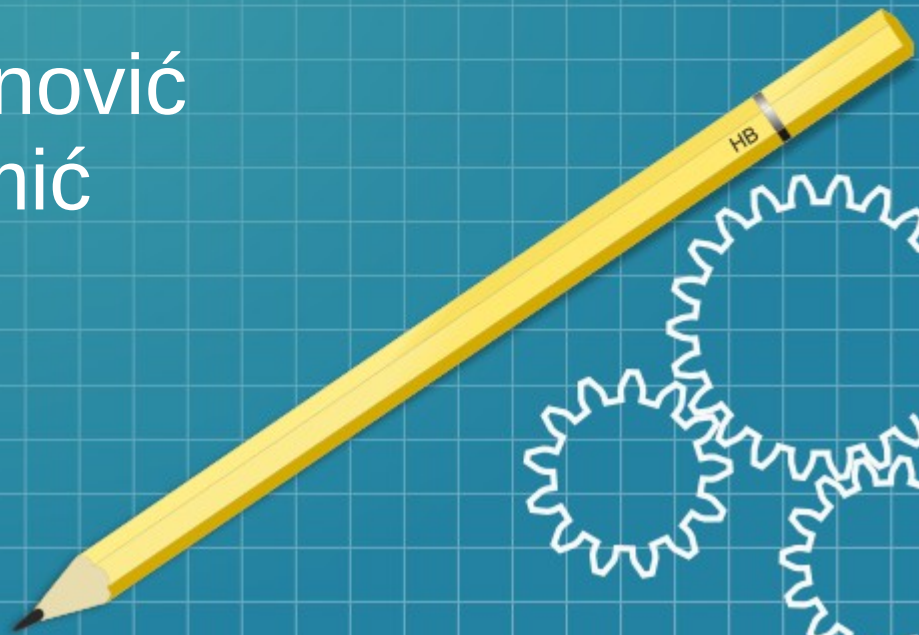
Kmeans metoda za klasterizaciju

Jelena Jovanović
Bojan Tomić

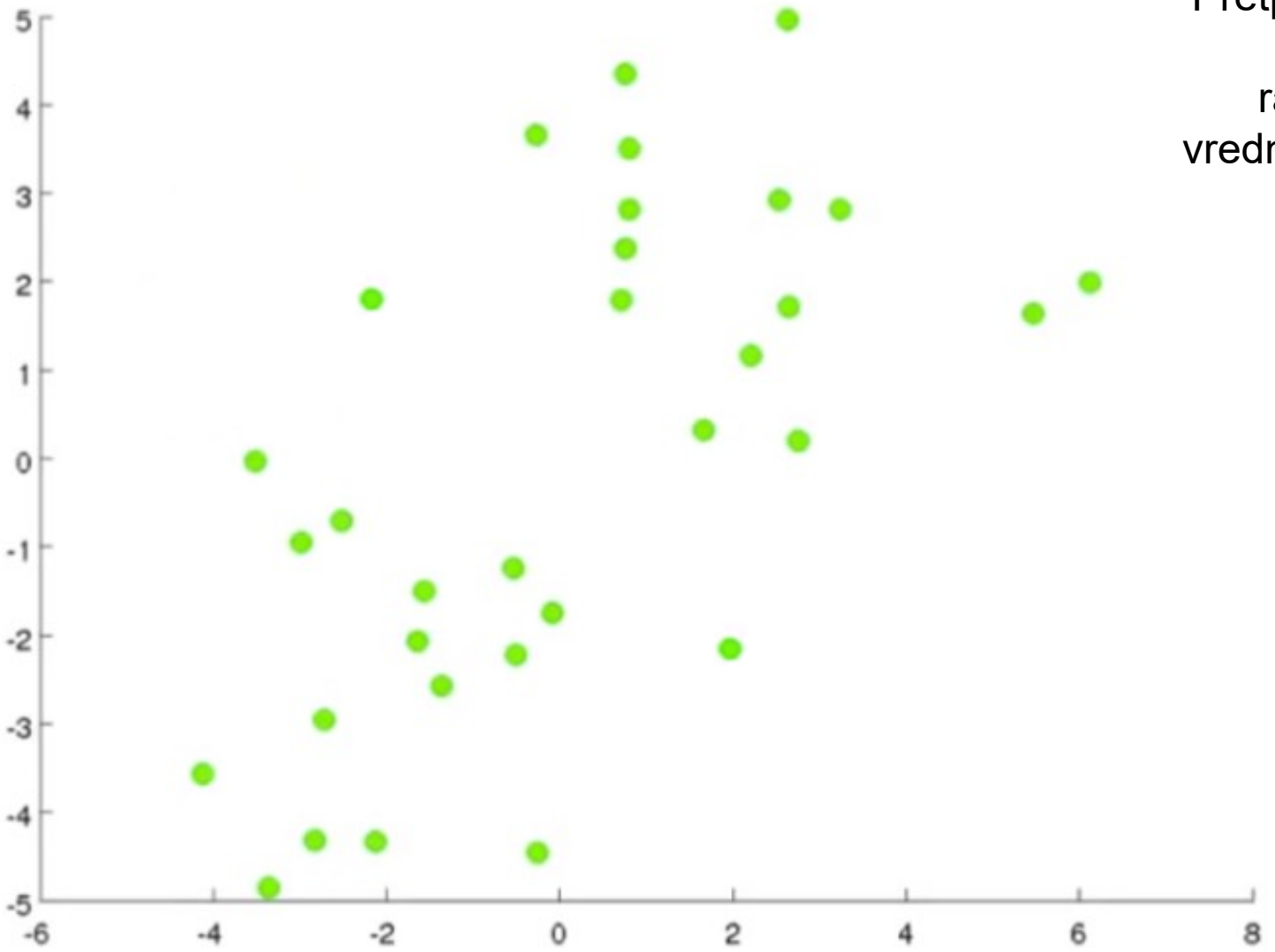


K-MEANS

- Jedan od najpoznatijih i najjednostavnijih algoritama klasterizacije
- Najlakše ga je razumeti na primeru, pa ćemo prvo razmotriti jedan primer

