Flat clustering

Algoritmul K-means

noiembrie 2017

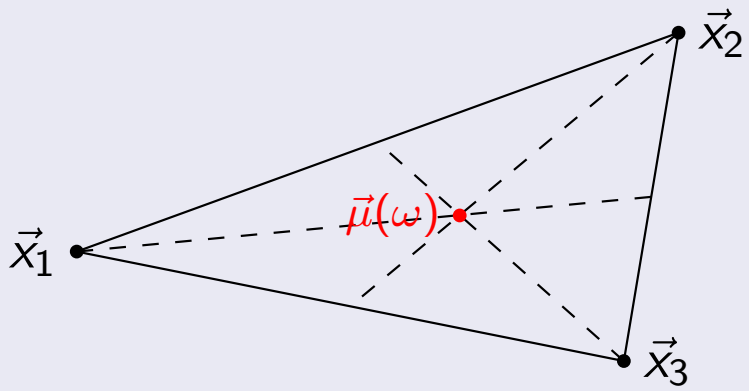
Flat clustering

- Se presupune dată o colecție de N documente reprezentate ca o mulțime de puncte $D = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\}$ într-un spațiu vectorial.
- Se dorește împărțirea mulțimii D în K submulțimi distincte $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$:

  $$\omega_1 \cup \omega_2 \cup \ldots \cup \omega_K = D \text{ și } \omega_i \cap \omega_j = \emptyset \text{ dacă } i \neq j$$

  Mulțimile $\omega_1, \ldots, \omega_K$ se numesc clustere. Fiecare cluster $\omega \in \Omega$ are un centroid $\vec{\mu}(\omega)$ care se calculează cu formula $\vec{\mu}(\omega) = \frac{\sum_{\vec{x} \in \omega} \vec{x}}{|\omega|}$

Exemplu: centroidul unui cluster de 3 documente $\omega = \{\vec{x}_1, \vec{x}_2, \vec{x}_3\}$



Algoritmul K-means

- Suma reziduală de pătrate (*engl.* residual sum of squares) a unui cluster $\omega_k \in \Omega$ este

$$RSS_k := \sum_{\vec{x} \in \omega_k} |\vec{x} - \vec{\mu}(\omega_k)|^2$$

- Suma reziduală de pătrate a unei partiții $\Omega = \{\omega_1, \ldots, \omega_K\}$ este

$$RSS := \sum_{k=1}^{k} RSS_k$$

- Algoritmul K-means are ca obiectiv să minimizeze valoarea lui RSS pentru o mulțime dată de documente $D = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\}$.

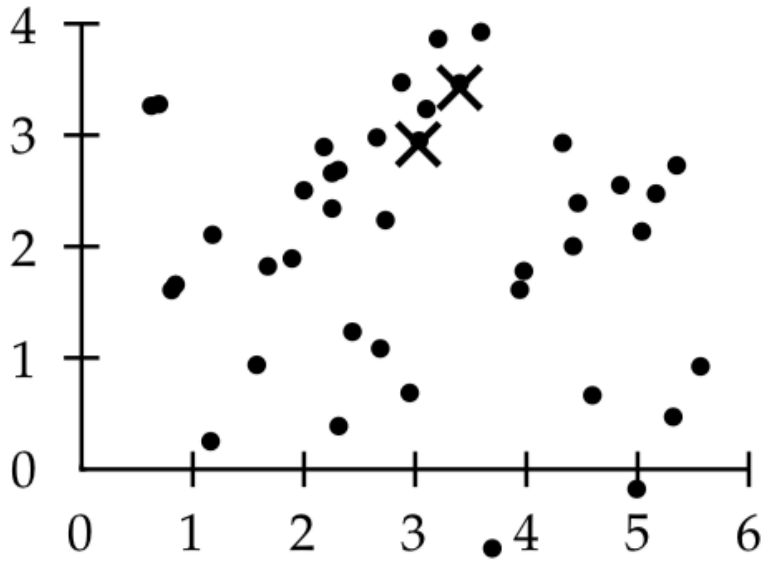
Algoritmul K-means
Pseudocod

K-MEANS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$
1 $(\vec{s}_1, \ldots, \vec{s}_K)$:=SELECTRANDOMSEEDS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$
2 **for** $k := 1$ **to** K **do**
3 $\quad \vec{\mu}_k := \vec{s}_k$
4 **while** *criteriul de terminare nu este satisfăcut* **do**
5 $\quad$ **for** $k := 1$ **to** K **do**
6 $\quad\quad \omega_k := \emptyset$
7 $\quad\quad$ **for** $n := 1$ **to** N **do**
8 $\quad\quad\quad j :=$**arg min**$_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
9 $\quad\quad\quad \omega_j := \omega_j \cup \{\vec{x}_n\}$ (*reatribuirea vectorilor în clustere*)
10 $\quad$ **for** $k := 1$ **to** K **do**
11 $\quad\quad \vec{\mu}_k := \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$
12 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

