Învățare Automată

Moișanu-Costinescu Ștefan

Decembrie 2020

1 Exercițiul 46, pagina 819: k-means++

1.1 Comentariu:

În exerciţiul de faţă vom considera că la faza de iniţializare algoritmul Kmeans alege centroizii în mod arbitrar dintre instanţele de clusterizat.
Dacă datele de clusterizat sunt [a priori] bine separate în K clustere, atunci este foarte posibil ca la finalul iniţializării să existe [măcar] un cluster din care algoritmul nu a selectat niciun punct. În astfel de situaţii, algoritmul K-means nu va produce clusterele dorite de noi.
În schimb, varianta K-means++ a algoritmului K-means, propusă de David Arthur şi Sergei Vassilvitskii în 2007, încearcă să selecteze pentru poziţiile iniţiale ale celor K centroizi instanţe care sunt [pe cât se poate] mai distanţate unele de altele. În acest fel, pot fi selectate, cu o probabilitate [destul de] mare, instanţe din toate clusterele.

1.2 Formalizare:

K-means++ face iniţializarea centroizilor în maniera următoare:

1. Alege primul centroid, μ_1, în manieră uniform aleatorie dintre instanţele de clusterizat, $x_1, \ldots, x_n$. Cu alte cuvinte, alegem mai întâi un indice i în mod uniform aleatoriu din mulţimea {1, . . . , n} şi fixăm $\mu_1 = x_i$.
2. Pentru j = 2, . . . ,K:
   - Pentru fiecare instanţă x_i, calculează distanţa D_i până la cel mai apropiat centroid ales / fixat la o iteraţie anterioară:

   $$D_i = \min_{j\prime=1,\ldots,j-1} ||x_i - \mu_{j\prime}||$$

   - Alege centroidul μ_j în mod aleatoriu dintre instanţele $x_1, \ldots, x_n$, cu probabilitate proporţională cu $D_1^2, \ldots, D_{2n}^2$. Altfel spus, alegem un indice i în mod aleatoriu din mulţimea {1, . . . , n} cu probabilităţi egale cu $\frac{D_1^2}{\sum_{i\prime=1}^n D_{i\prime}^2}$, ..., $\frac{D_n^2}{\sum_{i\prime=1}^n D_{i\prime}^2}$, şi fixăm $\mu_j = x_i$
3. Returnează $\mu = (\mu_1, \ldots, \mu_k)$, setul de asignări ale poziţiilor iniţiale ale centroizilor clusterelor pentru algoritmul lui Lloyd (K-means).

Vom ilustra acum diferenţa dintre K-means++ şi K-means [la iniţializare], folosind un set de date simplu, format din cinci puncte în planul euclidian bidimensional: A(-1, 0), B(1, 0), C(0, 1), D(3, 0) şi E(3, 1):

a Presupunem că aplicăm algoritmul K-means cu K = 2, făcând iniţializarea centroizilor cu instanţe din mulţimea pe care tocmai am precizat-o. Presupunem că a fost deja ales centroidul μ_1 = A. Dacă selecţia următorului centroid (μ_2) se face în manieră uniform aleatorie — aşa cum procedează îndeobşte algoritmul K-means — din mulţimea {B, C, D, E}, care este probabilitatea ca μ_2 să fie din submulţimea {B, C}? Justificaţi.

b Aplicăm acum pe acelaşi set de date algoritmul K-means++, tot cu K = 2. Presupunem, ca şi la punctul a, că a fost selectat centroidul μ_1 = A. Care este acum probabilitatea ca μ_2 să fie selectat din submulţimea {B, C}? Rezultă oare într-adevăr o îmbunătăţire semnificativă faţă de [iniţializarea făcută de] algoritmul K-means? Justificaţi riguros.

2 Rezolvare

2.1 Subpunctul *a.*

Dupa ce algoritmul alege μ_1 = A, ramane sa aleaga aleator uniform μ_2 din multimea {B, C, D, E}.
$\Rightarrow$ probabilitatea ca $\mu_2 \in$ {B, C} este de $\frac{2}{4} = 50\%$.

2.2 Subpunctul *b.*

Pentru a afla P($\mu_2 \in$ {B, C}) folosind k-means++ voi afla distributia de probabilitate constituita de acest algoritm la fiecare iteratie.