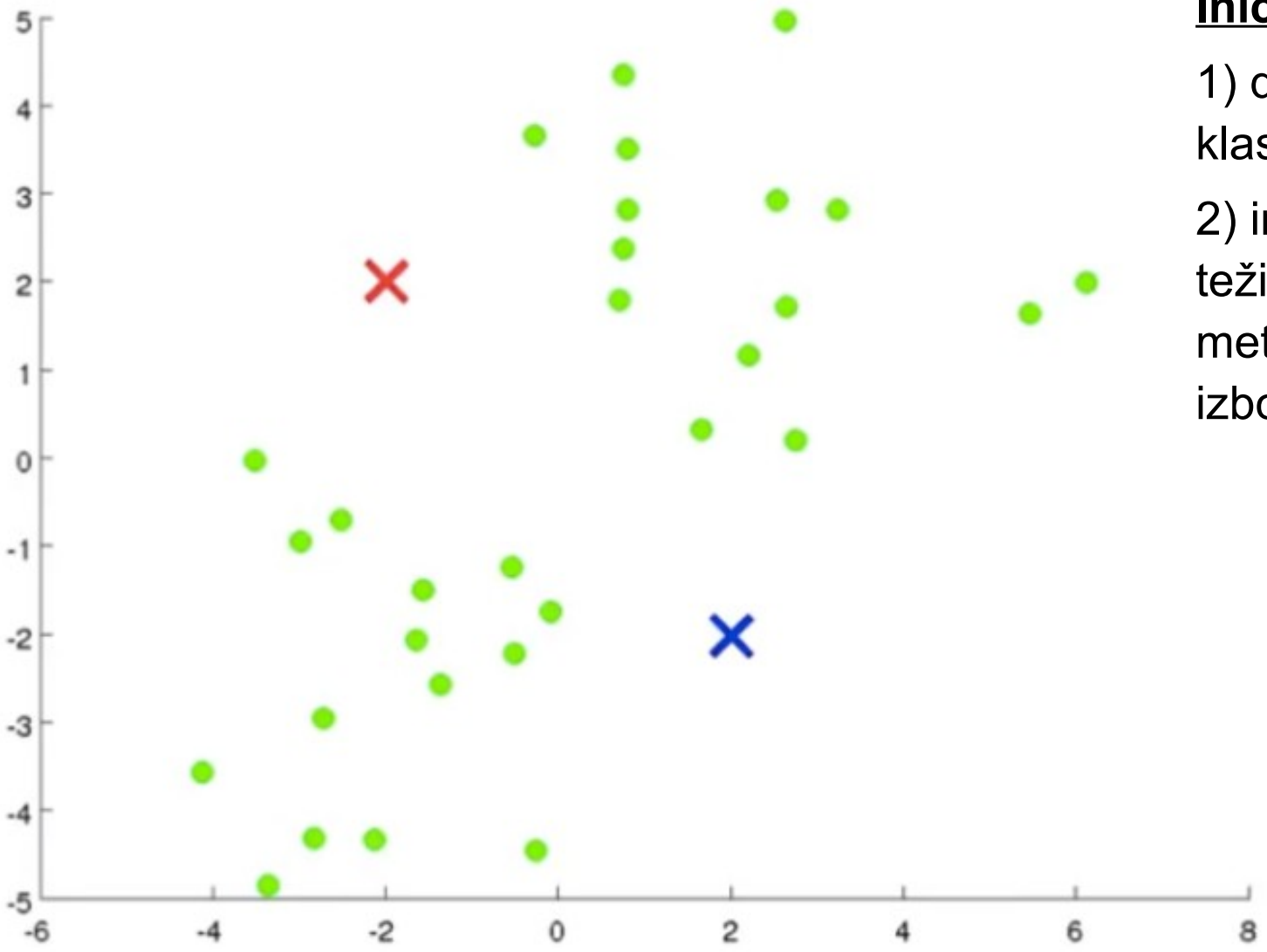
Primer je preuzet iz kursa: https://www.coursera.org/course/ml

K-MEANS ALGORITAM – PRIMER



Pretpostavimo da su ovo ulazni podaci kojima raspolažemo, opisani vrednostima dva atributa

K-MEANS: PRIMER

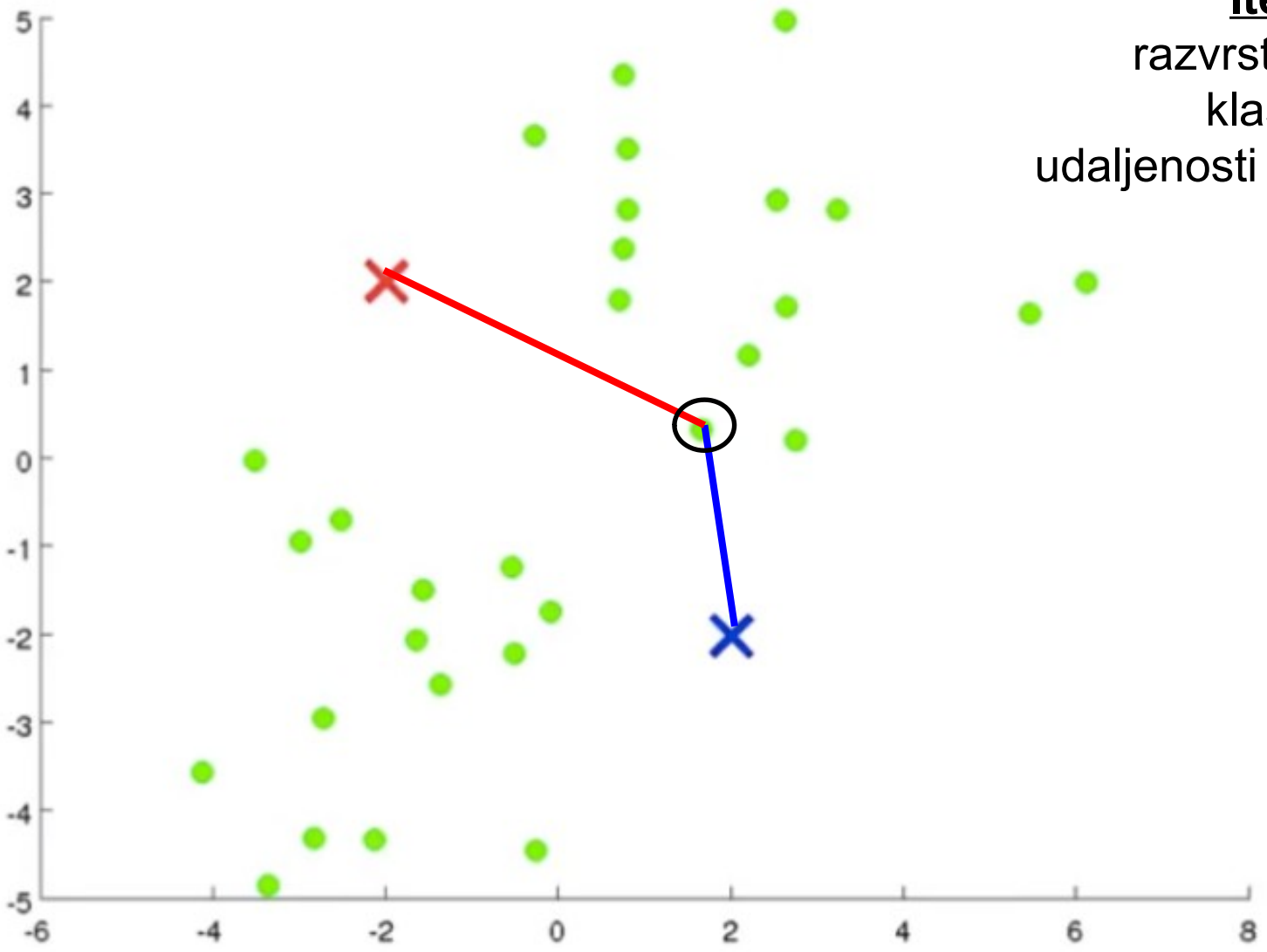


<u>Inicijalizacija</u>:

