ОСНОВИ РАЧУНАРСКЕ ИНТЕЛИГЕНЦИЈЕ

КЛАСТЕРОВАЊЕ

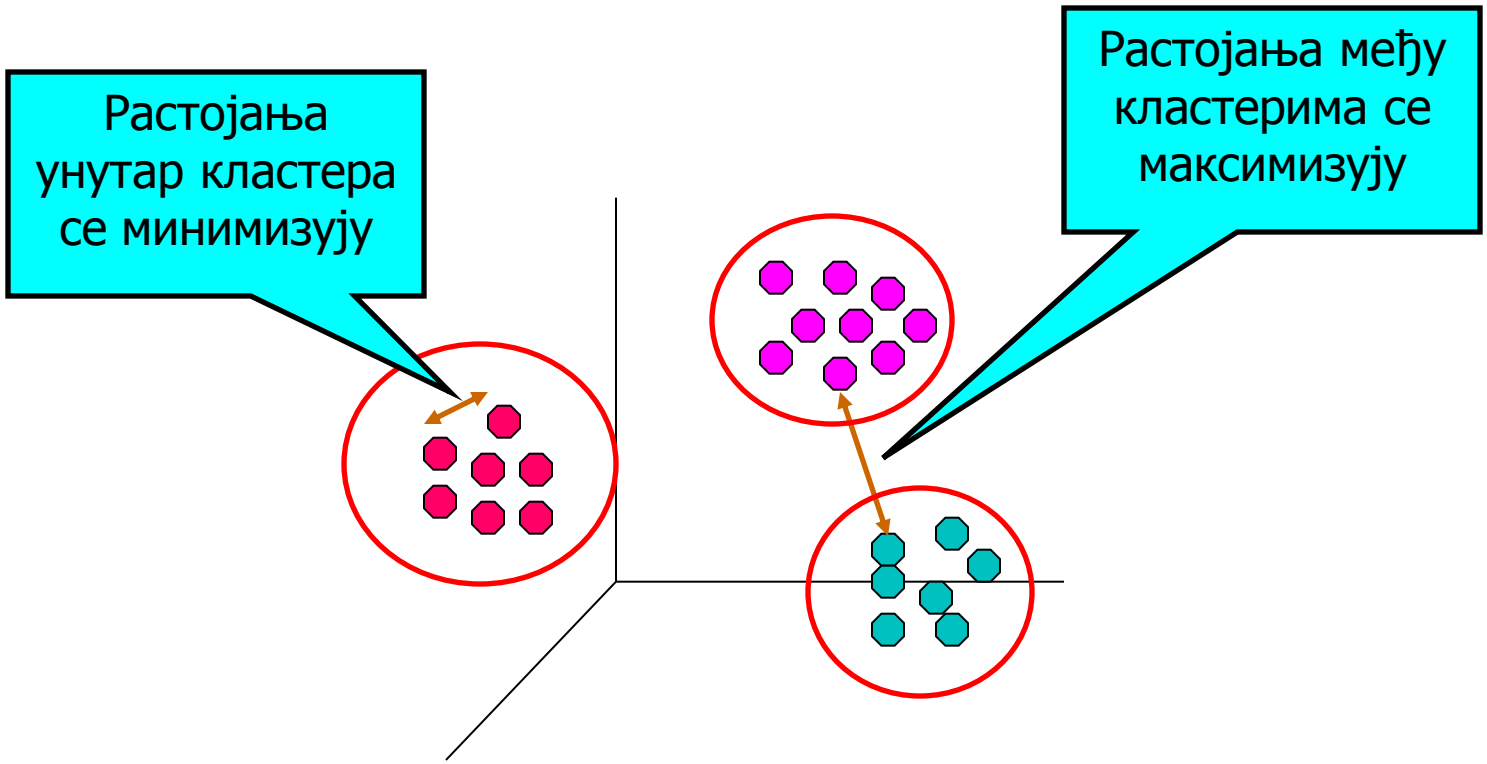
Предавач: Александар Ковачевић

Слајдови засновани на поглављу 8 књиге:

Introduction to Data Mining, Tan, Steinbach, Kumar

Шта је кластер анализа?