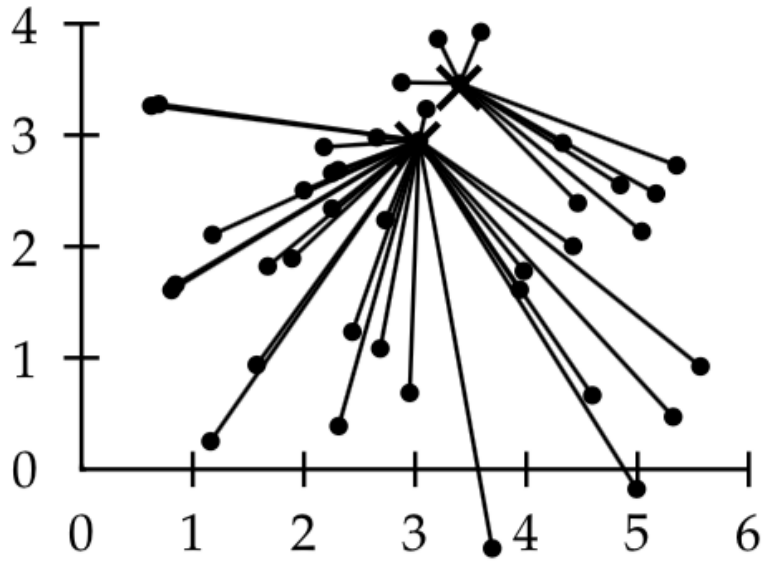
Observații

- Mai întâi, se aleg la întâmplare K centroizi inițiali de clustere (engl. *seeds*)
- algoritmul repetă modificarea centrelor inițiale încât să minimizeze *RSS*: la fiecare iterație: (1) se reatribuie fiecare vector $\vec{x}$ în clusterul centroidului celui mai apropiat de $\vec{x}$ și apoi (2) se recalculează coordonatele centroizilor

Evoluția algoritmului

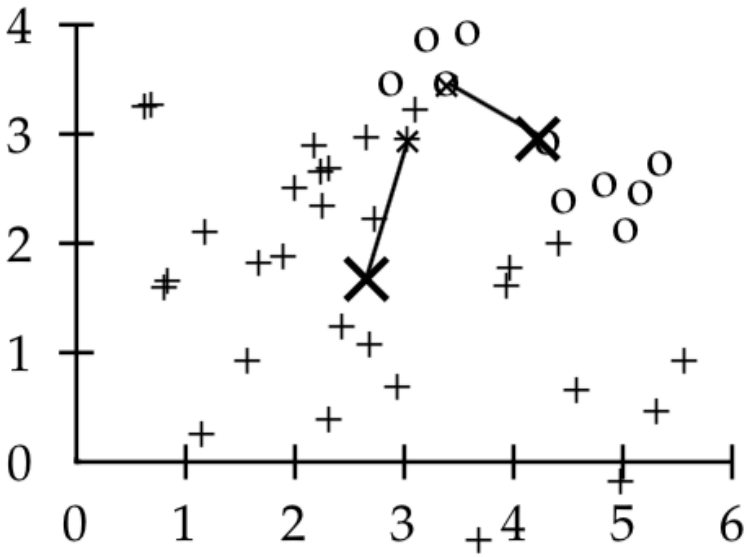


selection of seeds

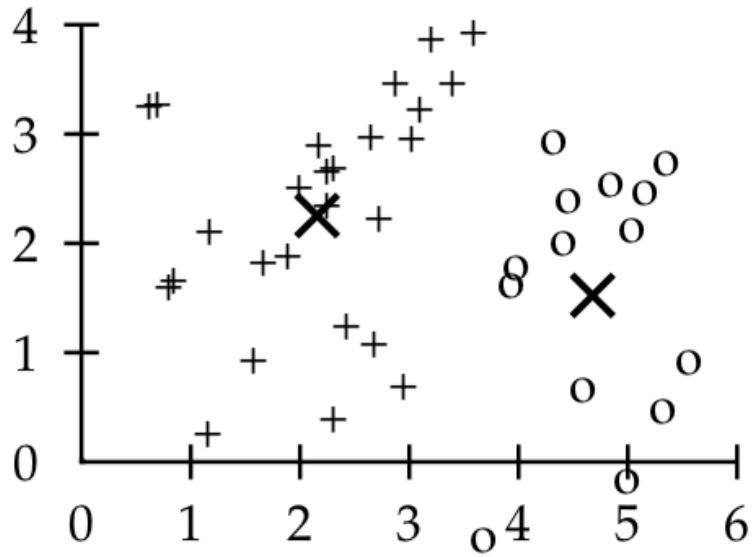


assignment of documents (iter. 1)