1) definisanje broja klastera, npr. K=2

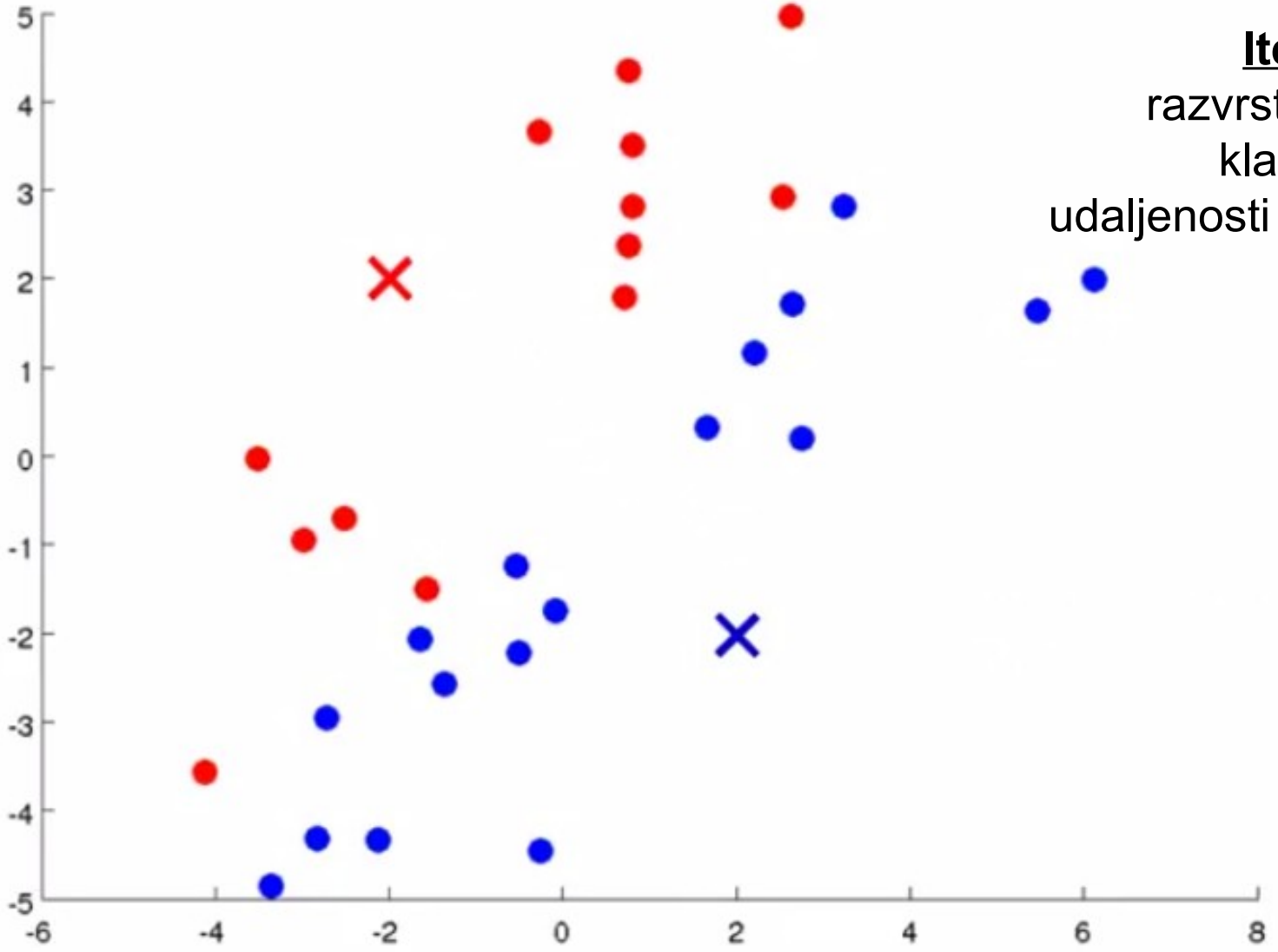
2) inicijalni izbor težišta klastera metodom slučajnog izbora

K-MEANS: PRIMER



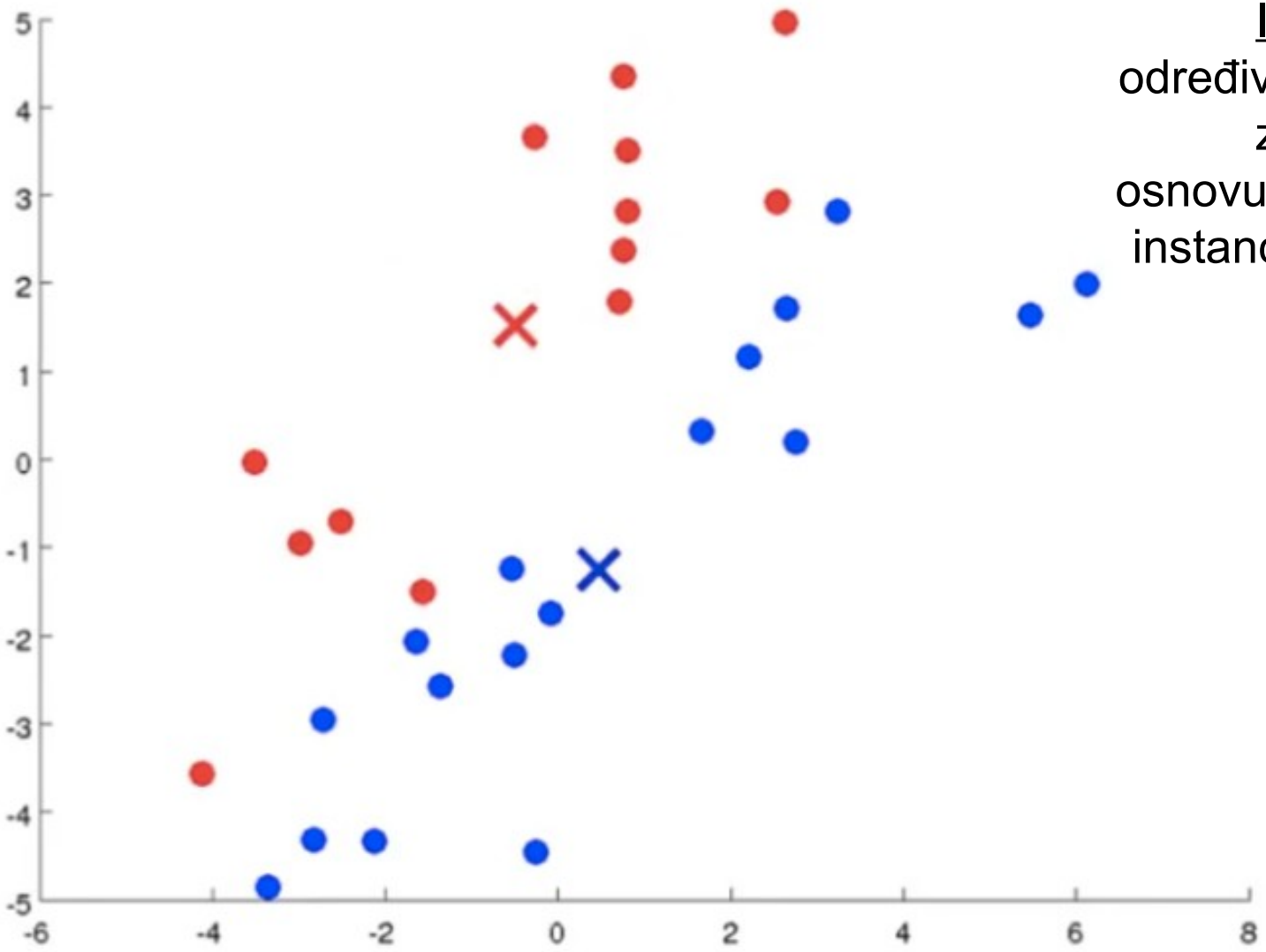
<u>Iteracija 1, korak 1</u>:
razvrstavanje instanci po klasterima na osnovu udaljenosti od težišta klastera

