

# KLASTERIZACIJA

**JELENA JOVANOVIĆ**

Email: [jeljov@gmail.com](mailto:jeljov@gmail.com)

Web: <http://jelenajovanovic.net>

# PREGLED PREDAVANJA

- Šta je klasterizacija?
- Koje su oblasti/primeri primene?
- Klasterizacija primenom K-Means algoritma
- Klasterizacija primenom EM algoritma

# ŠTA JE KLASTERIZACIJA?

Klasterizacija je jedan od oblika nenadgledanog m. učenja

- ono što je raspoloživo od podataka su podaci o instancama koje je potrebno na neki način grupisati;
- ne posedujemo podatke o poželjnoj / ispravnoj grupi (klasi) za ulazne instance

# ŠTA JE KLASTERIZACIJA?

Klasterizacija je zadatak grupisanja instanci, tako da za svaku instancu važi da je *sličnija* instancama iz svoje grupe (klastera), nego instancama iz drugih grupa (klastera)

# PROCENA SLIČNOSTI INSTANCI

Sličnost instanci se procenjuje primenom neke od mera za računanje:

- sličnosti (npr. Kosinusna sličnost ili koeficijent korelacije)

ili

- udaljenosti dve instance (npr. Euklidska ili Manhattan udaljenost)

# PROCENA SLIČNOSTI INSTANCI

Euklidska sličnost:

$$d(i, j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2)}$$

Manhattan sličnost:

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

$p$  – broj atributa kojima su instance opisane

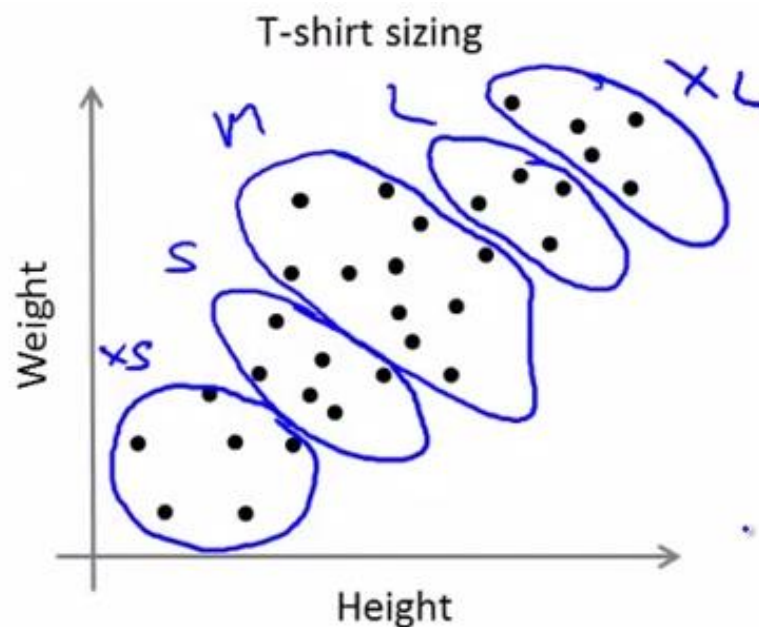
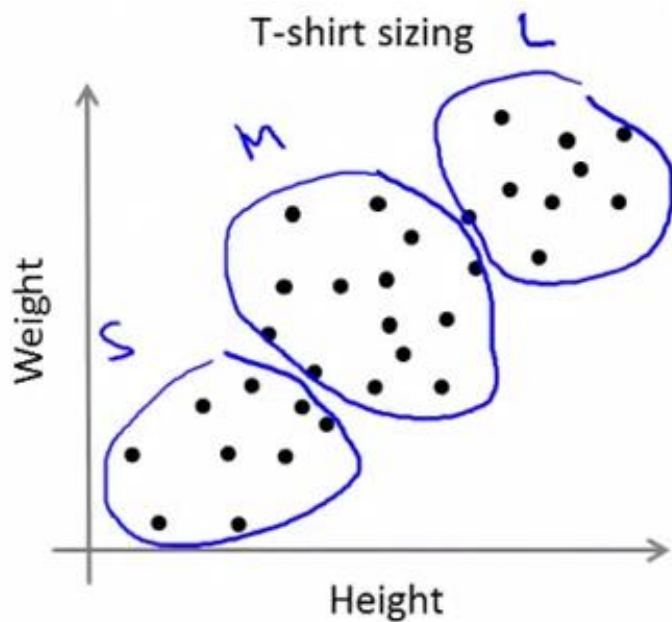
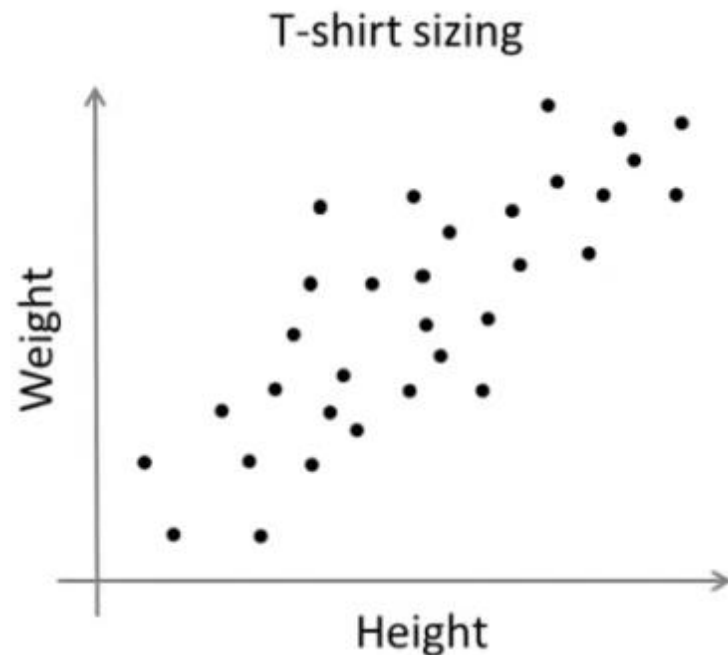


# PROCENA REZULTATA KLASTERIZACIJE

Za razliku od klasifikacije, ovde nemamo “tačno” rešenje

- ocena uspešnosti algoritma je dosta teža nego kod klasifikacije
- pogodnost rešenja zavisi od domena i slučaja primene
  - jedno isto rešenje može biti različito ocenjeno u različitim slučajevima primene
- zahteva angažovanje domenskih eksperata koji će evaluirati rešenje

Primer različnih dobrih rešenja za isti polazni skup podataka





# OBLASTI PRIMENE

- Segmentacija tržišta
- Uočavanje grupa u društvenim mrežama
- Identifikacija korisnika koje karakterišu slični oblici interakcije sa sadržajima nekog Web sajta/aplikacije
- Grupisanje objekata (npr., slika/dokumenata) prema zajedničkim karakteristikama
- ...