μ_2 = A (aleator)

Calculez D_A, D_B, D_C, D_D, D_E:

D_A = D (A, μ_1) = 0 (μ_1 si μ_2 nu pot coincide, dar deoarece dist = 0 oricum va avea probabilitate 0 de a fi ales ia A)

$D_B = 2$

$D_C = \sqrt{2}$

$D_D = 4$

$D_E = \sqrt{17}$

Calculez suma patratelor distantelor:

$$\sum_{i=1}^{n} D_i^2 = 0 + 4 + 2 + 16 + 17 = 39$$

Obtin urmatorul tabel al distributiei de probabilitate cu valorile asociate:

i	$\frac{D_i^2}{\sum_{i\prime=1}^{n} D_{i\prime}^2}$
A	0
B	$\frac{4}{39}$
C	$\frac{2}{39}$
D	$\frac{16}{39}$
E	$\frac{17}{39}$

$\Rightarrow$ μ_2 = E va avea cea mai mare probabilitate de a fi ales de k-means++.

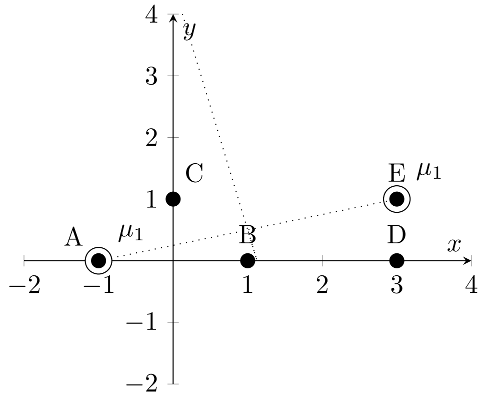
$\Rightarrow$ P($\mu_2 \in$ {B, C}) $= \frac{4}{39} + \frac{2}{39} = \frac{6}{39} = 0.1538$

Pentru a vedea daca am obtinut o imbunatatire voi simula in continuare k-means si voi compara numarul de iteratii pana la convergenta pentru varianta k-means++ si pentru k-means.

1. **K-means++**

- $\mu_2 = E$

  I Analiza geometrica:



$Cluster(\mu_1) = \{A, B, C\}$
$Cluster(\mu_2) = \{E, D\}$

$$x_{\mu_1} = \frac{x_A + x_B + x_C}{3} = \frac{-1+0+1}{3} = 0$$

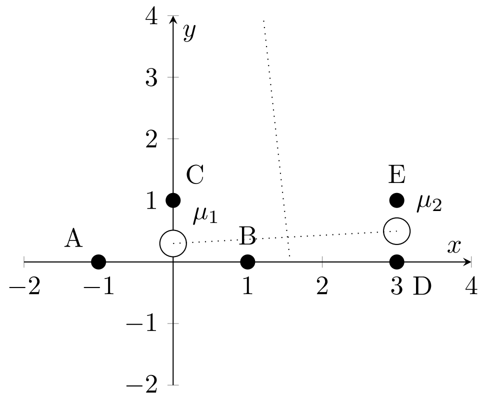
$$y_{\mu_1} = \frac{y_A + y_B + y_C}{3} = \frac{0+0+1}{3} = \frac{1}{3}$$

$$x_{\mu_2} = \frac{x_E + x_D}{2} = \frac{3+3}{2} = 3$$

$$y_{\mu_y} = \frac{y_E + y_D}{2} = \frac{1+0}{2} = \frac{1}{2}$$

$\Rightarrow \mu_1(0, \frac{1}{3}), \mu_2(3, \frac{1}{2})$

  II Analiza geometrica:

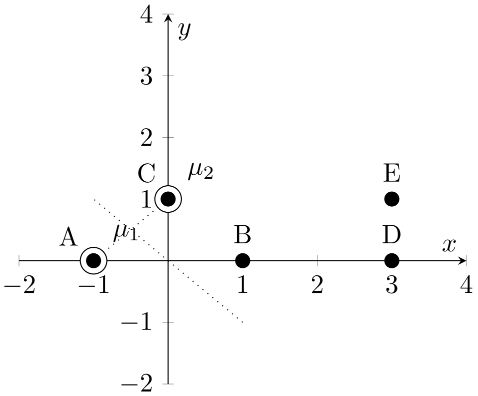


$Cluster(\mu_1) = \{A, B, C\}$
$Cluster(\mu_2) = \{E, D\}$
Clusterele nu s-au schimbat $\Rightarrow$ centroizii raman la fel $\Rightarrow$ algorimtul a convers in 2 iteratii.