K-MEANS: PRIMER



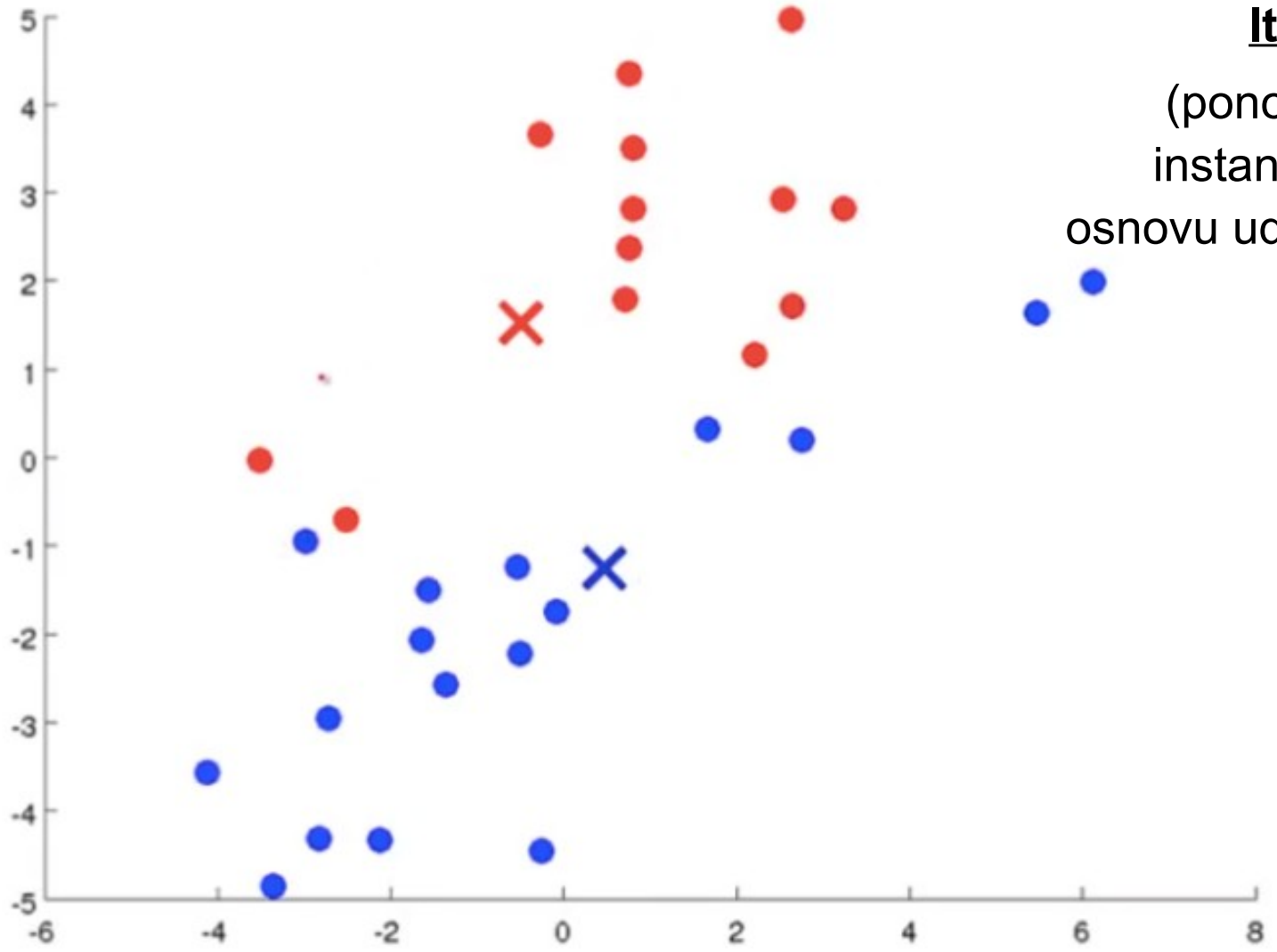
<u>Iteracija 1, korak 1</u>:
razvrstavanje instanci po klasterima na osnovu udaljenosti od težišta klastera

K-MEANS: PRIMER



Iteracija 1, korak 2:
određivanje novog težišta za svaki klaster, na osnovu proseka vrednosti instanci u datom klasteru