# K-MEANS ALGORITAM

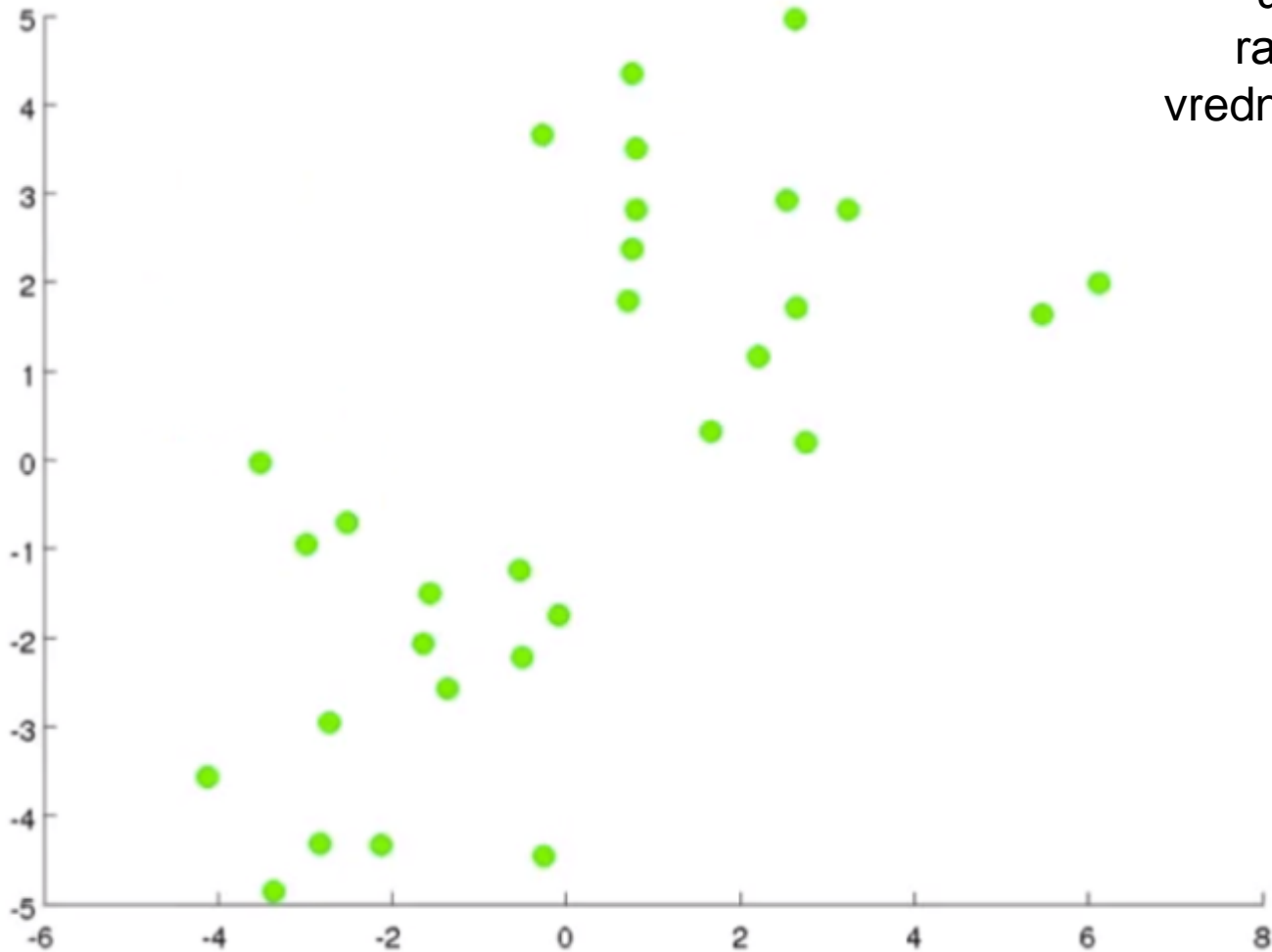
# K-MEANS

Jedan od najpoznatijih i najjednostavnijih algoritama klasterizacije

Najlakše ga je razumeti na primeru, pa ćemo prvo razmotriti jedan primer

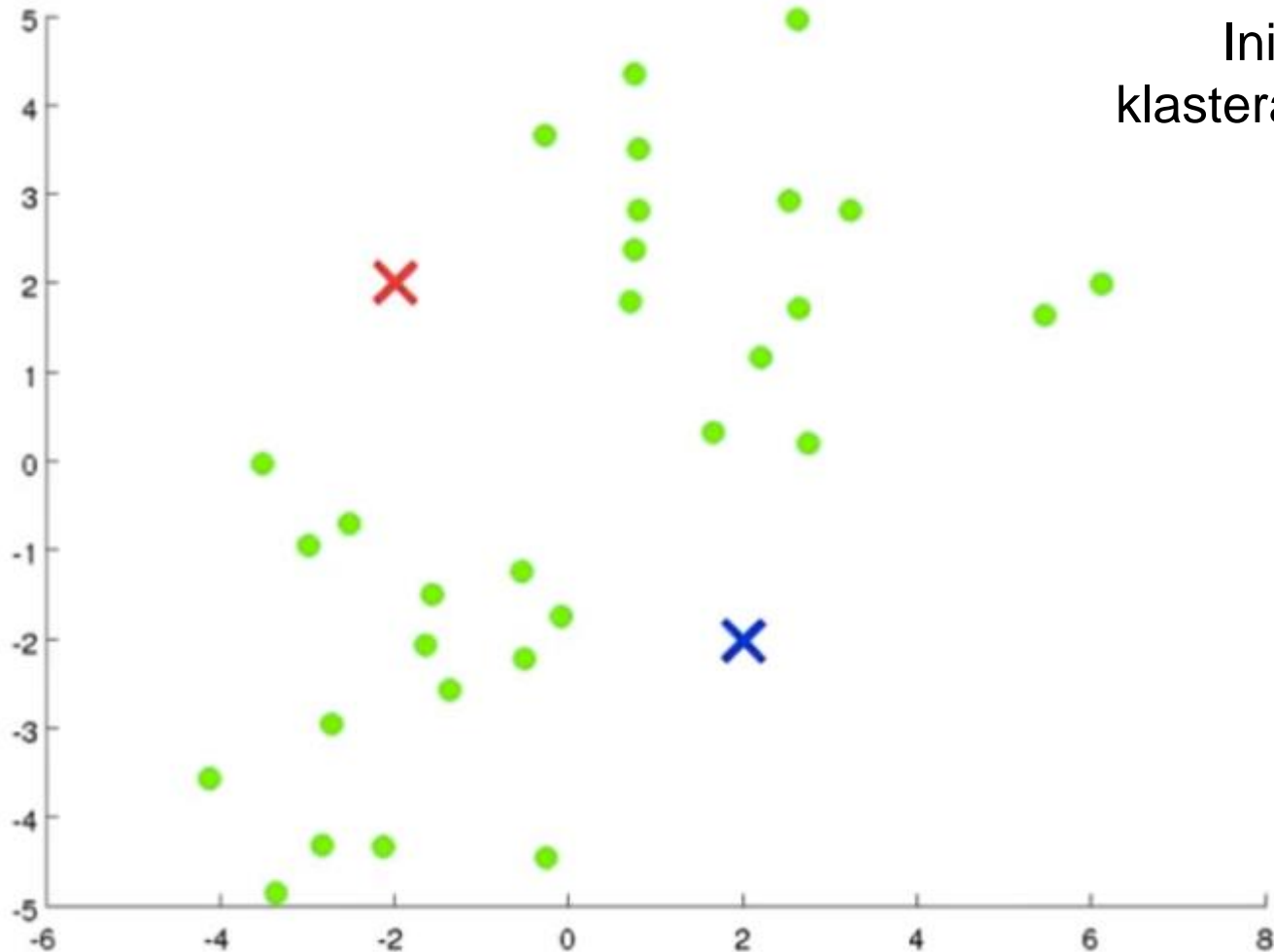
# K-MEANS ALGORITAM – PRIMER

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

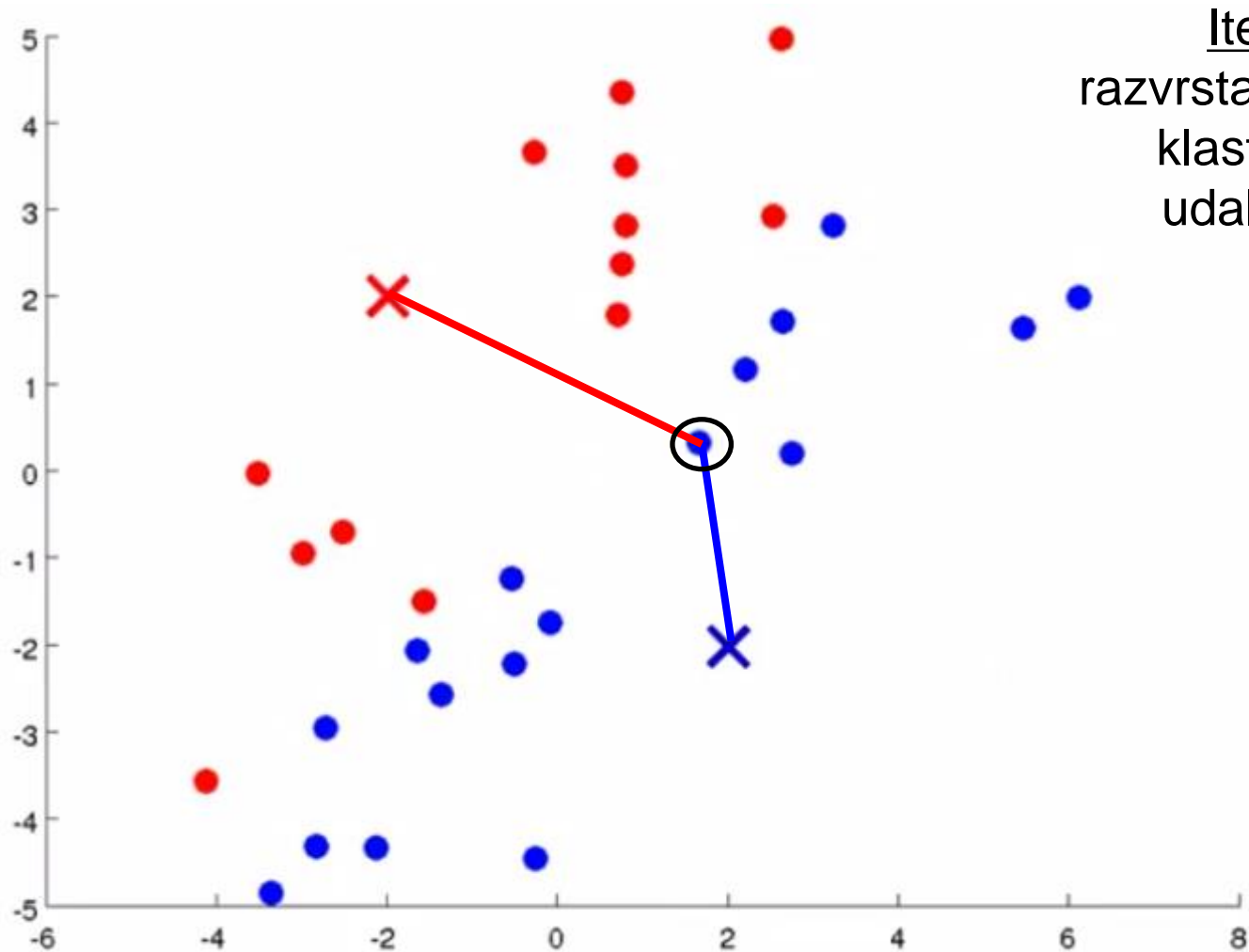


# K-MEANS: PRIMER

Inicijalizacija:  
Inicijalni izbor težišta  
klastera ( $K = 2$ ) metodom  
slučajnog izbora