2. **K-means**

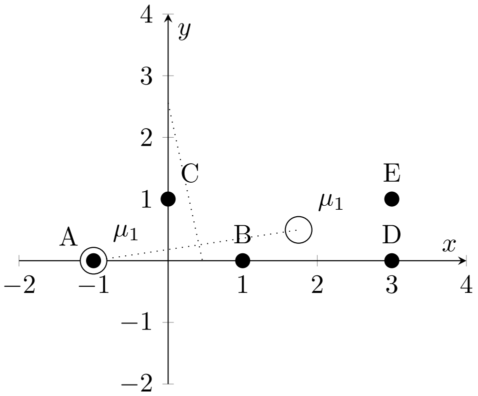
- $\mu_2 = C$

  I Analiza geometrica:



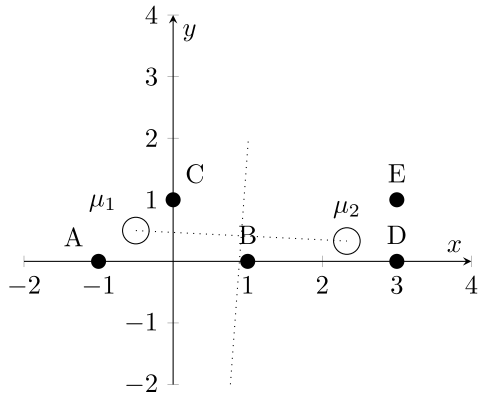
$Cluster(\mu_1) = \{A\}$
$Cluster(\mu_2) = \{B, C, E, D\}$
$\mu_1 = A \Rightarrow \mu_1(-1, 0)$
$x_{\mu_2} = \dfrac{0+1+3+3}{4} = \dfrac{7}{4}$
$y_{\mu_y} = \dfrac{1+0+0+1}{4} = \dfrac{1}{2}$
$\Rightarrow \mu_1(-1, 0), \mu_2(\frac{7}{4}, \frac{1}{2})$

II Analiza geometrica:



$Cluster(\mu_1) = \{A, C\}$
$Cluster(\mu_2) = \{B, E, D\}$
$x_{\mu_1} = \dfrac{-1+0}{2} = -\dfrac{1}{2}$
$y_{\mu_1} = \dfrac{0+1}{2} = \dfrac{1}{2}$
$x_{\mu_2} = \dfrac{1+3+3}{3} = \dfrac{7}{3}$
$y_{\mu_y} = \dfrac{0+0+1}{3} = \dfrac{1}{3}$
$\Rightarrow \mu_1(-\frac{1}{2}, \frac{1}{2}), \mu_2(\frac{7}{3}, \frac{1}{3})$

III Analiza geometrica:



$Cluster(\mu_1) = \{A, C\}$

$Cluster(\mu_2) = \{B, E, D\}$

Clusterele nu s-au modificat $\Rightarrow$ Algoritmul a convers in 3 iteratii

- $\mu_2 = B$

  C este la egala distanta de A si B $\Rightarrow$ aleg aleator, dar consistent.

  (a) C $\in$ Cluster(μ_1) $\Rightarrow$ convergenta in 2 iteratii (identic incepand cu itearatia 2 a cazului $\mu_2 = C$)

  (b) C $\in$ Cluster(μ_2) $\Rightarrow$ convergenta in 3 iteratii (dupa un rationament analog cu cel dezvoltat mai sus)

Concluzie:

Folosind k-means++ pentru initializarea centroizilor am favorizat convergenta algoritmului in 2 iteratii, comparativ cu posibilitatea de a se incheia dupa 3 iteratii in cazul μ_2 = C sau μ_2 = B (cu C $\in$ Cluster(μ_2)).

(1) $P_{k-means}(\mu_2 \in \{B, C\}) = 0.5$

(2) $P_{k-means++}(\mu_2 \in \{B, C\}) = 0.1535$

(1) + (2) $\Rightarrow$ Am favorizat alegerea ce duce la convergenta in 2 iteratii

Mai mult, pentru seturi de date mai mari, diferenta intre numarul de iteratii pana la convergenta cu si fara initializare k-means++ poate fi considerabila $\Rightarrow$ merita alocarea timpului necesar pentru calculele facute de k-means++, chiar daca sunt destul de complexe.