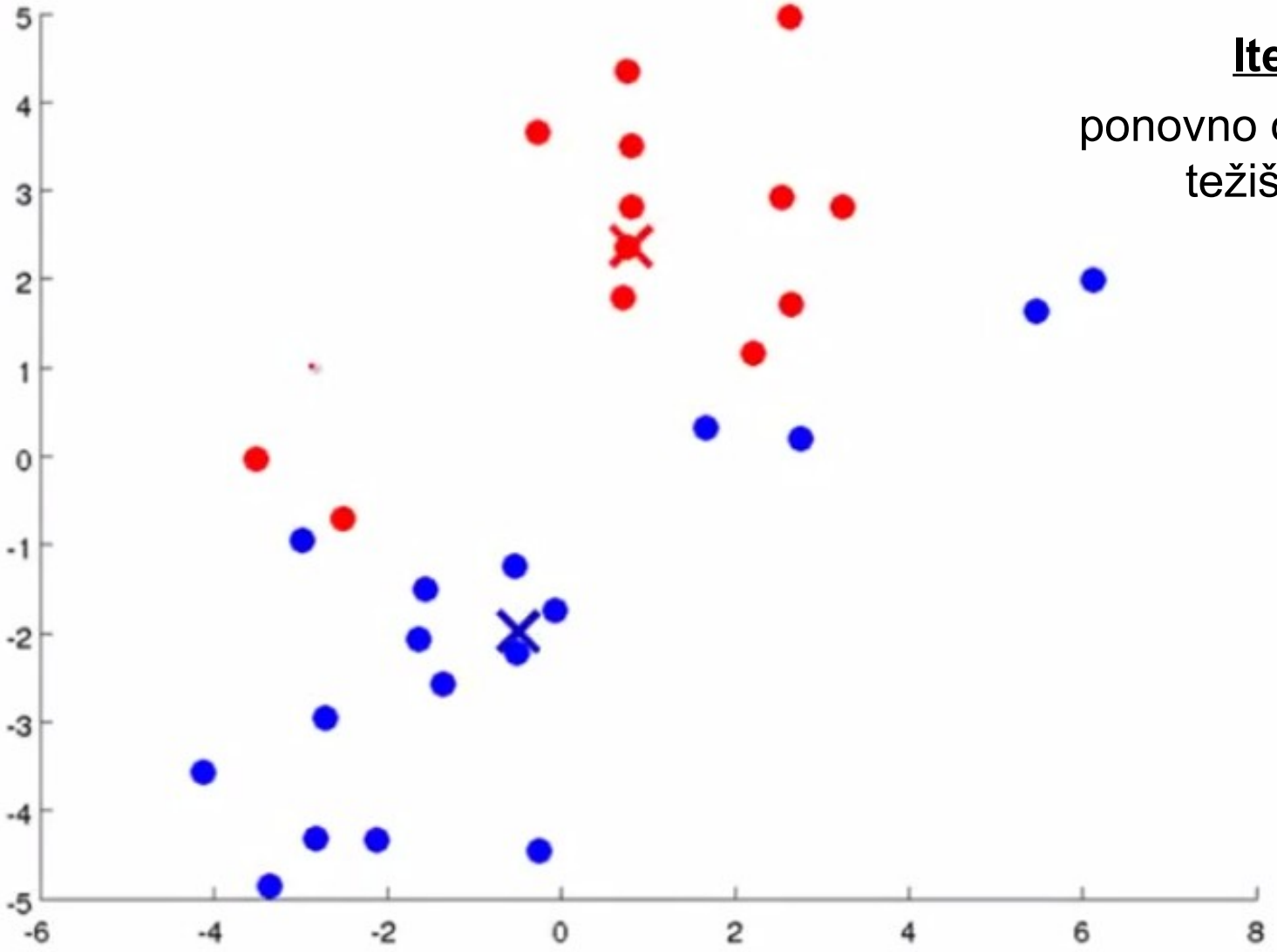
K-MEANS: PRIMER



<u>Iteracija 2, korak 1</u>:

(ponovno) razvrstavanje instanci po klasterima na osnovu udaljenosti od težišta klastera

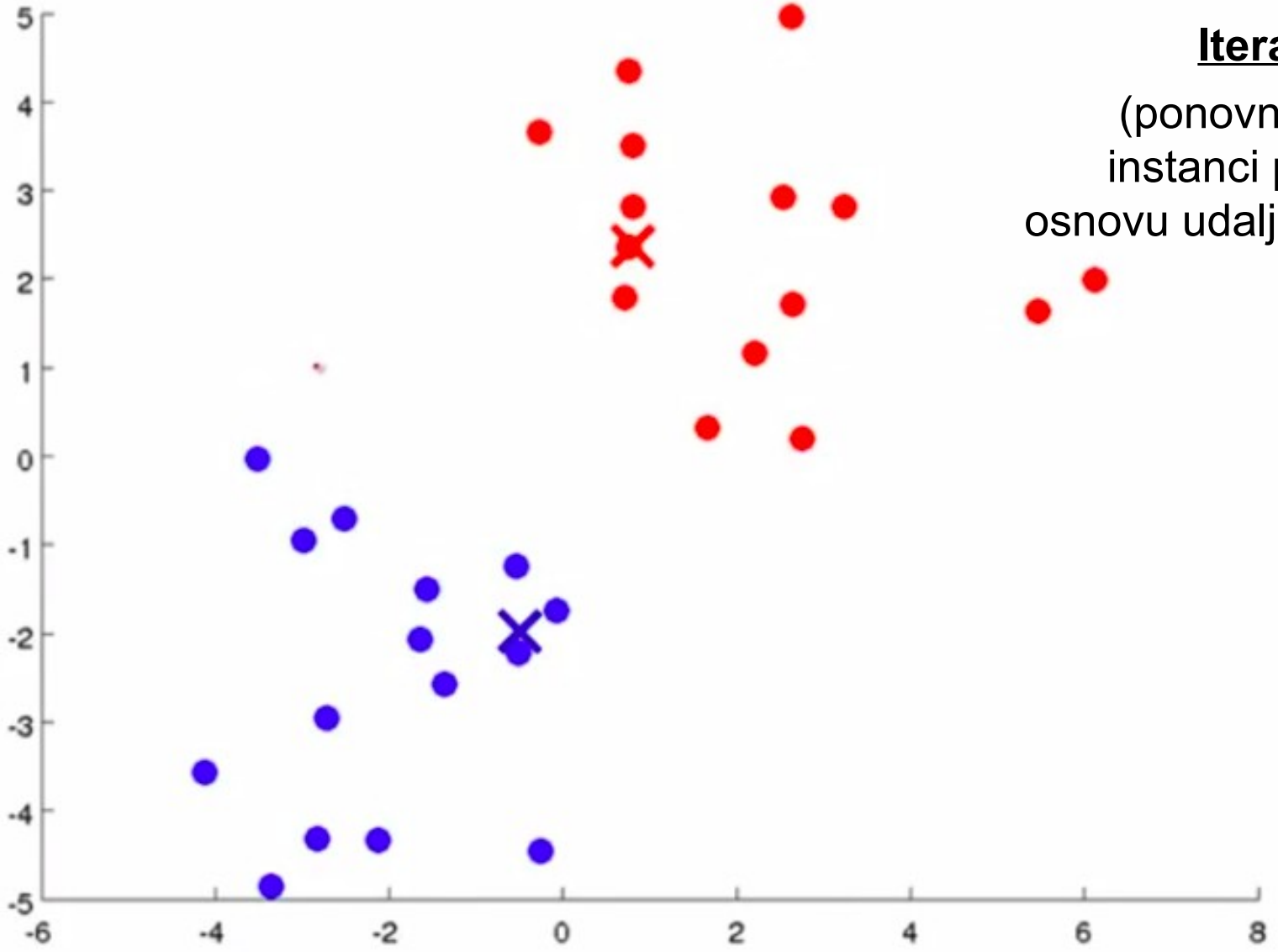
K-MEANS: PRIMER



<u>Iteracija 2, korak 2</u>:

ponovno određivanje novog težišta za svaki klaster

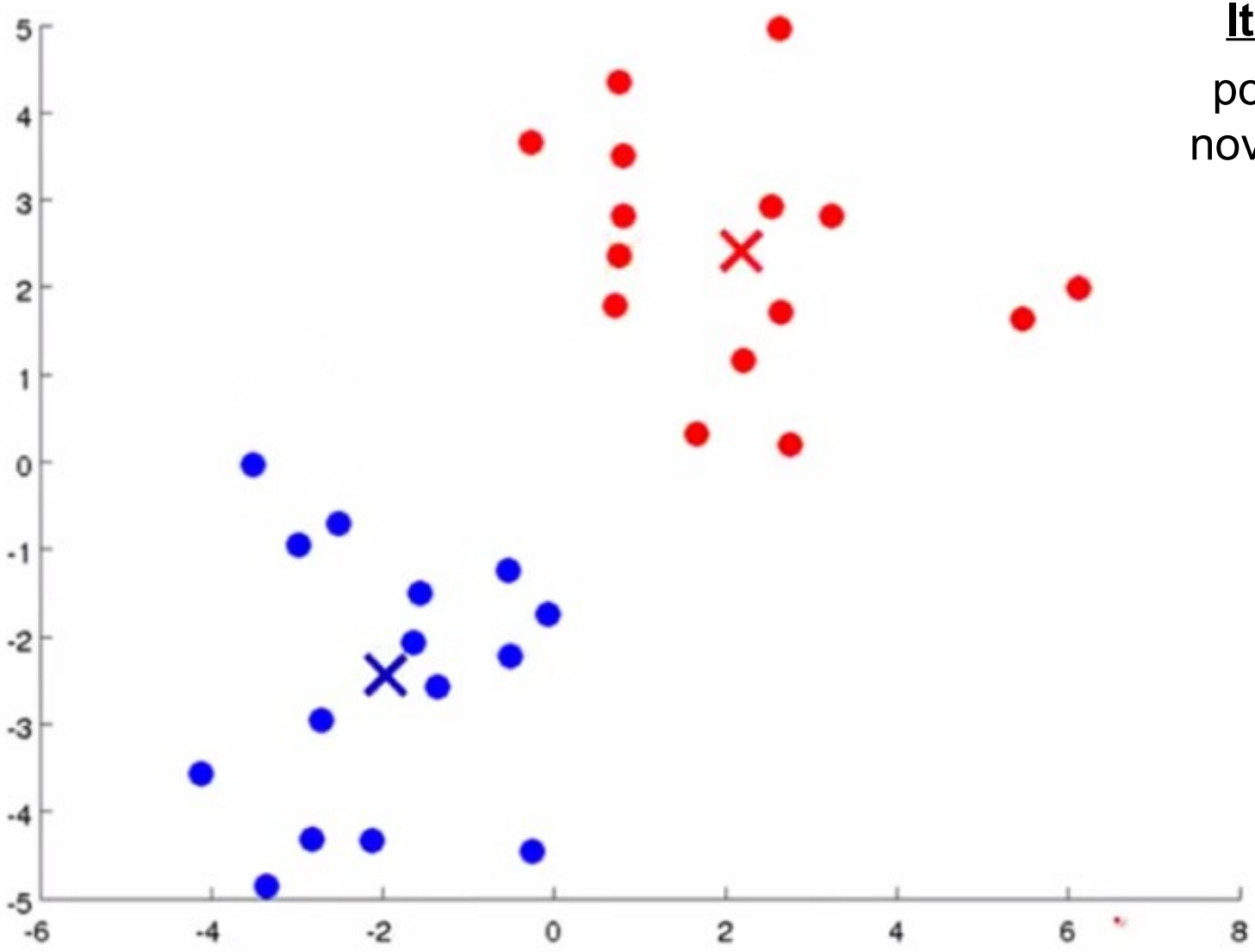
K-MEANS: PRIMER



<u>Iteracija 3, korak 1</u>:

(ponovno) razvrstavanje instanci po klasterima na osnovu udaljenosti od težišta klastera

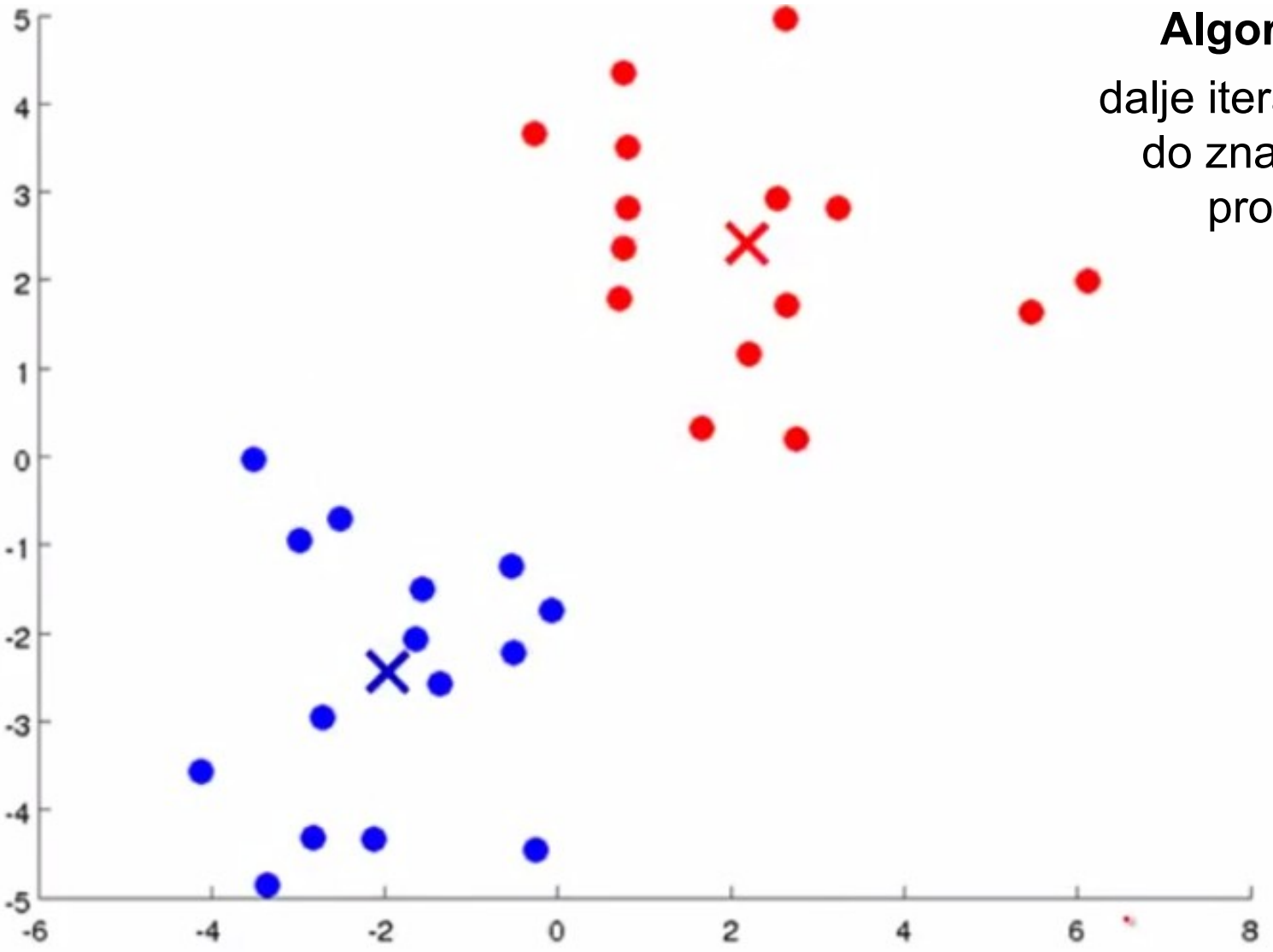
K-MEANS: PRIMER



<u>Iteracija 3, korak 2</u>:

ponovno određivanje novog težišta za svaki klaster

K-MEANS: PRIMER



Algoritam konvergira:

dalje iteracije neće dovesti do značajnijih promena i proces se zaustavlja

K-MEANS: ALGORITAM

Ulazni podaci:

- skup podataka sa *m* instanci; svaka instanca u skupu je vektor opisan sa *n* atributa (x_1, x_2, …, x_n)

Parametri algoritma:

- *K* - broj klastera
- *max* - max broj iteracija (opcioni parametar)

K-MEANS: ALGORITAM

Koraci:

1) Inicijalni izbor težišta klastera, slučajnim izborom
   - težišta se biraju iz skupa instanci, tj. *K* instanci se nasumično izabere i proglasi za težišta
2) Ponoviti
   1) *Grupisanje po klasterima*: za svaku instancu iz skupa podataka, *i* = 1,m, identifikovati najbliže težište i dodeliti instancu klasteru kome to težište pripada
   2) *Pomeranje težišta*: za svaki klaster izračunati novo težište uzimajući prosek instanci koje su dodeljene tom klasteru

   dok algoritam ne konvergira ili broj iteracija <= *max*

K-MEANS ALGORITAM: FUNKCIJA KOŠTANJA

Cilj K-means algoritma je *minimizacija funkcije koštanja* ***J*** (cost function):

$$J(c^{(1)},\ldots,c^{(m)},\mu_1,\ldots,\mu_K) = \frac{1}{m}\sum_{i=1}^{m}||x^{(i)} - \mu_{c^{(i)}}||^2$$

$x^{(i)}$ – *i*-ta instanca u skupu podataka, i=1,m

$\mu_{c(i)}$ – težište klastera u koji je instanca $x^{(i)}$ trenutno raspoređena

$c^{(i)}$ – indeks klastera u koji je instanca $x^{(i)}$ trenutno raspoređena

μ_j – težište klastera j, j=1,K

Ova funkcija se zove i funkcija distorzije (distortion function)

K-MEANS ALGORITAM: FUNKCIJA KOŠTANJA

$$\min_{\substack{c^{(1)},\ldots,c^{(m)},\\ \mu_1,\ldots,\mu_K}} J(c^{(1)},\ldots,c^{(m)},\mu_1,\ldots,\mu_K)$$

Minimizacija funkcije koštanja ***J*** kroz K-means algoritam:

- faza *Grupisanja po klasterima* minimizuje ***J*** po parametrima $c^{(1)},\ldots,c^{(m)}$, držeći $\mu_1,\ldots,\mu_K$ fiksnim
- faza *Pomeranja težišta* minimizuje ***J*** po parametrima $\mu_1,\ldots,\mu_K$, držeći $c^{(1)},\ldots,c^{(m)}$ fiksnim

K-MEANS: PROBLEMI

- Problemi pri određivanju parametara algoritma:
  - Kako odrediti ukupan broj klastera (težišta) – K
    - Uz znanje o pojavi/fenomenu koji se istražuje i pretpostavke o broju klastera
    - Bez znanja o pojavi/fenomenu koji se istražuje i pretpostavki o broju klastera
  - Inicijalni izbor (koordinata) težišta
    - Višestruka nasumična inicijalizacija težišta
    - K-Means++ algoritam

K-MEANS: KAKO ODREDITI K?

U slučaju da posedujemo znanje o fenomenu/pojavi koju podaci opisuju:

- Pretpostaviti broj klastera (K) na osnovu domenskog znanja
- Testirati model sa K-1, K, K+1 klastera i uporediti grešku*

*Na primer, korišćenjem *within cluster sum of squared errors* metrike

K-MEANS: KAKO ODREDITI K?

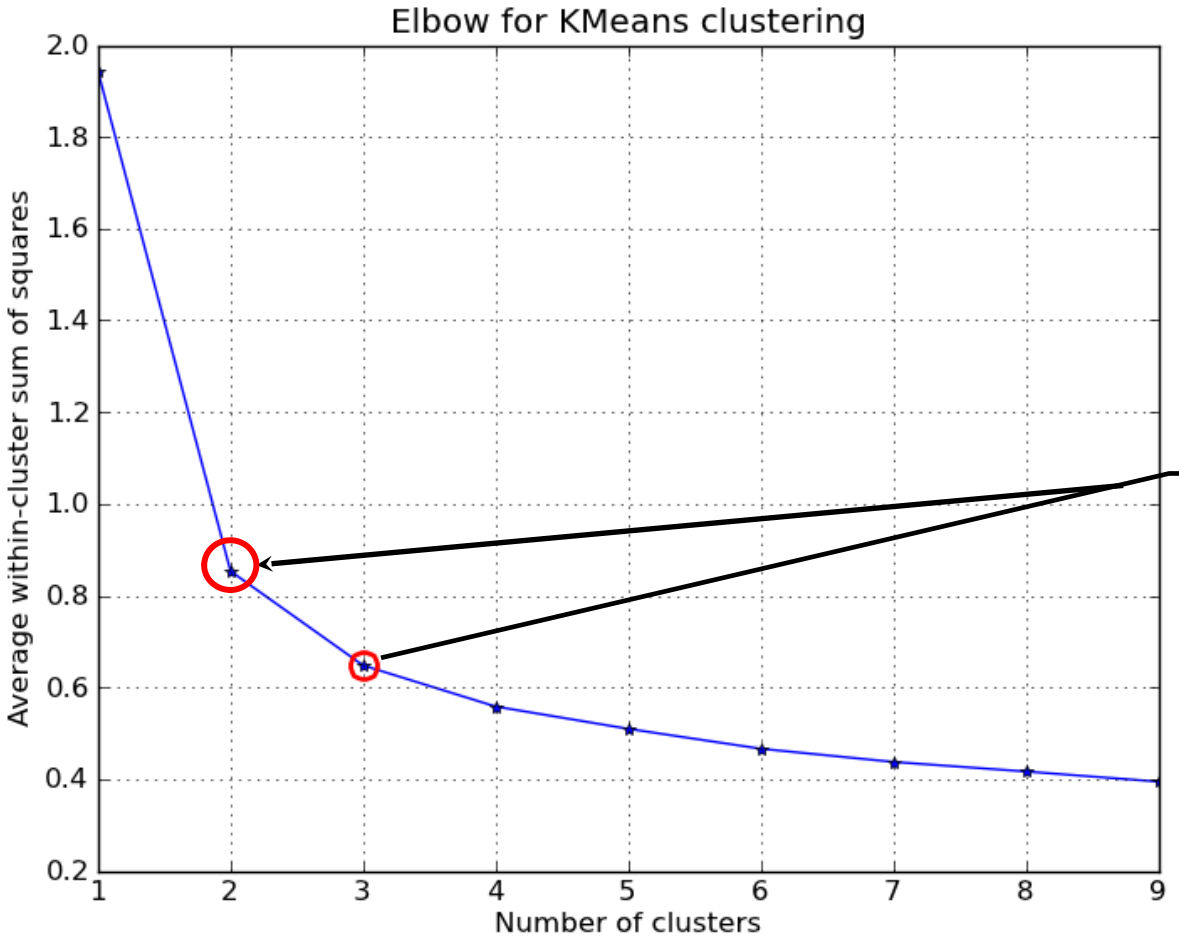
Ukoliko ne posedujemo znanje o fenomenu/pojavi