- Налажење група објеката таквих да су објекти из групе међусобно слични (или повезани) и да су различити (неповезани) од објеката у другим групама

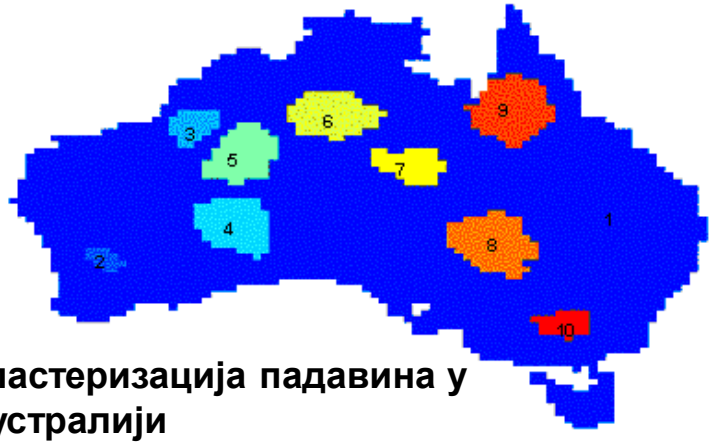


Примене кластер анализе

- **Разумевање**
  - Група повезаних докумената за претраживање, група гена и протеина који имају сличну функционалност, група акција са сличном флуктуацијом цене,...

	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP

- **Сажимање**
  - Смањење величине великих скупова података



Кластеризација падавина у Аустралији

Шта није кастеровање?

- Надгледана класификација
  - Постоји информација о ознакама класа
- Једноставна сегментација
  - Подела студената у различите наставне групе по презимену (алфабетски)
- Резултати упита
  - Груписања су резултат екстерне спецификације

Значење кластеровања може да буде неодређено

Колико кластера?