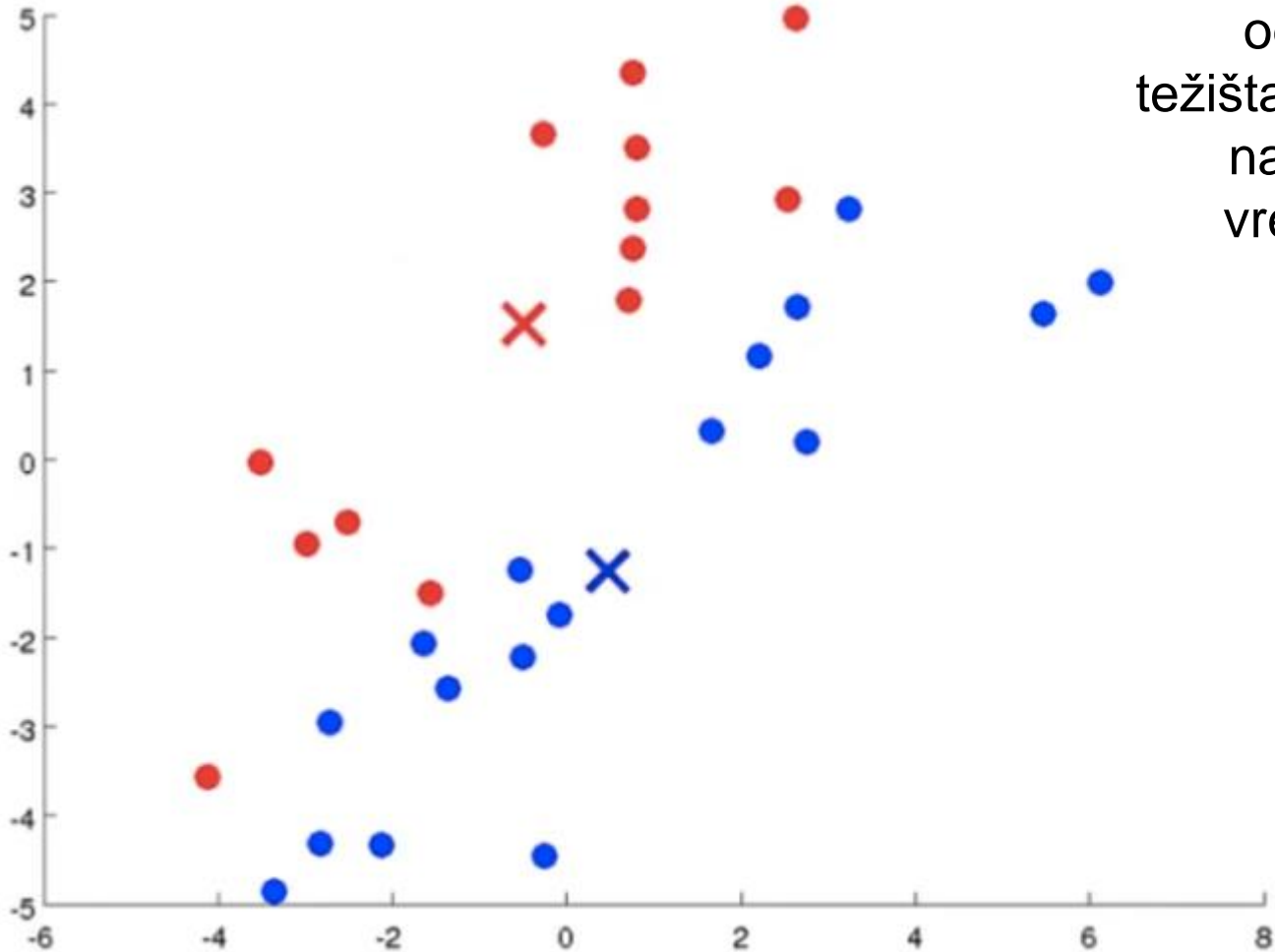


# K-MEANS: PRIMER



Iteracija 1, korak 1:  
razvrstavanje instanci po  
klasterima na osnovu  
udaljenosti od težišta  
klastera

