, i vrati se na korak 2.
3. Generiraj sljedeću prvu varijaciju $\pi^{(l)*}$. Ako je $J\left(\pi^{(l)*}\right) - J\left(\pi^{(l)}\right) < 0$, postavi $\pi^{(l+1)} = \pi^{(l)*}$, povećaj l za 1, i vrati se na korak 2.
4. Stani.

Primjer particioniranja MatLabovim k-meansom



Algoritam još uvijek ovisi o inicijalizaciji