Шест кластера

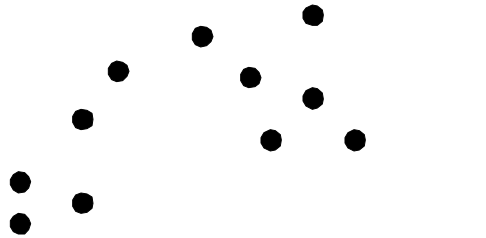
Два кластера

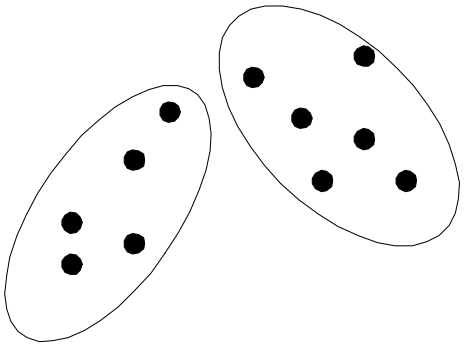
Четири кластера

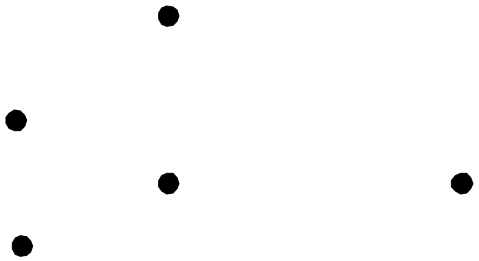
Типови кластеринга

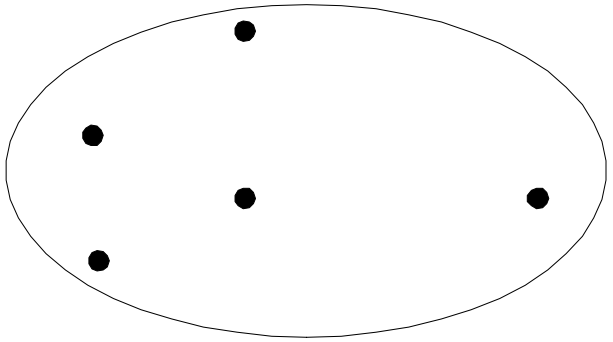
- Кластеринг је скуп кластера
- Важна разлика између хијерархијских и партитивних скупова кластера
- Партитивни кластеринг
  - Подела објеката у непреклапајуће подскупове (кластере) таква да је сваки објекат у тачно једном подскупу
- Хијерархијски кластеринг
  - Скуп угњеждених кластера организованих као хијерархијско стабло

Партитивни кластеринг





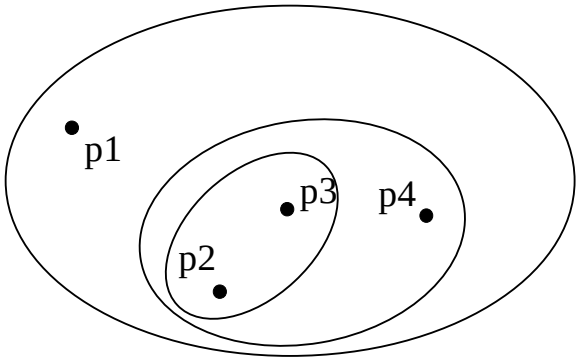




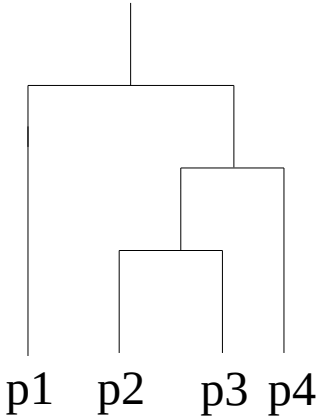
Оригиналне тачке

Партитивни кластеринг

Хијерархијски кластеринг



Хијерархијски кластеринг



Дендограм

Друга аспекти растојања између скупова кластера

- Експклузивни наспрам не-експклузивног
  - Код не-екслузвином кластерингу тачке могу да спадају у више кластера.
  - Репрезентација више класа или ‘граничних’ тачака
- Фази наспрам не-фази
  - Код фази кластеринга, тачка припада сваком кластеру са неком вредношћу између 0 и 1
  - Збир тежина мора да буде 1
- Парцијално наспрам комплетног
  - У неким случајевима желимо да кластеризујемо део података
- Хетерогено наспрам хомогеног
  - Кластери са значајним разликама по величини, облику и густини