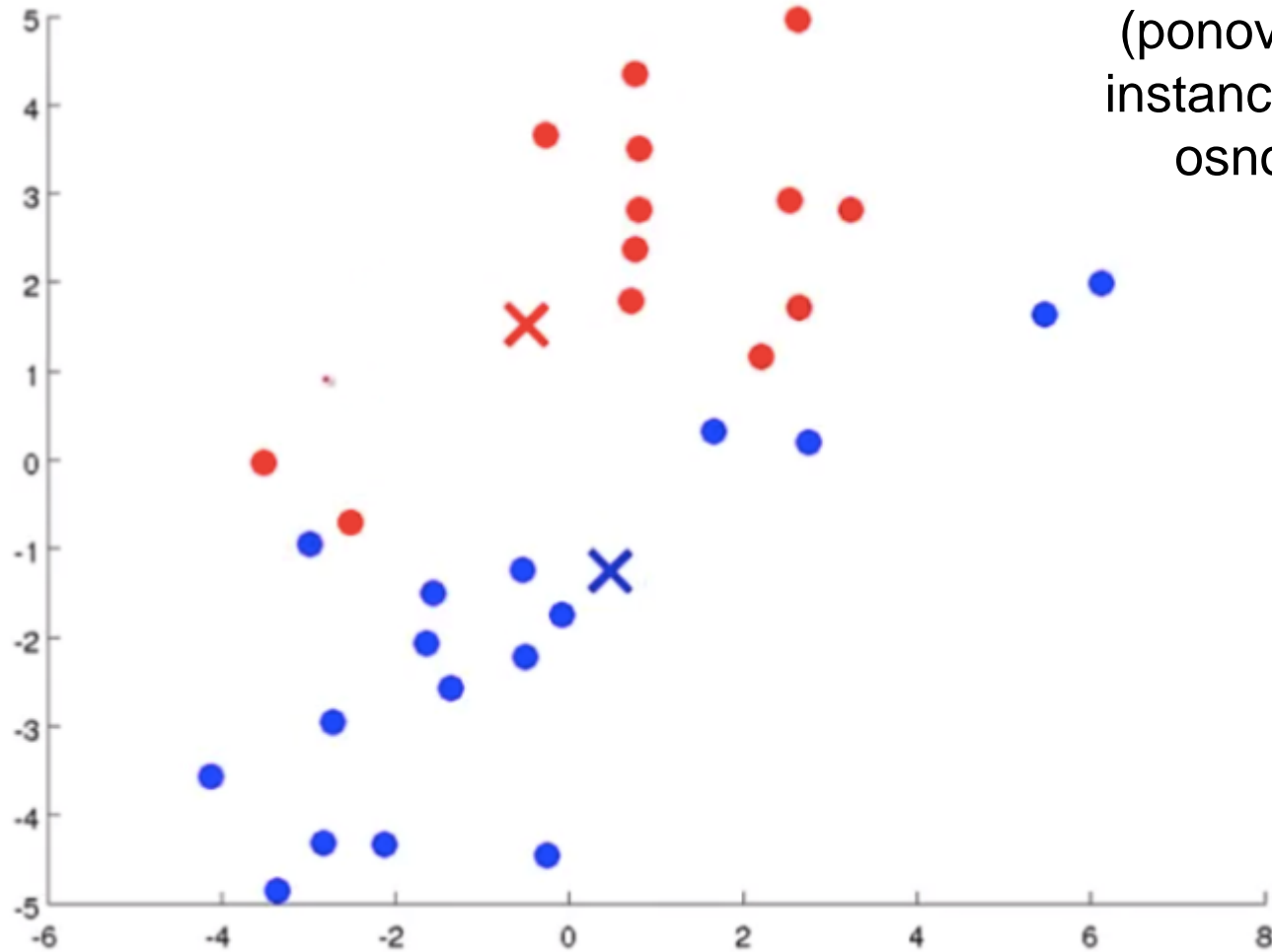
# K-MEANS: PRIMER



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

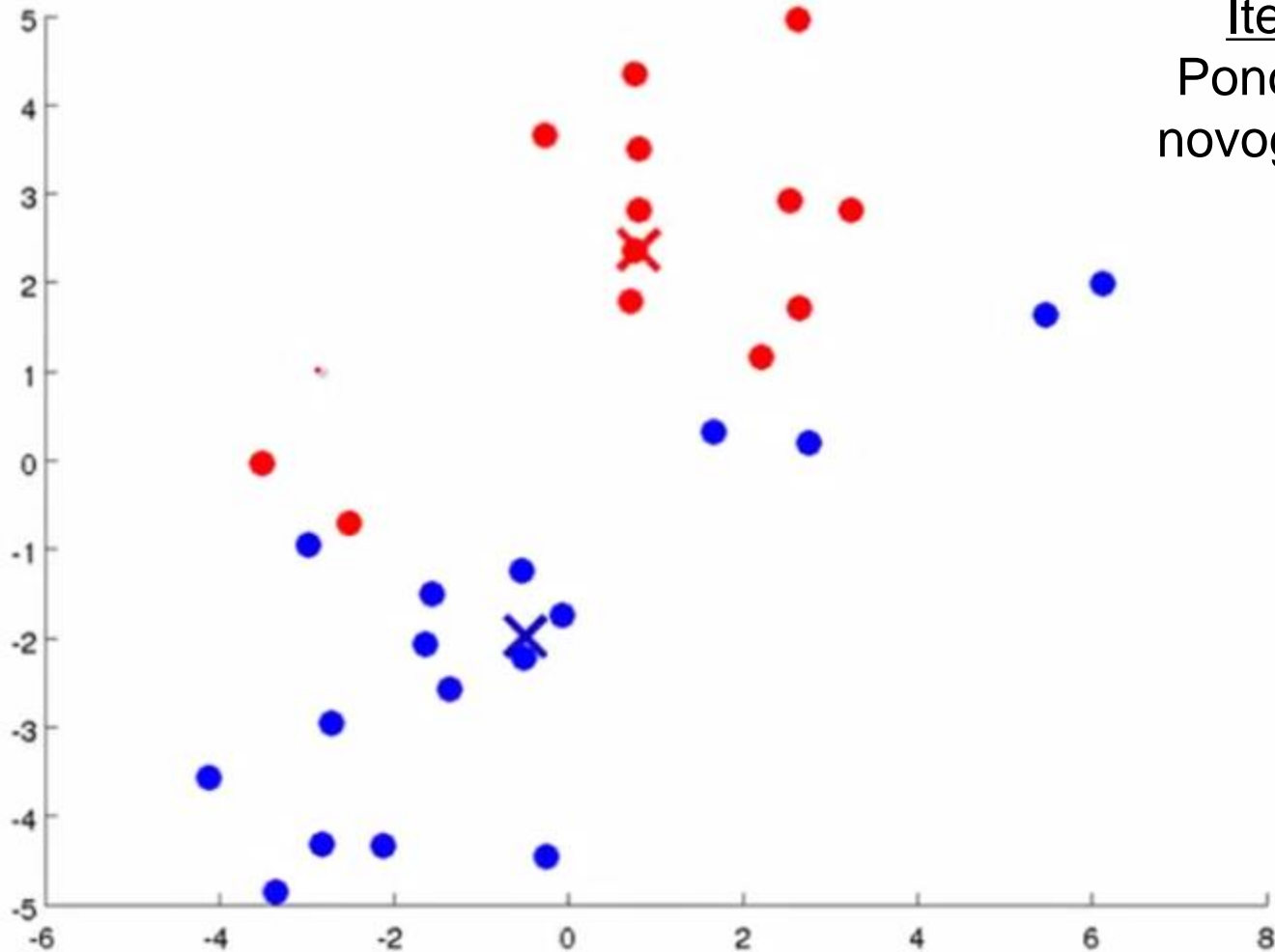
# K-MEANS: PRIMER

Iteracija 2, korak 1:  
(ponovno) razvrstavanje  
instanci po klasterima na  
osnovu udaljenosti od  
težišta klastera