Evoluția algoritmului în 9 iterații:

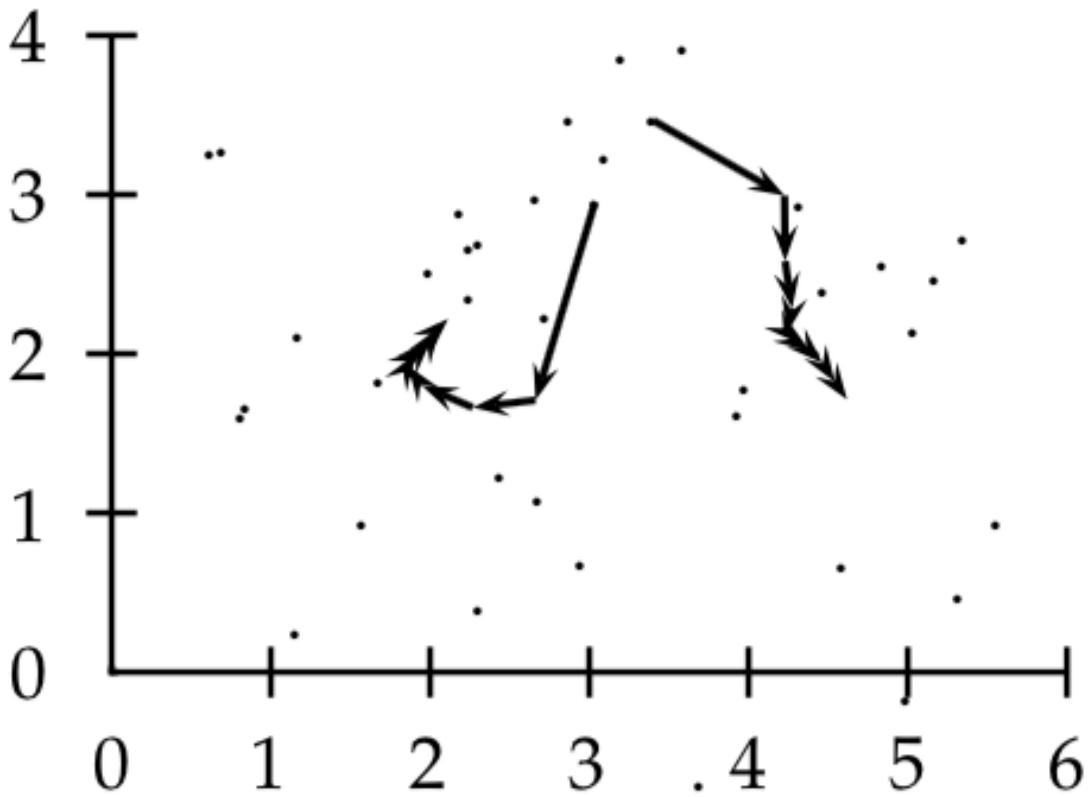


recomputation/movement of $\vec{\mu}$'s (iter. 1)



$\vec{\mu}$'s after convergence (iter. 9)

Evoluția algoritmului în 9 iterații:



1. După un număr prestabilit de iterații
2. Când se ajunge la o iterație care nu modifică atribuirea documentelor în clustere
3. Când centroizii $\vec{\mu}_k$ nu se mai schimbă de la o iterație la alta
4. Când *RSS* devine mai mică dacât o valoare prestabilită
5. Când diferența dintre două *RSS*-uri consecutive este mai mică decât o valoare θ

1. După un număr prestabilit de iterații
2. Când se ajunge la o iterație care nu modifică atribuirea documentelor în clustere
3. Când centroizii $\vec{\mu}_k$ nu se mai schimbă de la o iterație la alta
4. Când *RSS* devine mai mică dacât o valoare prestabilită
5. Când diferența dintre două *RSS*-uri consecutive este mai mică decât o valoare θ