Алгоритми за кластеровање

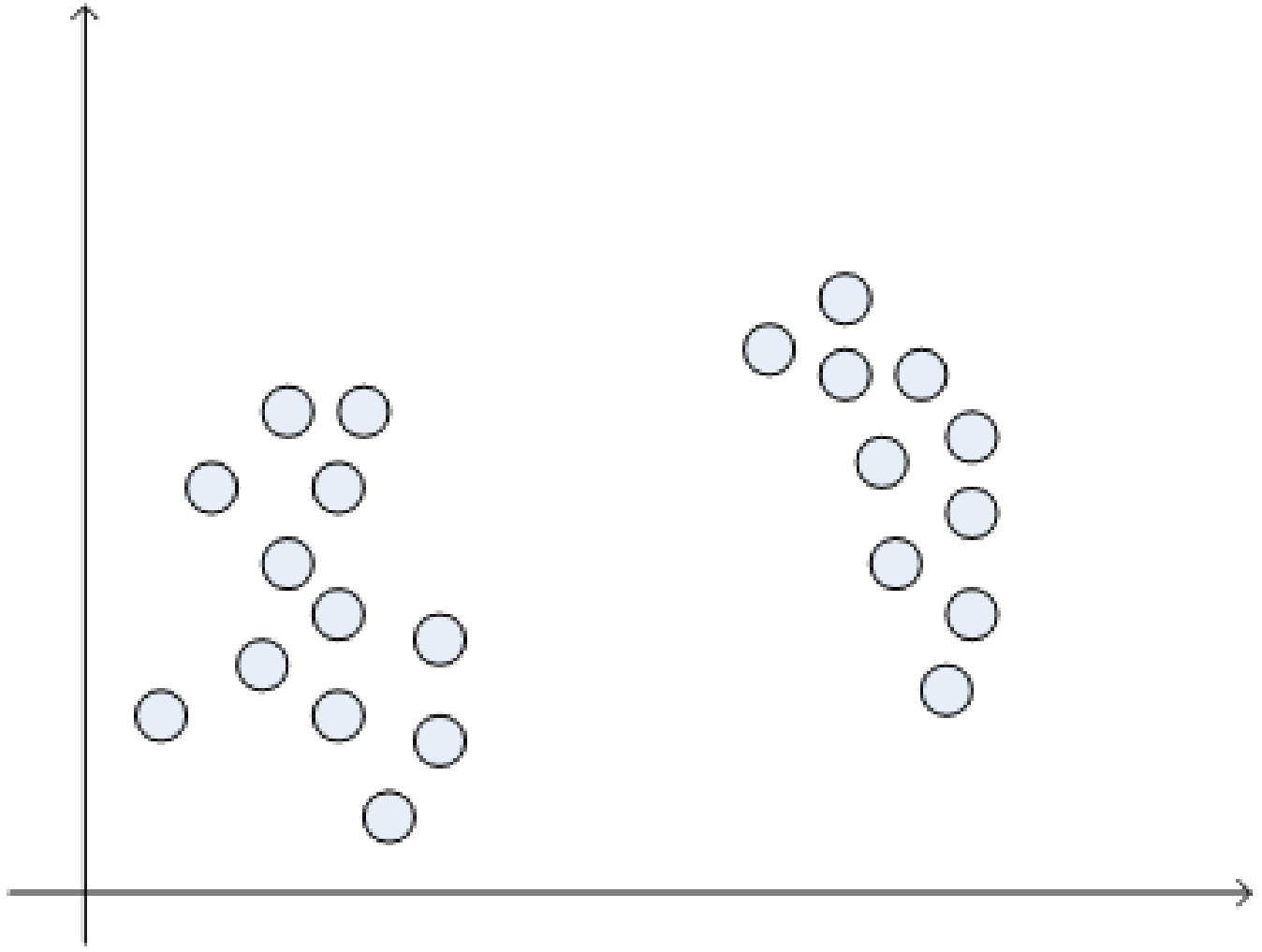
- K-средине и његове варијације
- Хијерархијско кластеровање
- Кластеровање базирано на густини