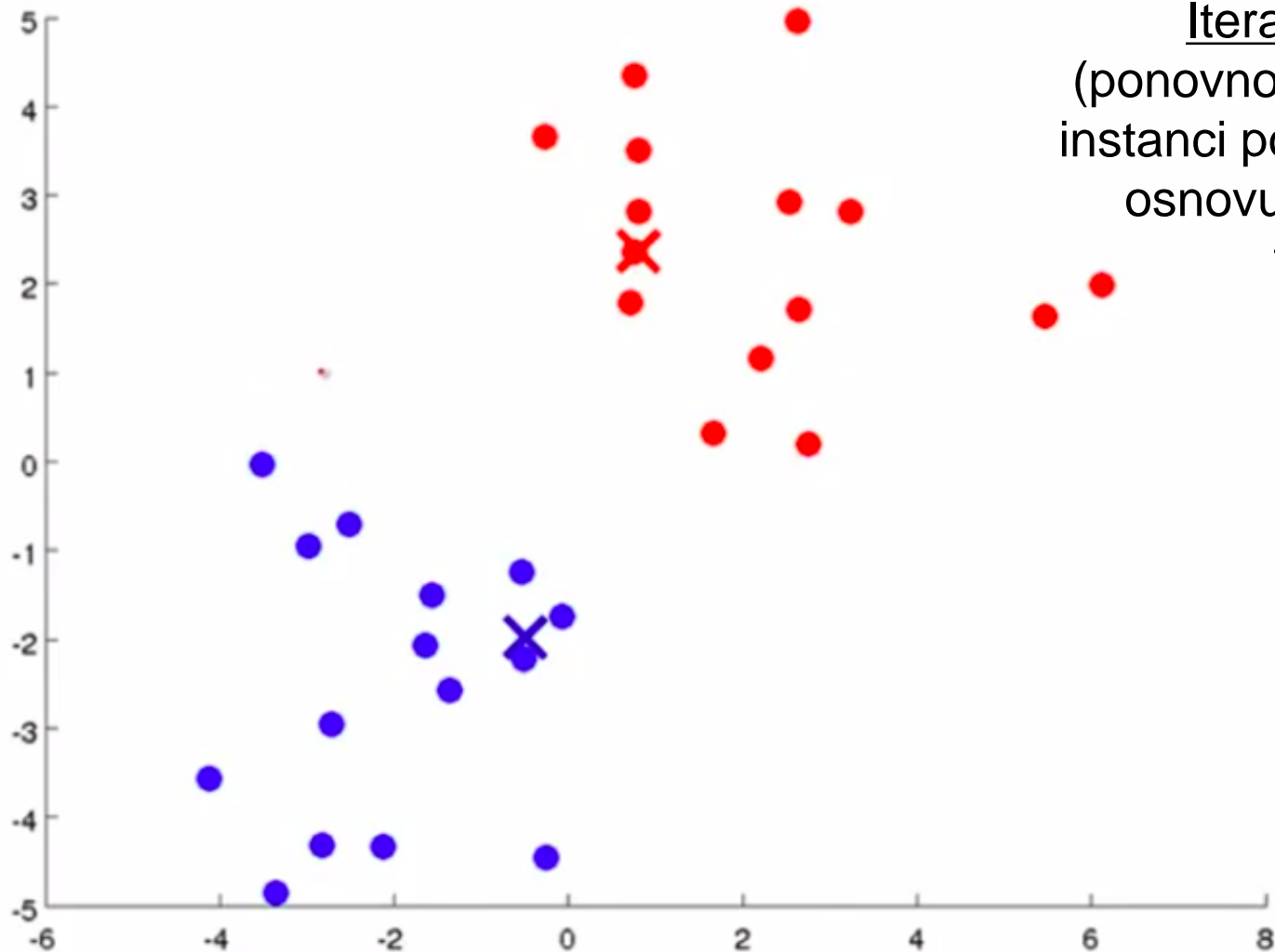


# K-MEANS: PRIMER



Iteracija 2, korak 2:  
Ponovno određivanje  
novog težišta za svaki  
klaster

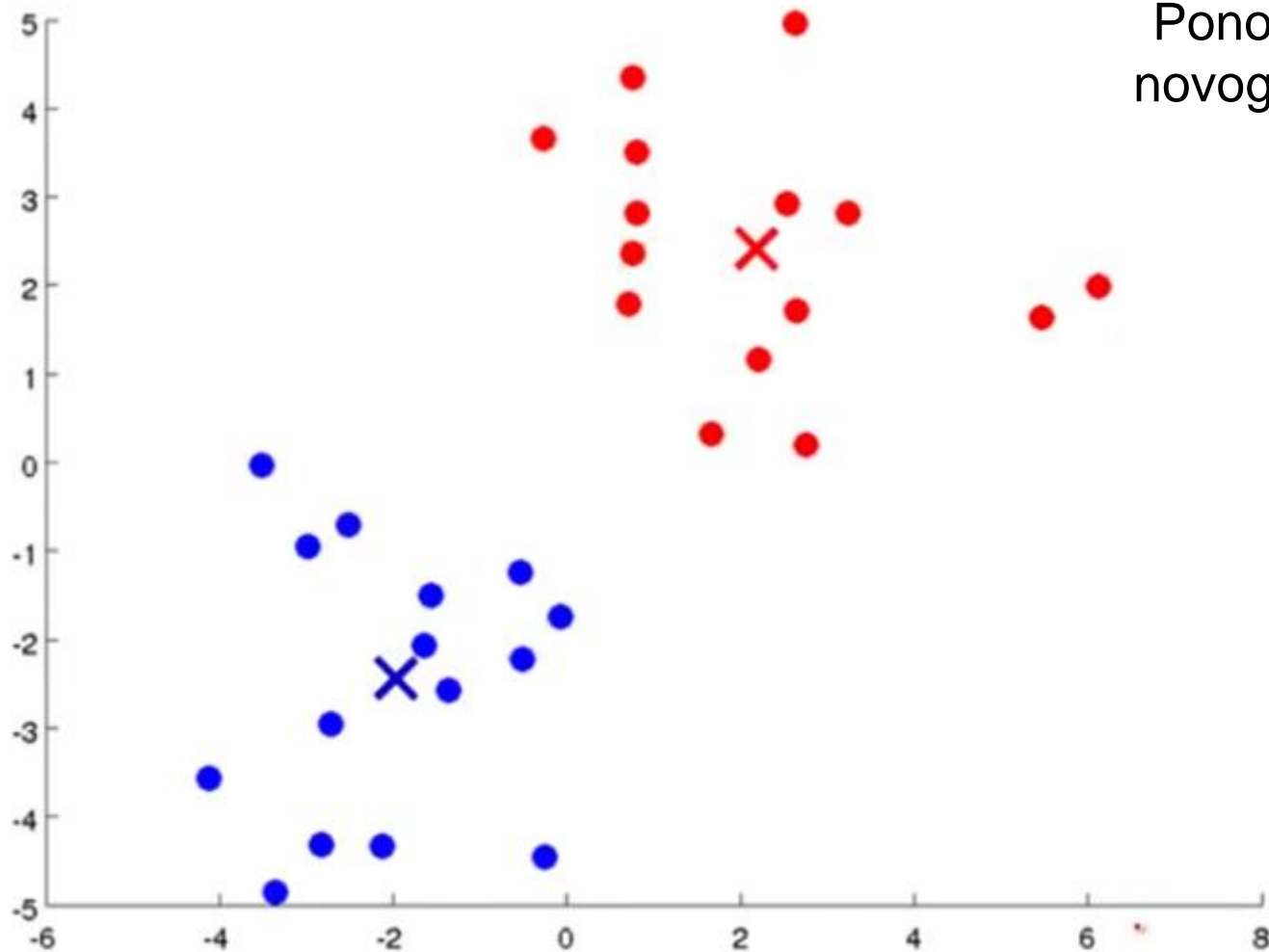
# K-MEANS: PRIMER



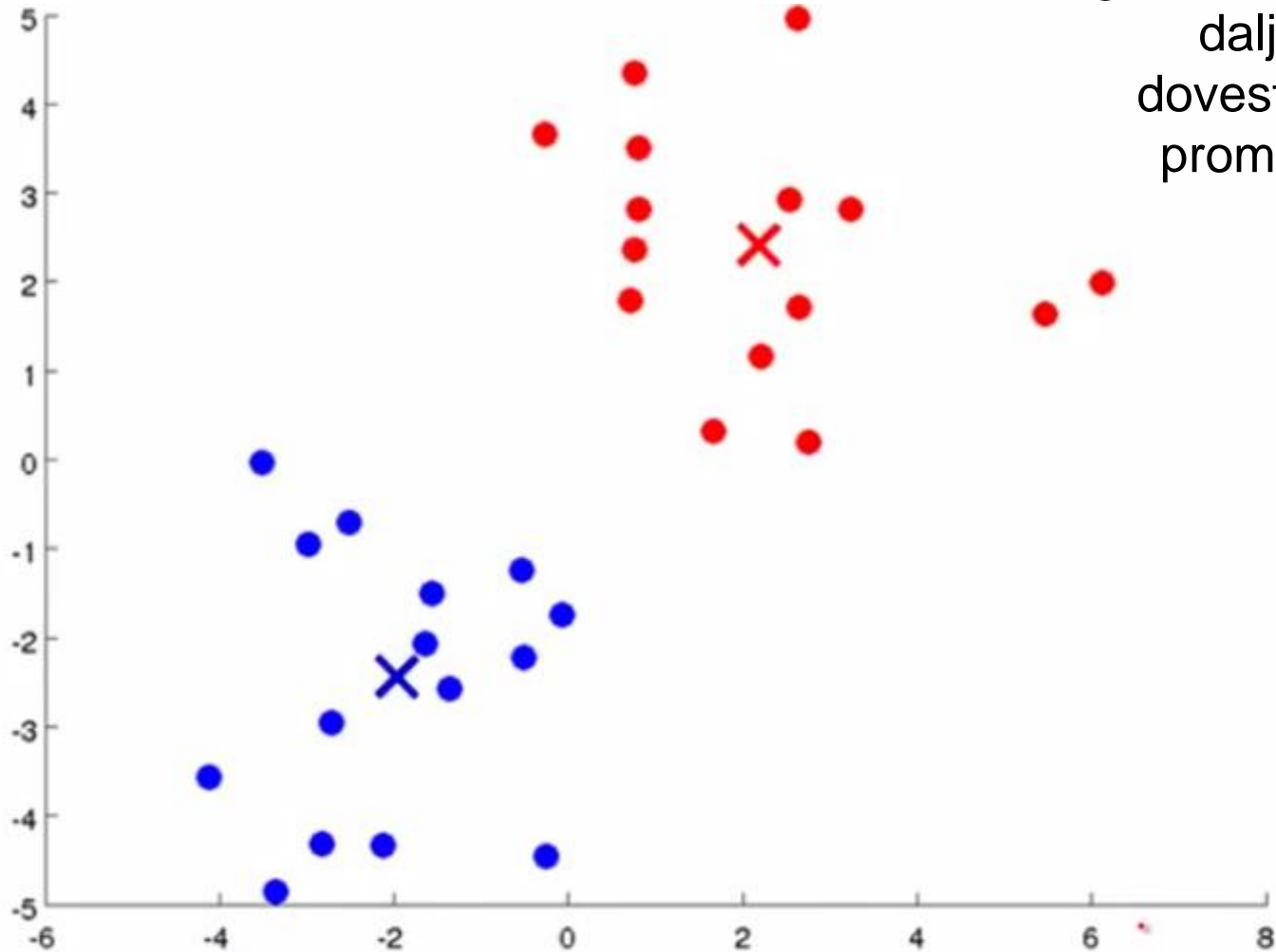
Iteracija 3, korak 1:  
(ponovno) razvrstavanje  
instanci po klasterima na  
osnovu udaljenosti od  
težišta klastera

# K-MEANS: PRIMER

Iteracija 3, korak 2:  
Ponovno određivanje  
novog težišta za svaki  
klaster



# K-MEANS: PRIMER



Algoritam već konvergira:  
dalje iteracije neće  
dovesti do značajnijih  
promena i proces se  
zaustavlja

# K-MEANS: ALGORITAM

Ulaz:

- $K$  - broj klastera
- (neobeležen) skup za trening sa  $m$  instanci; svaka instanca u skupu je vektor opisan sa  $n$  atributa  $(x_1, x_2, \dots, x_n)$
- *max* - max broj iteracija (opcionni parametar)

# K-MEANS: ALGORITAM

Koraci:

1) Inicijalni izbor težišta klastera, slučajnim izborom

- težišta se biraju iz skupa instanci za trening, tj.  $K$  instanci za trening se nasumično izabere i proglašeni za težišta

2) Ponoviti

- 1) *Grupisanje po klasterima*: za svaku instancu iz skupa za trening,  $i = 1, m$ , 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  $\leq max$

# K-MEANS ALGORITAM: FUNKCIJA KOŠTANJA

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

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

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

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

$\mu_j$  – težište klastera  $j$ ,  $j=1, K$

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

Ova funkcija se zove i funkcija distorzije (distortion function)

# K-MEANS ALGORITAM: FUNKCIJA KOŠTANJA

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

Minimizacija funkcije koštanja  $\mathbf{J}$  kroz K-means algoritam:

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



# K-MEANS: EVALUACIJA

Kriterijumi za procenu “kvaliteta” kreiranih klastera:

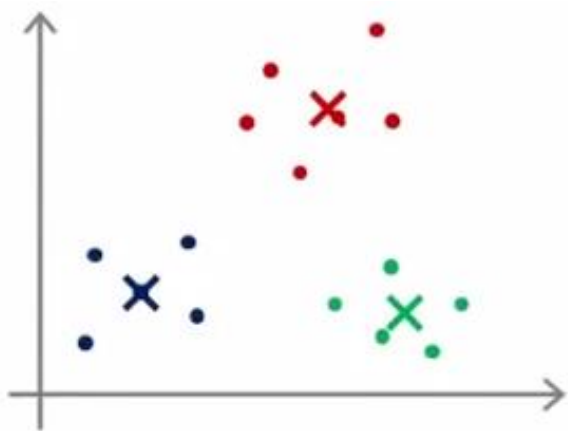
- Međusobna udaljenost težišta
  - što su težišta dalje jedno od drugog, to je stepen preklapanja klastera manji, i njihov kvalitet viši
- St. devijacija pojedinačnih instanci u odnosu na težište
  - što je st. devijacija manja, to su instance tešnje grupisane oko težišta i klasteri se smatraju boljim
- Suma kvadrata unutar klastera (*within cluster sum of squares*)
  - suma kvadrata odstupanja instanci u okviru klastera od centra klastera

# K-MEANS:

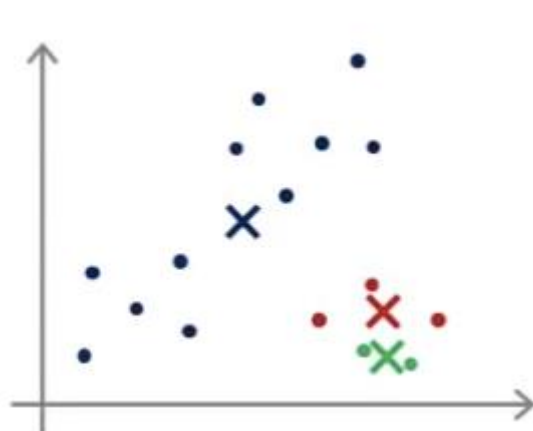
## PROBLEM INICIJALNOG IZBORA TEŽIŠTA

Zavisno od inicijalnog izbora težišta:

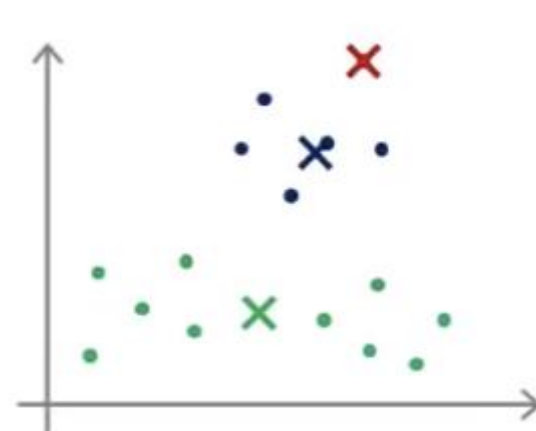
- K-means algoritam može konvergirati brže ili sporije;
- Takođe, može “upasti” u tzv. lokalni minimum i dati loše rešenje
  - reč je o lokalnom min. funkcije koštanja



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

Alternativa: [K-means++ algoritam](#)