- Krenuti od malog broja klastera i u više iteracija testirati model uvek sa jednim klasterom više
- U svakoj od iteracija, uporediti grešku* tekućeg i prethodnog modela i kad smanjenje greške postaje zanemarljivo, prekinuti postupak
- Ova metoda je poznata kao lakat (eng. „elbow“) metoda

*Na primer, korišćenjem *within cluster sum of squared errors* metrike

K-MEANS: KAKO ODREDITI K?

ELBOW (LAKAT) METODA



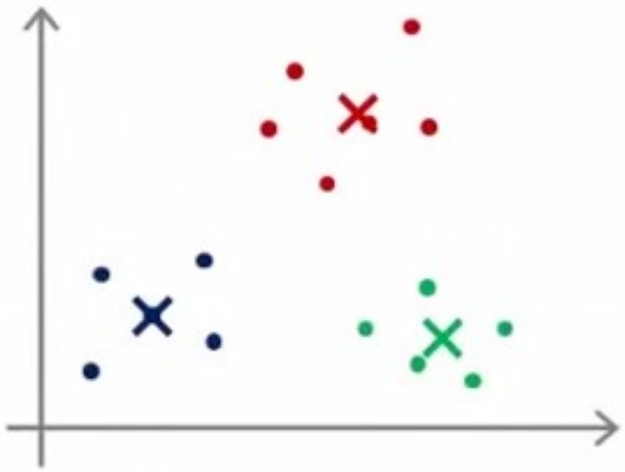
Kandidati za optimalan broj klastera (K)

Izvor slike: http://i.stack.imgur.com/BzwBY.png

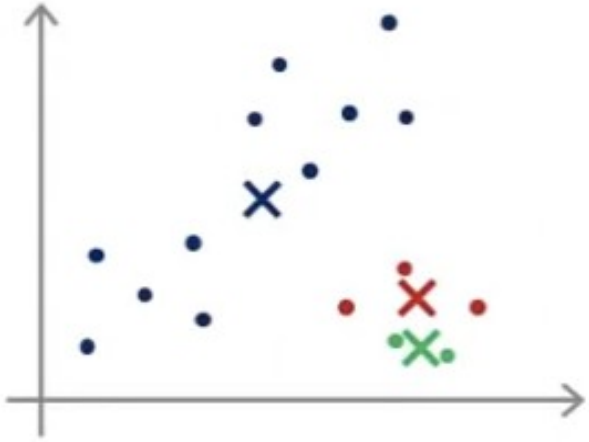
K-MEANS:
PROBLEM INICIJALNOG IZBORA TEŽIŠTA

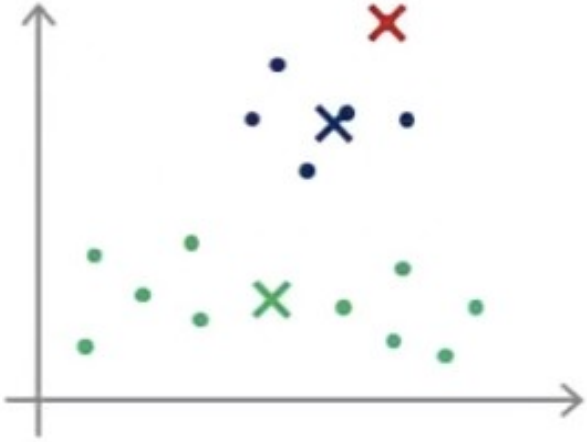
Kada znamo K, sledeći problem je inicijalni izbor (koordinata) tih K težišta:

- K-means algoritam može konvergirati brže ili sporije
- Takodje, može “upasti” u lokalni minimum funkcije koštanja i dati loše rešenje



Idealna inicijalizacija





Inicijalizacija koja vodi u lokalne minimume

K-MEANS: VIŠESTRUKA NASUMIČNA INICIJALIZACIJA

Omogućuje da se izbegnu situacije koje K-means dovode u lokalni minimum

Sastoji se u sledećem:

```
for i = 1 to n { //n obično uzima vrednosti 50 - 1000
      Nasumično odabrati inicijalni skup težišta;
      Izvršiti K-Means algoritam;
      Izračunati funkciju koštanja (cost function)
}
Izabrati instancu algoritma koja daje najmanju vrednost za f.
koštanja
```

Ovaj pristup daje dobre rezultate ukoliko je broj klastera relativno mali (2 - 10); za veći broj klastera ne bi ga trebalo koristiti

K-MEANS++:
BOLJI PRISTUP INICIJALIZACIJE ALGORITMA

Osnovna ideja: inicijalno odabrati *k* težišta koja su što dalje jedna od drugog

Postupak se sastoji u sledećem:

1. Nasumično izabrati prvo težište među instancama
2. Za svaku instancu izračunati njenu udaljenost od prethodno izabranih težišta
3. Nasumično izabrati sledeće težište među instancama koje su najviše udaljene od njima najbližih, prethodno izabranih težišta
4. Ponoviti korake 2 i 3 dok se ne uzorkuje *k* težišta

Kmeans algoritam

- Prednosti
  - Umereno procesorski i memorijski intenzivna (razdaljine instanci od težišta klastera)
  - Jednostavna za implementaciju, razumevanje i objašnjavanje
- Mane
  - Potrebno je pretpostaviti broj klastera (K) ili izračunati optimalan
  - Nekada ne konvergira
  - Osetljiva na inicijalni izbor težišta klastera
  - Osetljiva na izbor metrike udaljenosti
  - Osetljiva na šum i outlier-e