Кластеровање К-средина

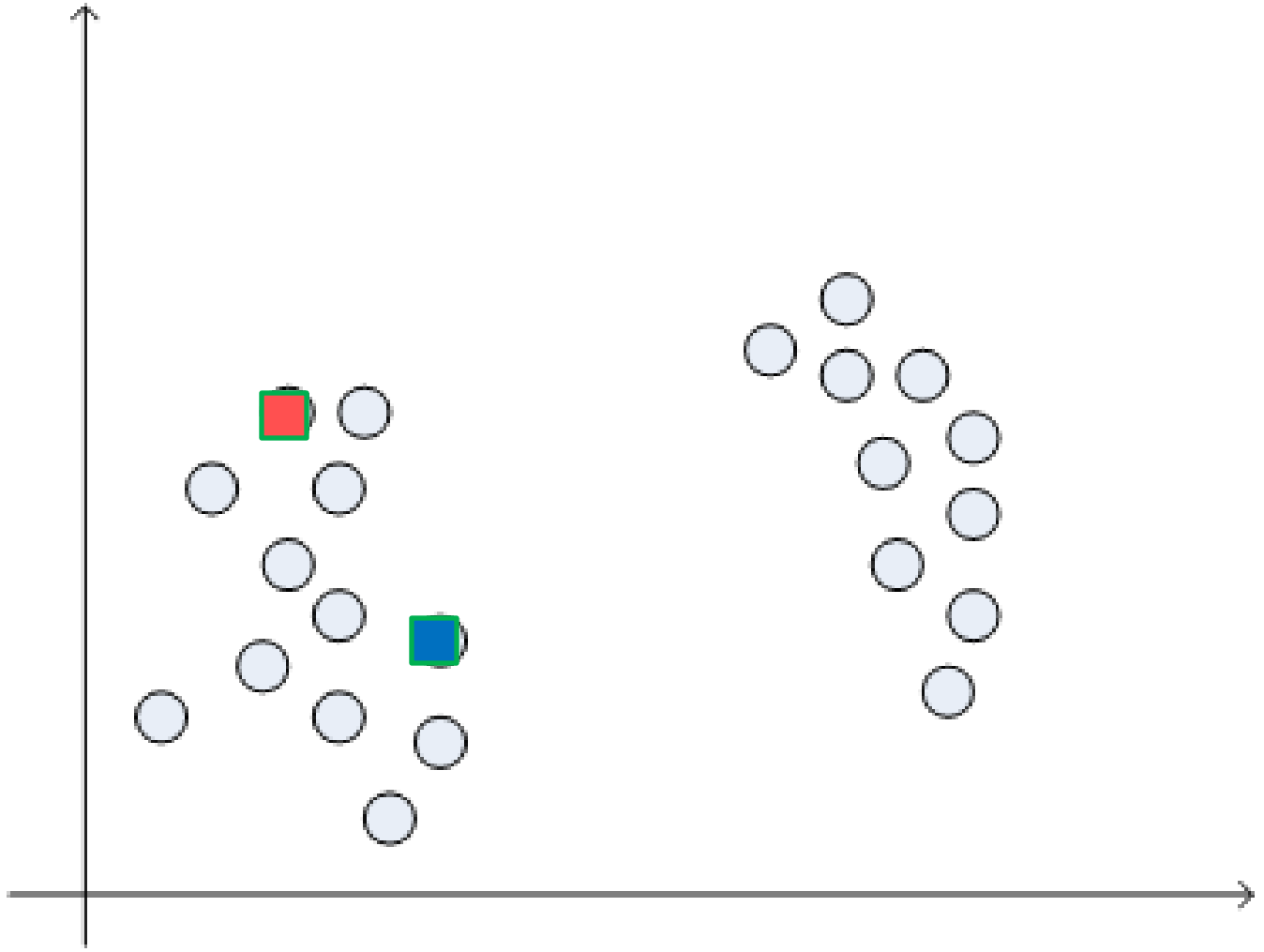
- Партитивни приступ кластеровању
- Сваком кластеру се додељује центроид (центар)
- Свака тачка се сврстава у кластер са најближим центроидом
- Број кластера К мора бити задат
- Основни алгоритам је врло једноставан:

1. Selektovati *K* tačaka za početne centroide
2. **repeat**
    3. Formirati *K* klastera svrstavanjem tačaka u najbliži centroid
    4. Sračunati novi centroid za svaku klasu (na bazi svrstanih tačaka)
5. **until** centroid se ne menja