# K-MEANS: KAKO ODREDITI K?

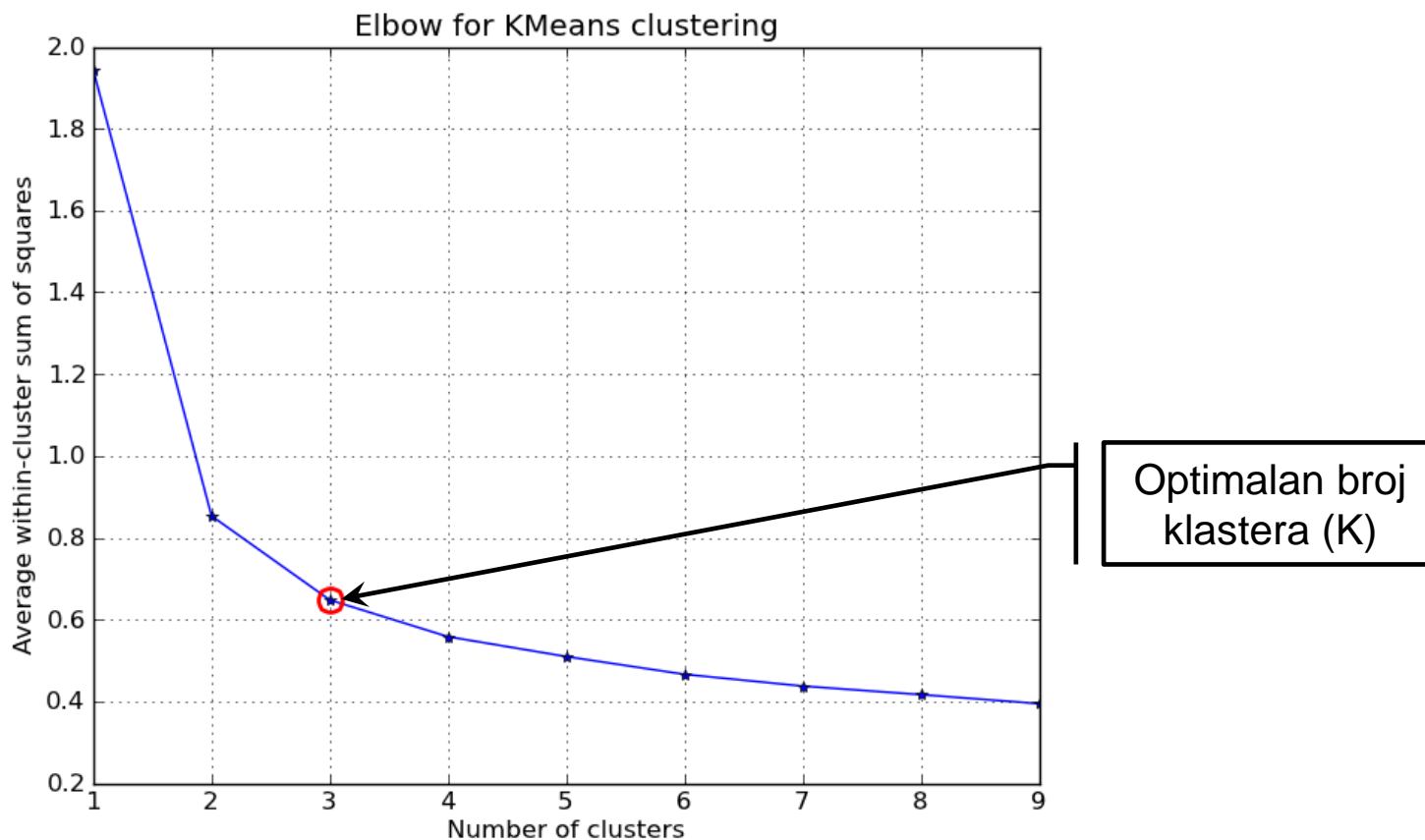
Kako odrediti broj klastera K?

- 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\*
- 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

\*Na primer, *within cluster sum of squared errors*

# K-MEANS: KAKO ODREDITI K?

Kada ne posedujemo znanje o fenomenu/pojavi koja je u osnovi zadatka klasterizacije



**EXPECTATION  
MAXIMIZATION (EM)  
ALGORITAM**

# PROBABILISTIČKA KLASTERIZACIJA

EM klasterizacija spada u grupu probabilističkih pristupa klasterizaciji

Osnovna ideja: instance *ne* pripadaju jednom i samo jednom klasteru, već svaka instanca sa određenom verovatnoćom pripada svakom od klastera

# PROBABILISTIČKA KLASTERIZACIJA

Ovaj oblik klasterizacije podrazumeva:

- Svaki klaster je određen/opisan raspodelom verovatnoća
  - to može biti jedna raspodela, zajednička za sve attribute kojima su instance opisane, ili više, po jedna za svaki od atributa
  - ove raspodele određuju verovatnoću da instance koja pripadaju datom klasteru bude opisane određenim skupom vrednosti (atributa)
- Postoji, takođe, i raspodela verovatnoća koja karakteriše pripadanost klasteru tzv. *prior probability*



# PROBABILISTIČKA KLASTERIZACIJA

Razmotrimo najjednostavniji oblik prob. klasterizacije:

- Sve instance su opisane jednim numeričkim atributom koji ima Normalnu raspodelu u svim klasterima (ukupno  $k$  klastera)
- Svaki klaster ( $C_i$ ) ima specifične vrednosti parametara Normale raspodele – srednje vrednosti ( $\mu_i$ ) i st. devijacije ( $\sigma_i$ )
- $p_i$  predstavlja tzv *prior probability*  $i$ -tog klastera ( $C_i$ ) tj. verovatnoću da nasumice izabrana instanca pripada klasteru  $C_i$

# PROBABILISTIČKA KLAŠTERIZACIJA

Razmotrimo najjednostavniji oblik prob. klast. (nastavak):

- Pretpostavimo da smo dobili skup instanci za koje znamo da dolaze iz opisanih  $k$  klastera, ali ne znamo iz kojih, niti znamo parametre modela  $(\mu_i, \sigma_i, p_i, i=1, k)$
- Zadatak / problem koji treba rešiti:
  - odrediti parametre klastera  $(\mu_i, \sigma_i, p_i, i=1, k)$
  - za svaku instancu odrediti verovatnoću pripadnosti svakom od  $k$  klastera

# EM ALGORITAM

Opisani problem se može rešiti primenom postupka koji predstavlja generalizaciju K-means algoritma:

- 1) Inicijalno, definisati broj klastera ( $k$ ) i nasumice izabrati vrednosti parametara modela ( $\mu_i, \sigma_i, p_i, i=1, k$ )
- 2) Za date vrednosti parametara, za svaku instancu iz dataset-a, izračunati verovatnoću pripadanja svakom od klastera
- 3) Na osnovu izračunatih verovatnoća pripadnosti klasterima, odrediti nove vrednosti parametara modela

Iterativno ponavljati korake 2) i 3) dok vrednosti parametara ne počnu da konvergiraju

Opisani postupak predstavlja suštinu EM algoritma

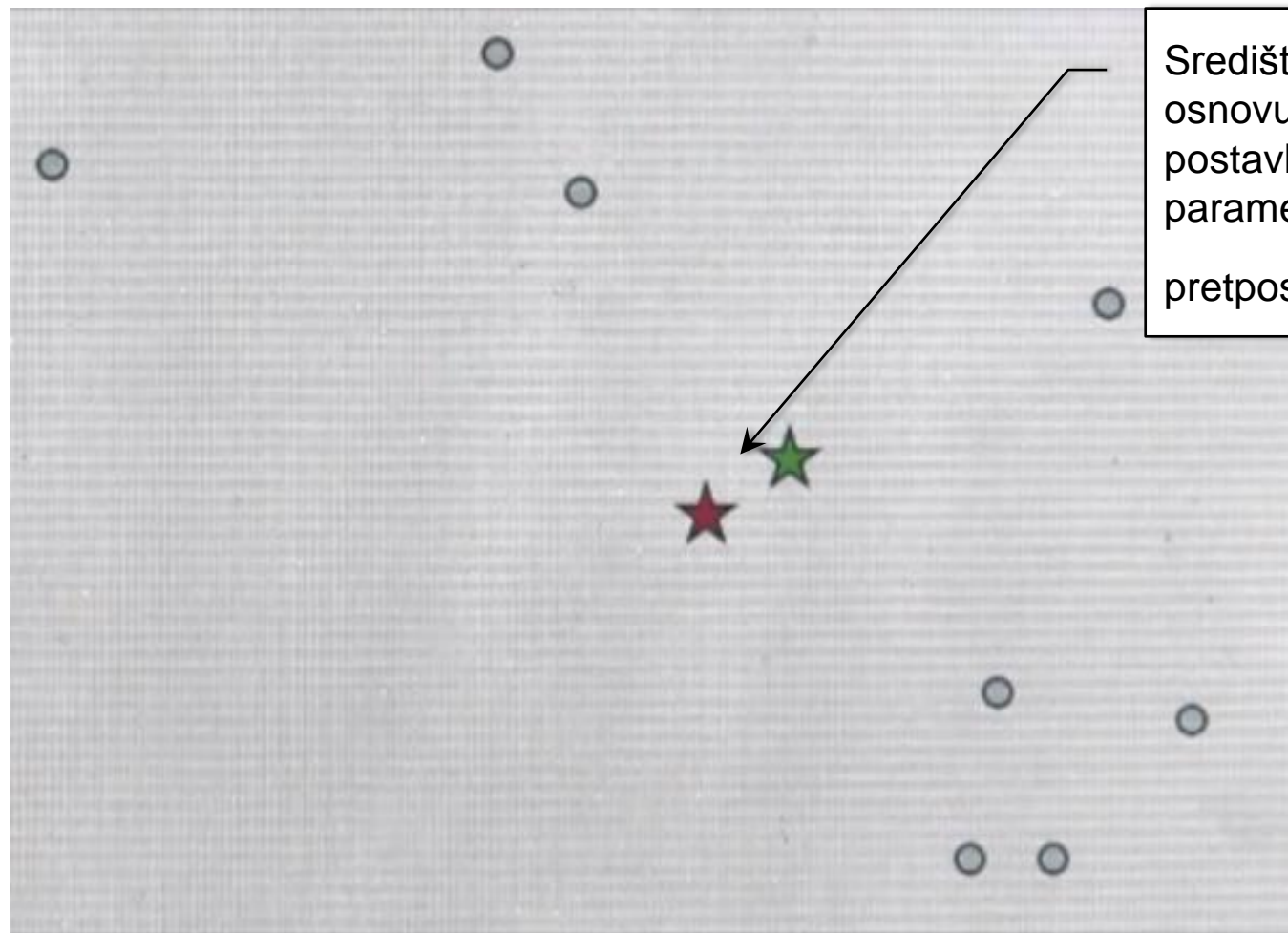
# EM ALGORITAM

Sastoji se iz dva osnovna koraka:

- **E (expectation) korak** – u ovom koraku podrazumevamo da znamo vrednosti parametara modela i na osnovu njih, za svaku instancu, računamo verovatnoću pripadanja svakom od klastera
- **M (maximization) korak** – na osnovu datih instanci, računamo (ponovo) vrednosti parametara modela; maksimizacija se odnosi na usklađivanje (parametara) modela sa datim podacima

Ovi koraci se ponavljaju sve dok algoritam ne počne da konvergira

# EM ALGORITAM: INICIJALIZACIJA



Središta klastera na osnovu inicijalno postavljenih vrednosti parametara modela; pretpostavka:  $k=2$

# EM ALGORITAM: E KORAK

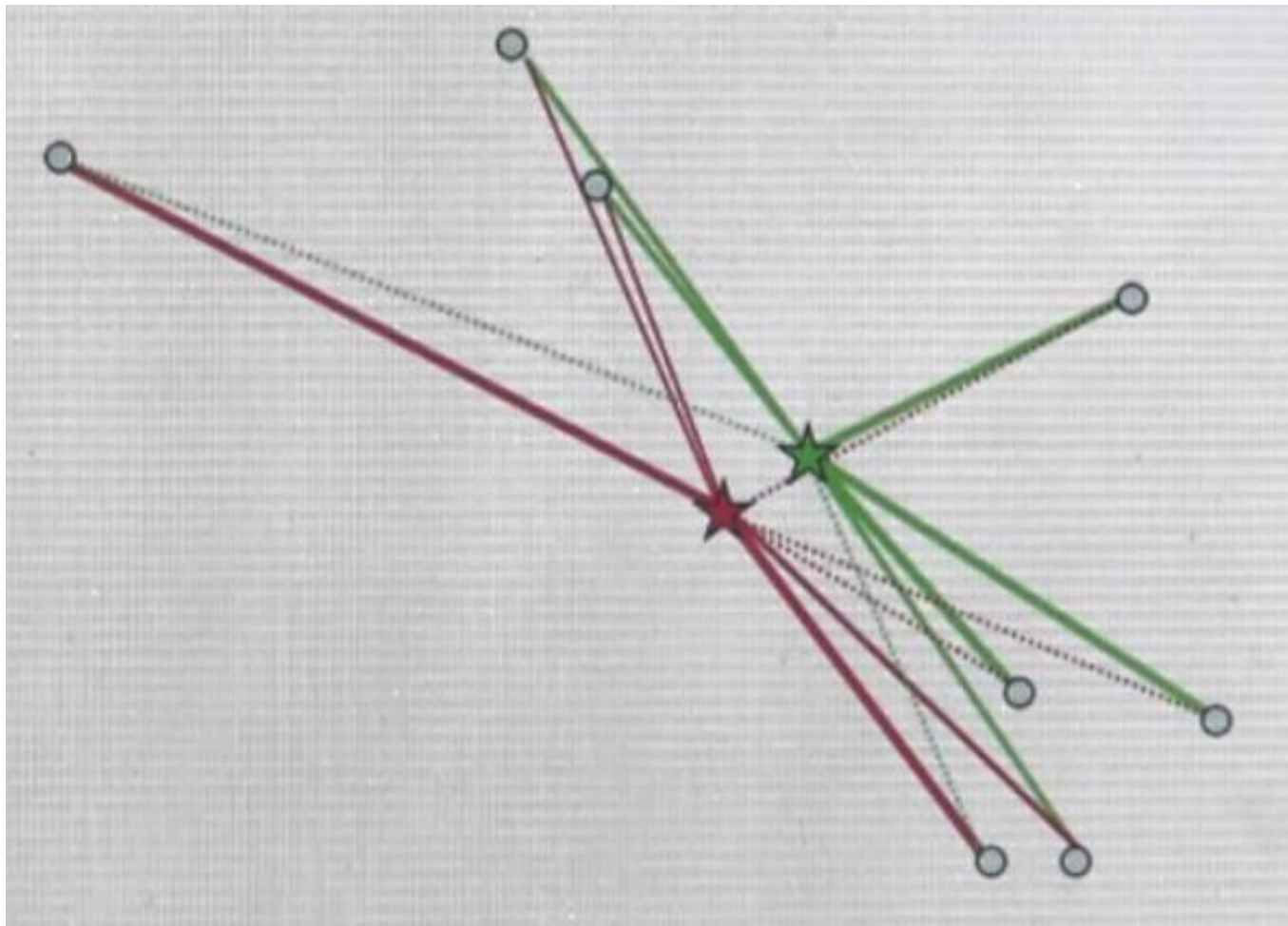
Za svaku instancu iz dataset-a  $x_j$  ( $j=1,n$ ), računamo verovatnoću pripadanja  $i$ -tom klasteru ( $C_i$ ,  $i=1,k$ )

$$e_{ij} = p_i * P(x_j | C_i)$$

$P(x_j | C_i)$  se računa primenom funkcije Normalne raspodele  $f(x; \mu, \sigma)$

Podsećanje: pretpostavka je da su nam parametri –  $\mu_i$ ,  $\sigma_i$ ,  $p_i$  – poznati za svako  $i$ ,  $i=1,k$

# EM ALGORITAM: E KORAK



Debljina linije ukazuje na verovatnoću pripadnosti određenom klasteru tj. odražava izračunatu vrednost za  $e_{ij}$

# EM ALGORITAM: M KORAK

Određuju se nove vrednosti parametara modela:

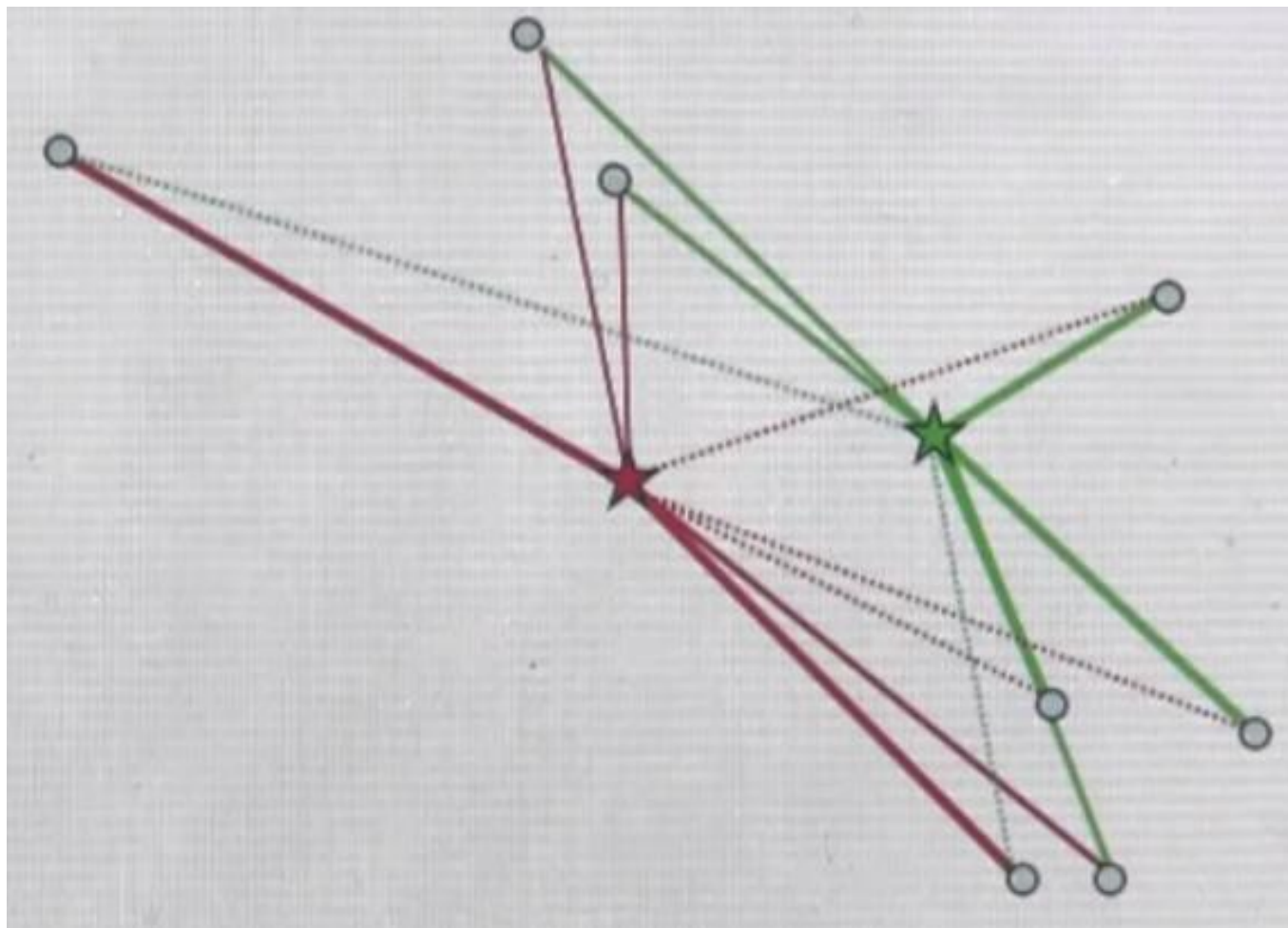
$$\text{prior probability: } p_i = \sum_j \frac{e_{ij}}{n}$$

$$\text{srednja vrednost: } \mu_i = \frac{\sum_j e_{ij} * x_j}{\sum_j e_{ij}}$$

$$\text{varijansa: } \sigma_i^2 = \frac{\sum_j e_{ij} * (x_j - \mu_i)^2}{\sum_j e_{ij}}$$



# EM ALGORITAM: M KORAK



Središta klastera se pomeraju na osnovu novo izračunatih vrednosti parametara modela