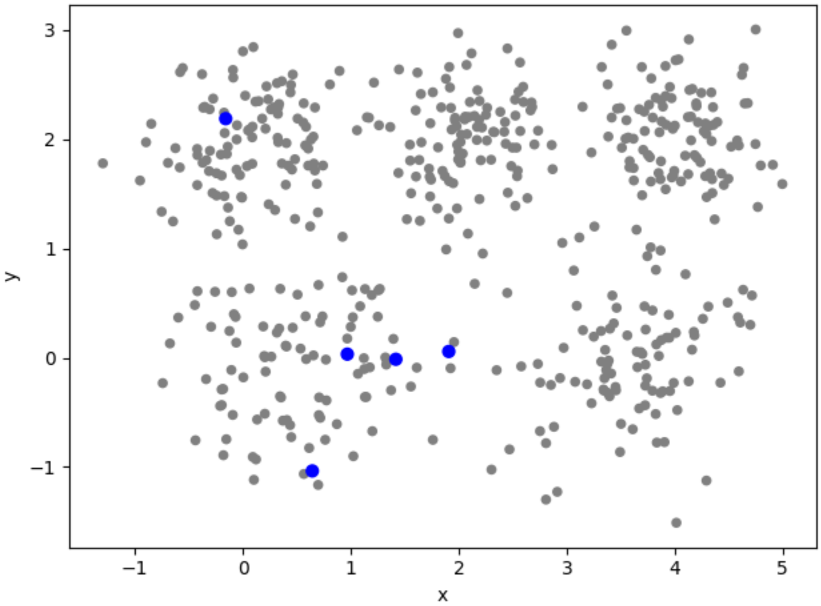
3 Studiu

Am rulat de 200 de ori algoritmul k-means (k = 5) pe un set de date cu 500 de intrari si am reprezentat grafic in plan punctele din datele de intrare (gri), centroizii alesi la initializarea algoritmului in cadrul primei iteratii a unei rulari a experimentului (albastru) si toate centrele cluster-urilor obtinute ca rezultat in urma fiecarei iteratii (rosu).

Prima data am initializat centroizii in maniera aleatorie uniforma pentru fiecare iteratie si am obtinut urmatoarele rezultate:

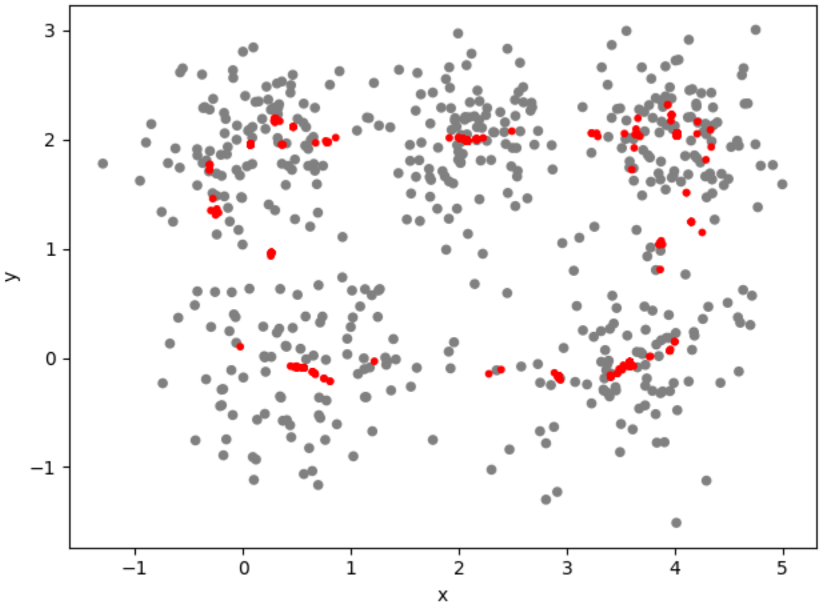
- Centroizii alesi la prima initializare a algoritmului:

Figure 1



- Cele 5 x 200 clusterele obtinute la finalul experimentului:

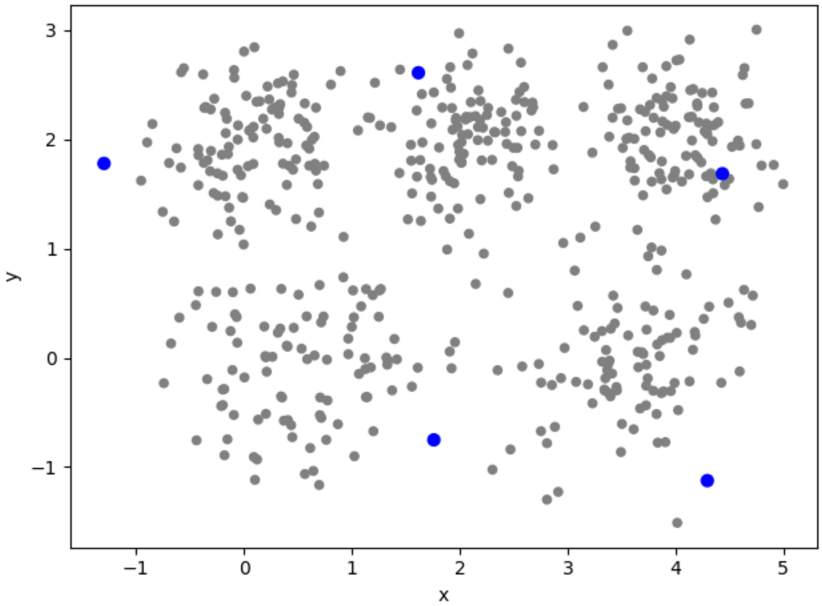
Figure 2



Pentru urmatoarea rulare a experimentului am initializat algoritmul folosind distributia de probabilitate furnizata de k-means++ si am obtinut urmatoarele rezultate:

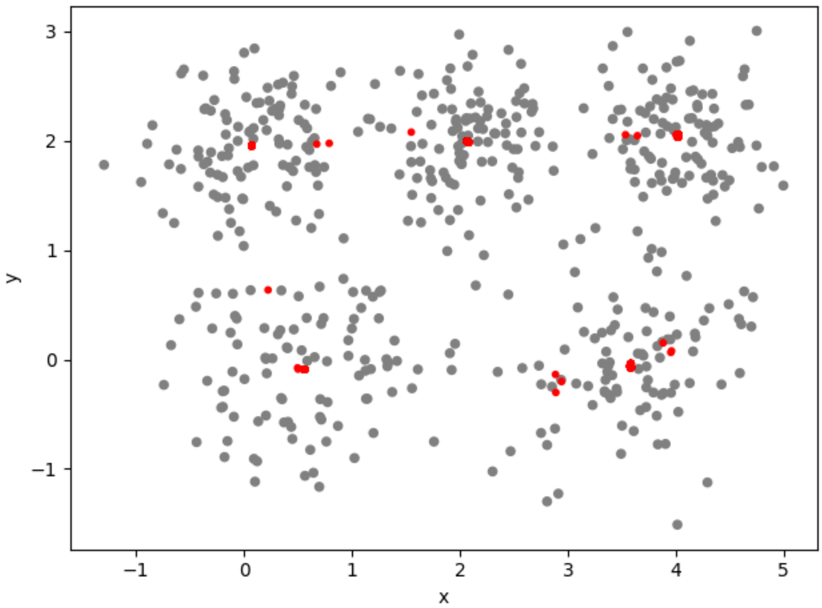
- Centroizii alesi la prima initializare a algoritmului:

Figure 3



- Cele 5 x 200 clusterele obtinute la finalul experimentului:

Figure 4



De asemenea, am afisat la finalul fiecarei rulari o statistica ce include: minimul obiectivelor obtinute (min_obj), media acestora (avg_objective), deviatia standard si numarul mediu de iteratii a algoritmului pana a convers:

Figure 5: Statistics

(a) K-means

```
min_obj:  222.36598776929088
avg_objective:  252.2581628589124
standard deviation:  68.8992271769586
avg_iterations_till_convergence:  9.58
run_time:  65.3121440410614
```

(b) K-means++

```
min_obj:  222.36598776929088
avg_objective:  225.47792603039224
standard deviation:  77.03268890965262
avg_iterations_till_convergence:  6.02
run_time:  70.8408350944519
```

Informatii extrase:

Prin compararea figurii 1 cu 3 se poate observa ca centroizii alesi cu k-means++ sunt mult mai bine distribuiti in setul de date, cate unul in fiecare din cele 5 grupari in care sunt dispuse punctele. Astfel, sunt din prima plasati in vecinatatea centrelor clusterelor ce vor fi obtinute de algoritm ceea ce se reflecta atat in numarul scazut de iteratii necesare pana la convergenta ($6.02 < 9.58$), cat si in scaderea mediei obiectivului: $225.47 < 252.25$.