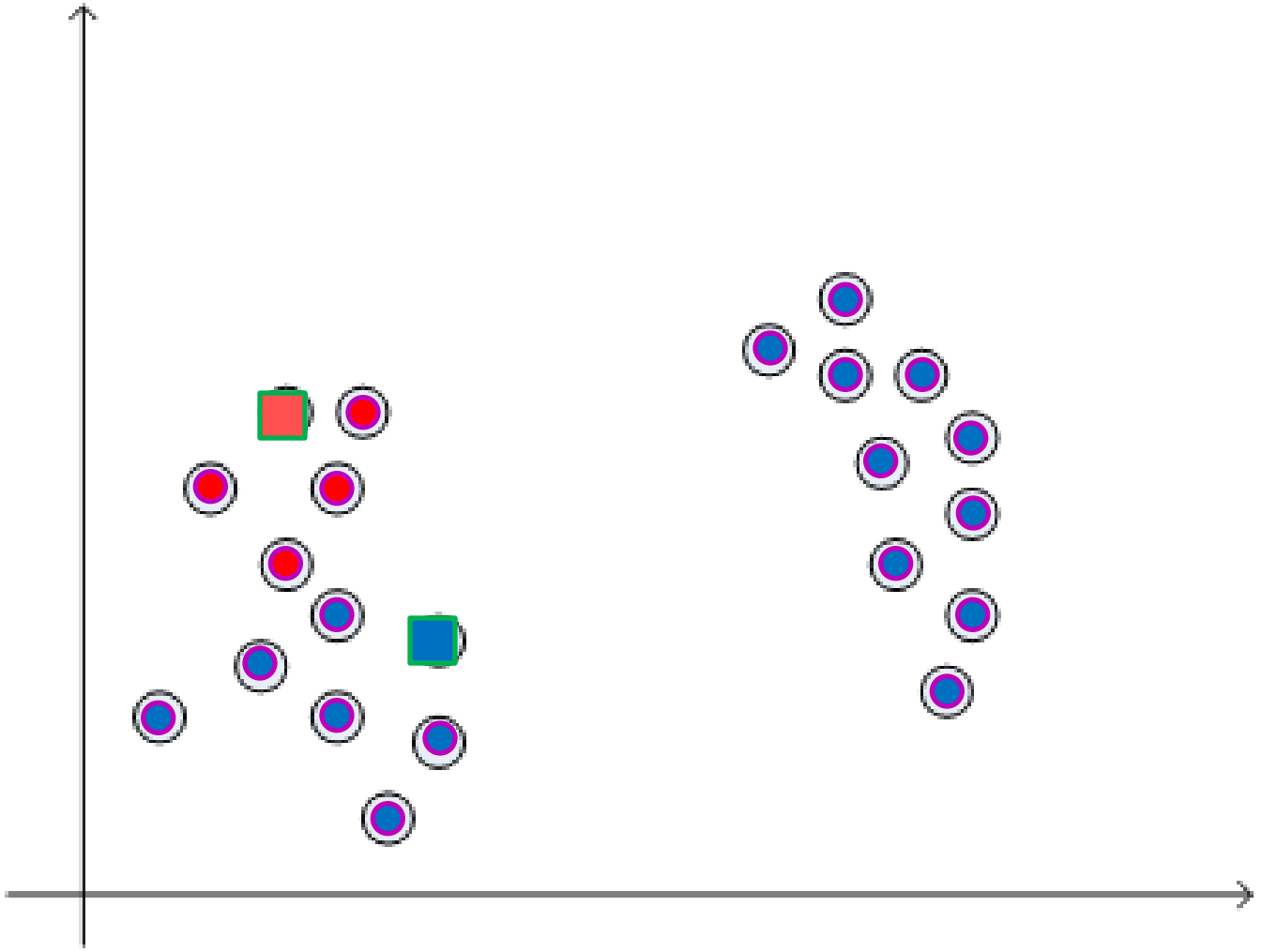
K-Means algoritam