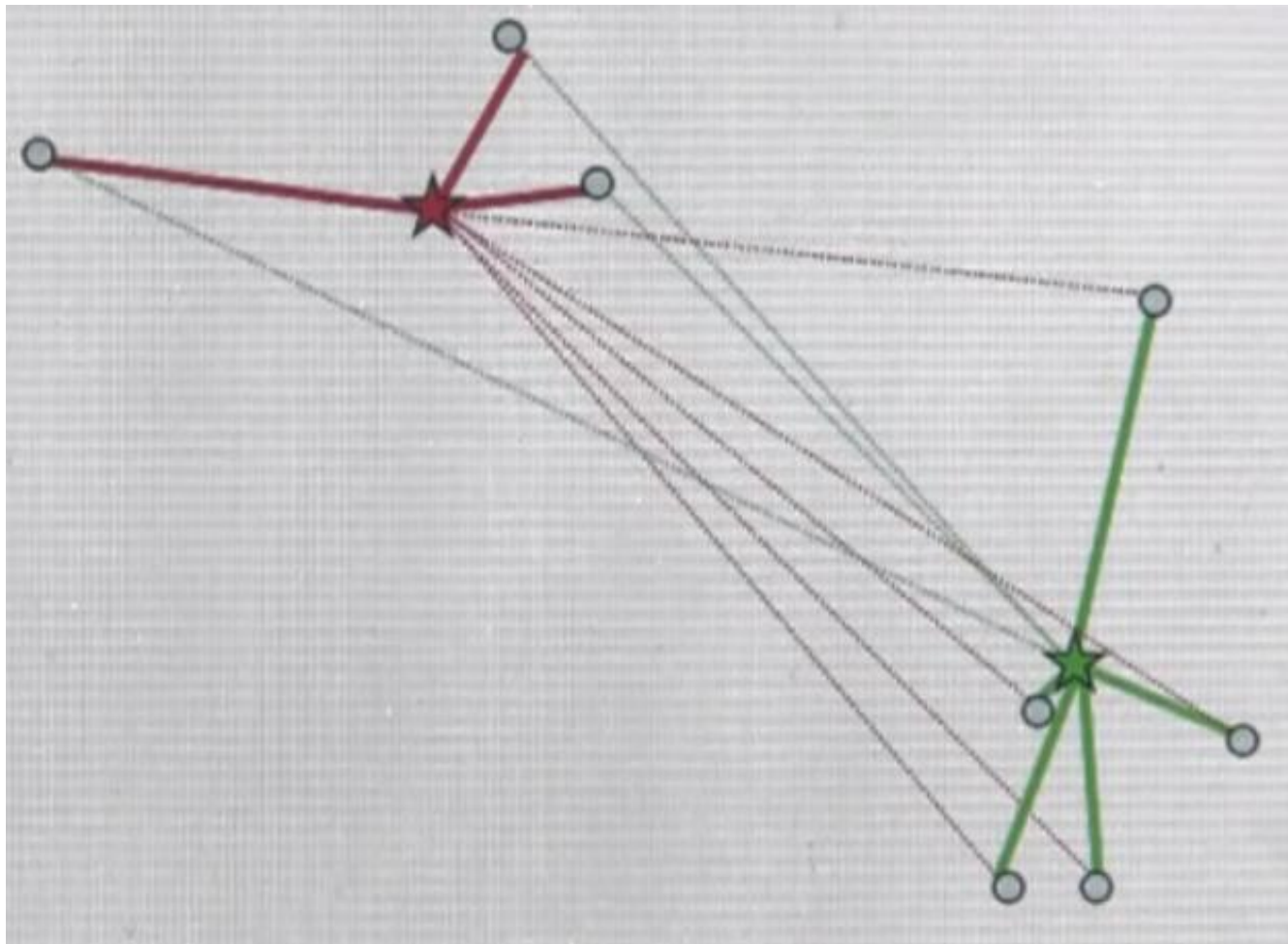
# EM ALGORITAM: KONVERGENCIJA

Koraci algoritma se ponavljaju sve dok ima značajne promene (tj. porasta) loga ukupne verovatnoće modela (*overall log-likelihood*):

$$\log P(x) = \log \sum_i (p_i * P(x|C_i))$$

Tipično, ova vrednost intenzivno raste tokom prvih nekoliko iteracija algoritma, a zatim vrlo brzo dolazi u stanje gde maltene nema promene

# EM ALGORITAM: KONVERGENCIJA



Stanje konvergencije (parametara) modela

# EM ALGORITAM

Za EM algoritam se pokazuje da sigurno konvergira ka maksimumu *log-likelihood* funkcije

Međutim, reč je o lokalnom maksimumu, koji ne mora biti i globalni maksimum

Da bi se obezbedilo dostizanje globalnog maksimuma, potrebno je celu proceduru ponoviti više puta, uz različit izbor inicijalnih vrednosti parametara modela

Na kraju se, naravno, bira konfiguracija koja generiše najvišu vrednost *log-likelihood* funkcije

# EM ALGORITAM

Mi smo razmatrali najjednostavniji slučaj primene EM algoritma; ali EM se može na isti način primeniti i na složenije probleme

- Umesto jednog atributa, svaka instanca može biti opisana proizvoljnim brojem atributa, pod uslovom da su atributi međusobno nezavisni
  - u tom slučaju, verovatnoće atributa se množe kako bi se dobila ukupna verovatnoća za svaku instancu
- Atributi ne moraju biti numerički, već mogu biti i nominalni
  - u tom slučaju, Normalna raspodela se ne može primeniti;
  - nominalni atribut sa  $v$  mogućih vrednosti se predstavlja preko  $v$  brojeva koji predstavljaju verovatnoće vrednosti nominalnog atributa

# ZAHVALNICA I PREPORUKA

## Stanford

### Machine Learning

Andrew Ng

Learn about the most effective machine learning techniques, and gain practice implementing them and getting them to work for yourself.

**Workload:** 5-7 hours/week

**Taught In:** English

**Subtitles Available In:** English

Preview



**Sessions:**

Oct 14th 2013 (10 weeks long)

Sign Up

Apr 22nd 2013 (10 weeks long)

Sign Up



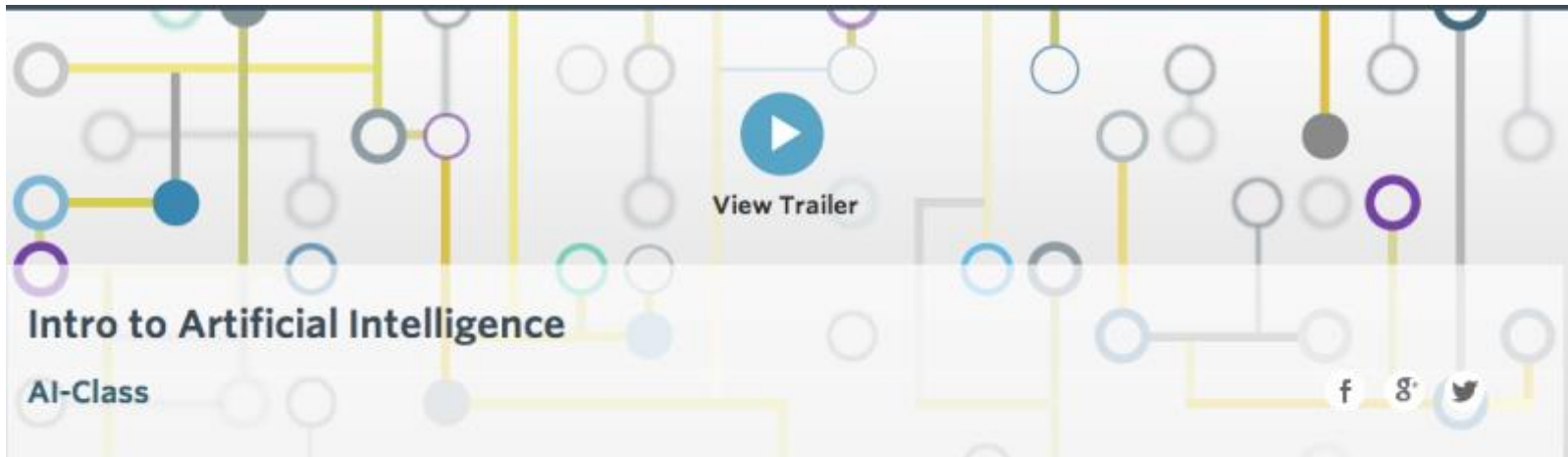
Coursera:

<https://www.coursera.org/course/ml>

Stanford YouTube channel:

[http://www.youtube.com/view\\_play\\_list?p=A89DCFA6ADACE599](http://www.youtube.com/view_play_list?p=A89DCFA6ADACE599)

# ZAHVALNICA I PREPORUKA



Intermediate

Join 52,215 Students

## Class Summary

The objective of this class is to teach you modern AI. You will learn about the basic techniques and tricks of the trade. We also aspire to excite you about the field of AI.

## Course Instructors



### Peter Norvig

INSTRUCTOR

Peter Norvig is Director of Research at Google Inc. He is also a Fellow of the American Association for Artificial Intelligence and the Association for Computing Machinery. Norvig is co-author of the popular textbook *Artificial Intelligence: A Modern Approach*. Prior to joining Google he was the head of the Computation Sciences Division at NASA Ames Research Center.



### Sebastian Thrun

INSTRUCTOR

Sebastian Thrun is a Research Professor of Computer Science at Stanford University, a Google Fellow, a member of the National Academy of Engineering and the German Academy of Sciences. Thrun is best known for his research in robotics and machine learning, specifically his work with self-driving cars.

URL: <https://www.udacity.com/course/cs271>

(Anonimni) upitnik za vaše kritike,  
komentare, predloge:

<http://goo.gl/cqdp3l>