REMARCĂ: K-means converge în sensul că dacă definim

$$RSS_k(\vec{v}) = \sum_{\vec{x} \in \omega_k} |\vec{v} - \vec{x}|^2 \quad \text{și} \quad RSS(\vec{v}_1, \ldots, \vec{v}_K) = \sum_{k=1}^{K} RSS_k(\vec{v}_k)$$

atunci $(\vec{\mu}_1, \ldots, \vec{\mu}_K)$ este punct de minim local pentru $RSS(\vec{v}_1, \ldots, \vec{v}_K)$

(*vezi slide următor*)

Presupunem că documentele sunt puncte în spațiul euclidean M-dimensional $\mathbb{R}^M$

$$\Rightarrow RSS_k(\vec{v}) = \sum_{\vec{x} \in \omega_k} \sum_{m=1}^{M} (v_m - x_m)^2$$

$$\Rightarrow \frac{\partial RSS_k(\vec{v})}{\partial v_m} = \sum_{\vec{x} \in \omega_k} 2(v_m - x_m)$$

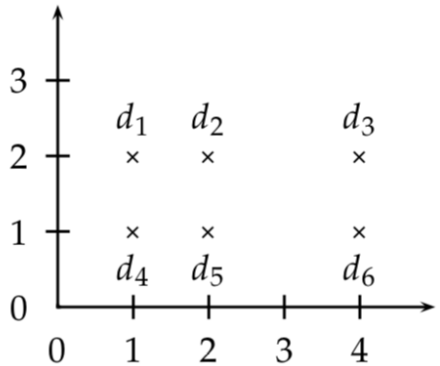
Proprietăți ale algoritmului K-means

1. În pașii 8-9, $RSS(\vec{v}_1, \ldots, \vec{v}_K) = \sum_{k=1}^{K} \sum_{\vec{x} \in \omega_k} |\vec{v}_k - \vec{x}|^2$ scade pentru $\vec{v} = (\vec{\mu}_1, \ldots, \vec{\mu}_k)$ deoarece fiecare document $\vec{x} \in \omega_k$ este reatribuit în clusterul ω_j pentru care $|\vec{v}_j - \vec{x}| \leq |\vec{v}_k - \vec{x}|$
2. În pașii 10-11 se recalculează fiecare $\vec{\mu}_k$ a.î. fiecare $RSS_k(\vec{v})$ să ia valoarea minimă pentru $\vec{v} = \vec{\mu}_k$.

Este de dorit să se calculeze K centroizi $(\vec{\mu}_1, \ldots, \vec{\mu}_K)$ care sunt un minim global pentru *RSS*

- Algoritmul K-means garantează calculul unui minim local
  - Uneori, minim local $\neq$ minim global
  - Rezultatul calculat depinde de centroizii inițiali aleși
    - selecția unui centroid inițial îndepărtat de celelalte documente produce generarea de **clustere singleton**, sau clustere vide.

EXEMPLU:



- K-means pt. centroizii inițiali d_2 și d_5 converge la $\{\{d_1, d_2, d_3\}, \{d_4, d_5, d_6\}\}$
- K-means pt. centroizii inițiali d_2 și d_3 converge la $\{\{d_1, d_2, d_4, d_5\}, \{d_3, d_6\}\}$

Complexitatea algoritmului

- Distanța dintre două puncte în $\mathbb{R}^M$ durează ...
- Pașii 7-9 (reatribuirea în clustere) durează ...
- Pașii 10-11 (recalcularea centroizilor) durează ...
- K-means care efectuează I iterații durează ...

Determinarea numărului K de clustere

Metode euristice

Se presupune dată o mulțime de N documente: $D = \{\vec{x}_1, \ldots, \vec{x}_N\}$

- Algoritmul K-means ia ca **parametru de intrare** valoarea lui K și calculează K centroizi pentru K clustere $\omega_1, \ldots, \omega_K$
- Uneori e greu de să se estimeze o valoare bună pentru K

Problemă: **Cum putem găsi o valoare plauzibilă pentru K?**

Se presupune dată o mulțime de N documente: $D = \{\vec{x}_1, \ldots, \vec{x}_N\}$

- Algoritmul K-means ia ca **parametru de intrare** valoarea lui K și calculează K centroizi pentru K clustere $\omega_1, \ldots, \omega_K$
- Uneori e greu de să se estimeze o valoare bună pentru K

Problemă: **Cum putem găsi o valoare plauzibilă pentru K?**

EURISTICA 1: RSS să aibe valoare minimă:

$\Rightarrow$ Dacă $K = N$ atunci $\omega_i = \{\vec{x}_i\}$ pentru $1 \leq i \leq K$ și $RSS = 0$. Deși $RSS = 0$, acesta nu este un clustering optim: sunt prea multe clustere!