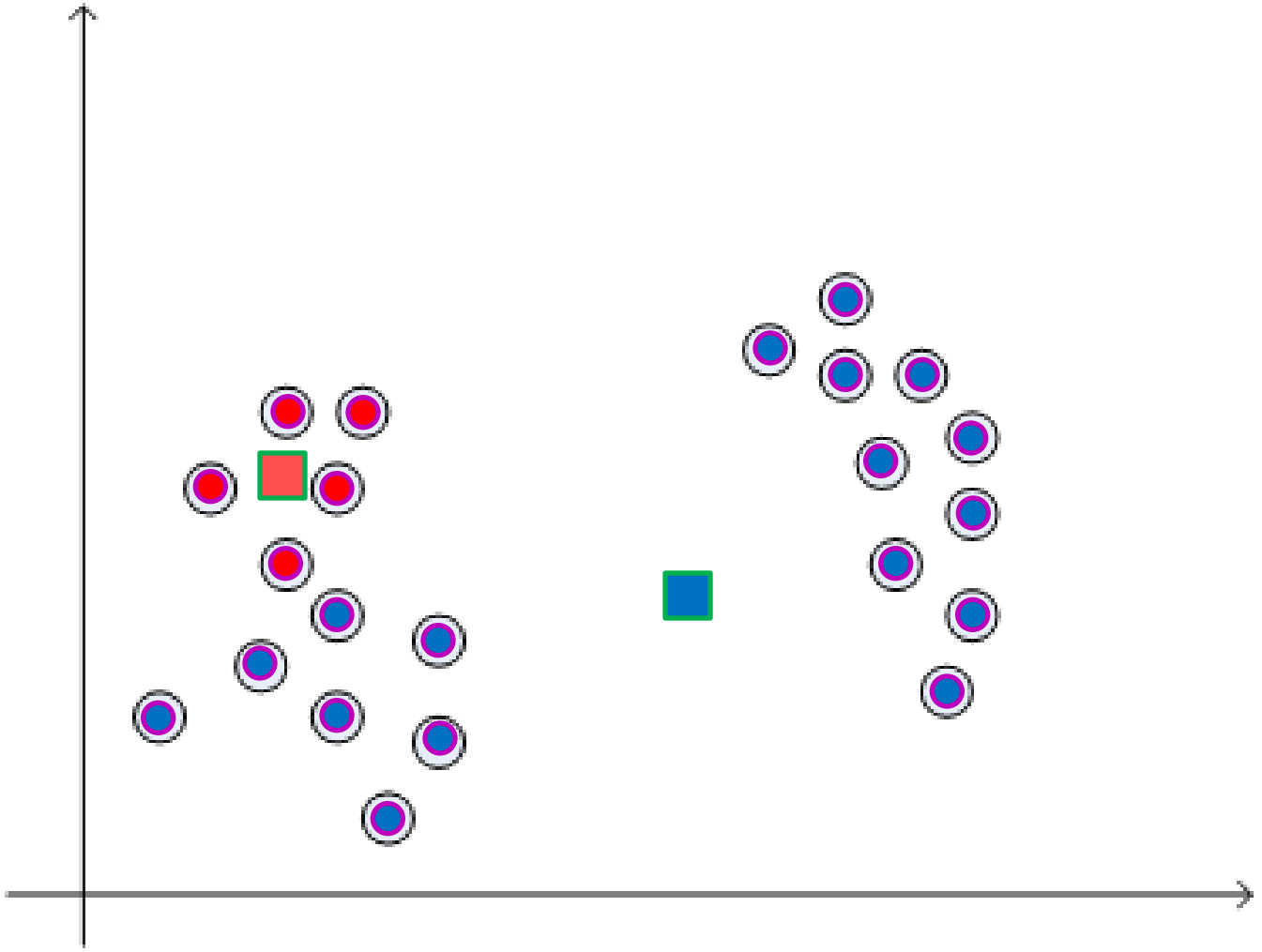
K-Means algoritam