Se presupune dată o mulțime de N documente: $D = \{\vec{x}_1, \ldots, \vec{x}_N\}$

- Algoritmul K-means ia ca **parametru de intrare** valoarea lui K și calculează K centroizi pentru K clustere $\omega_1, \ldots, \omega_K$
- Uneori e greu de să se estimeze o valoare bună pentru K

Problemă: **Cum putem găsi o valoare plauzibilă pentru K?**

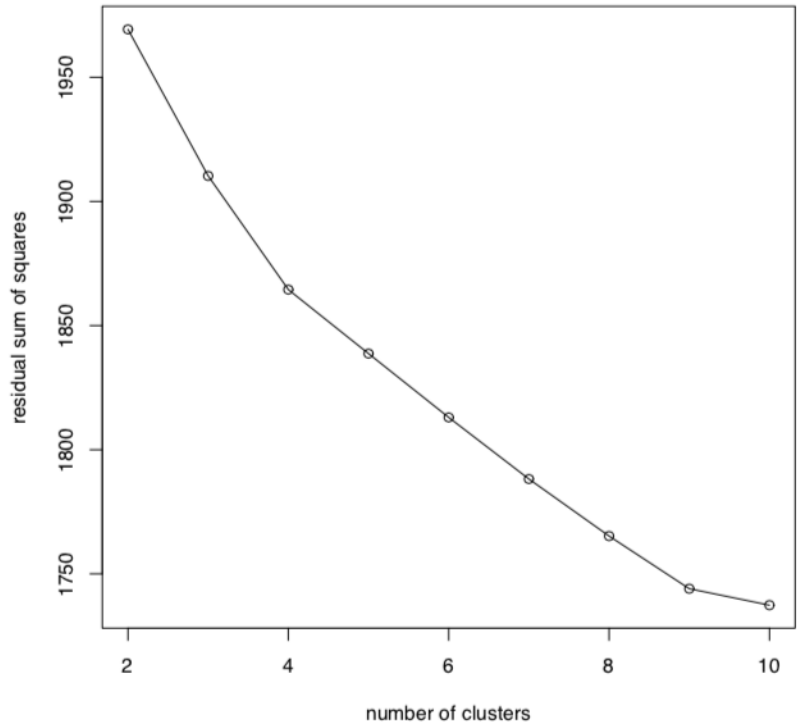
EURISTICA 1: RSS să aibe valoare minimă:

$\Rightarrow$ Dacă $K = N$ atunci $\omega_i = \{\vec{x}_i\}$ pentru $1 \leq i \leq K$ și $RSS = 0$. Deși $RSS = 0$, acesta nu este un clustering optim: sunt prea multe clustere!

EURISTICA 2: Pentru mai multe valori valori ale lui K, se calculează $\widehat{RSS}_{\min}(K) = \min\{rss_1, \ldots, rss_i\}$ unde rss_j $(1 \leq j \leq i)$ este valoarea lui RSS pentru un set de K centroizi inițiali $(\vec{s}_{j,1}, \ldots, \vec{s}_{j,K})$. i are o valoare prestabilită, de ex., $i = 10$.
Se alege K pentru care vectorul $(\widehat{RSS}_{\min}(K+1) - \widehat{RSS}_{\min}(K), 1)$ începe să își mențină direcția.

Determinarea numărului K de clustere

Estimarea valorii lui K cu euristica 2



Exemplu ilustrat pentru o mulțime de 1203 documente Reuters-RCV1

- curba $\widehat{RSS}_{\text{min}}$ se netezește semnificativ după $K = 4$ și $K = 9$ $\Rightarrow$ 4 și 9 sunt valori plauzibile pentru K.
- pentru $K = 4$, algoritmul K-means calculează centroizi în jurul categoriilor *China*, *Germany*, *Russia* și *Sports*.

- Minimizarea distorsiunii ca măsură a complexității modelului

$$K = \arg\,\min_K(RSS_{\min}(K) + \lambda\, K)$$

unde $0 \leq \lambda \leq 1$ este un factor de pondere.

- Criteriul AIC (engl. *Aikake Information Criterion*) are forma generală

$$K = \arg\,\min_K(-2\, L(K) + 2\, q(K))$$

unde ...

Pentru algoritmul Kmeans, are forma specială

$$K = \arg\,\min_K(RSS_{\min}(K) + 2\, M\, K)$$

Bibliografie

Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze: Capitolul 16 din An Introduction to Information Retrieval.
Ediție online (c) 2009 Cambridge UP.