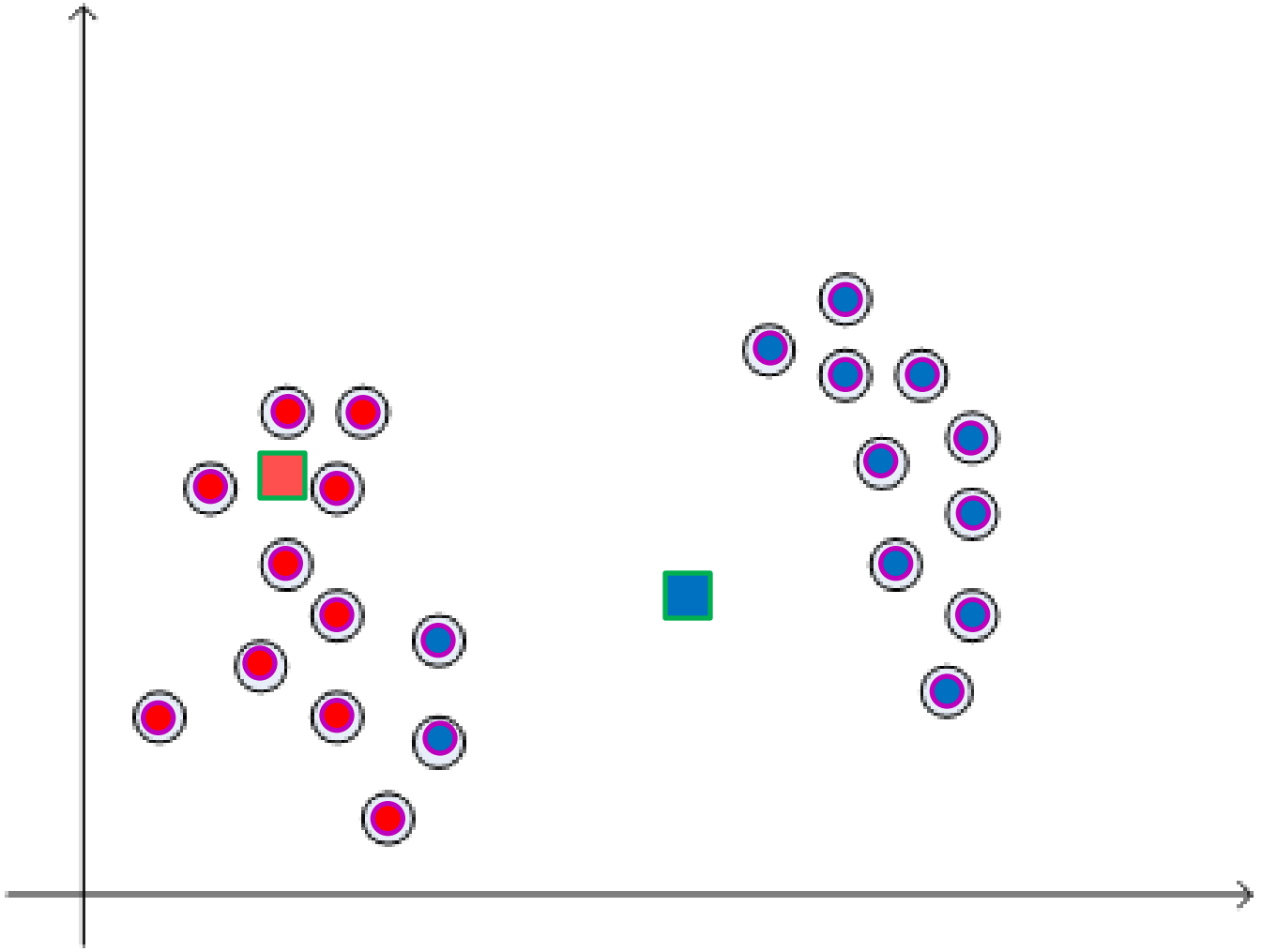
K-Means algoritam



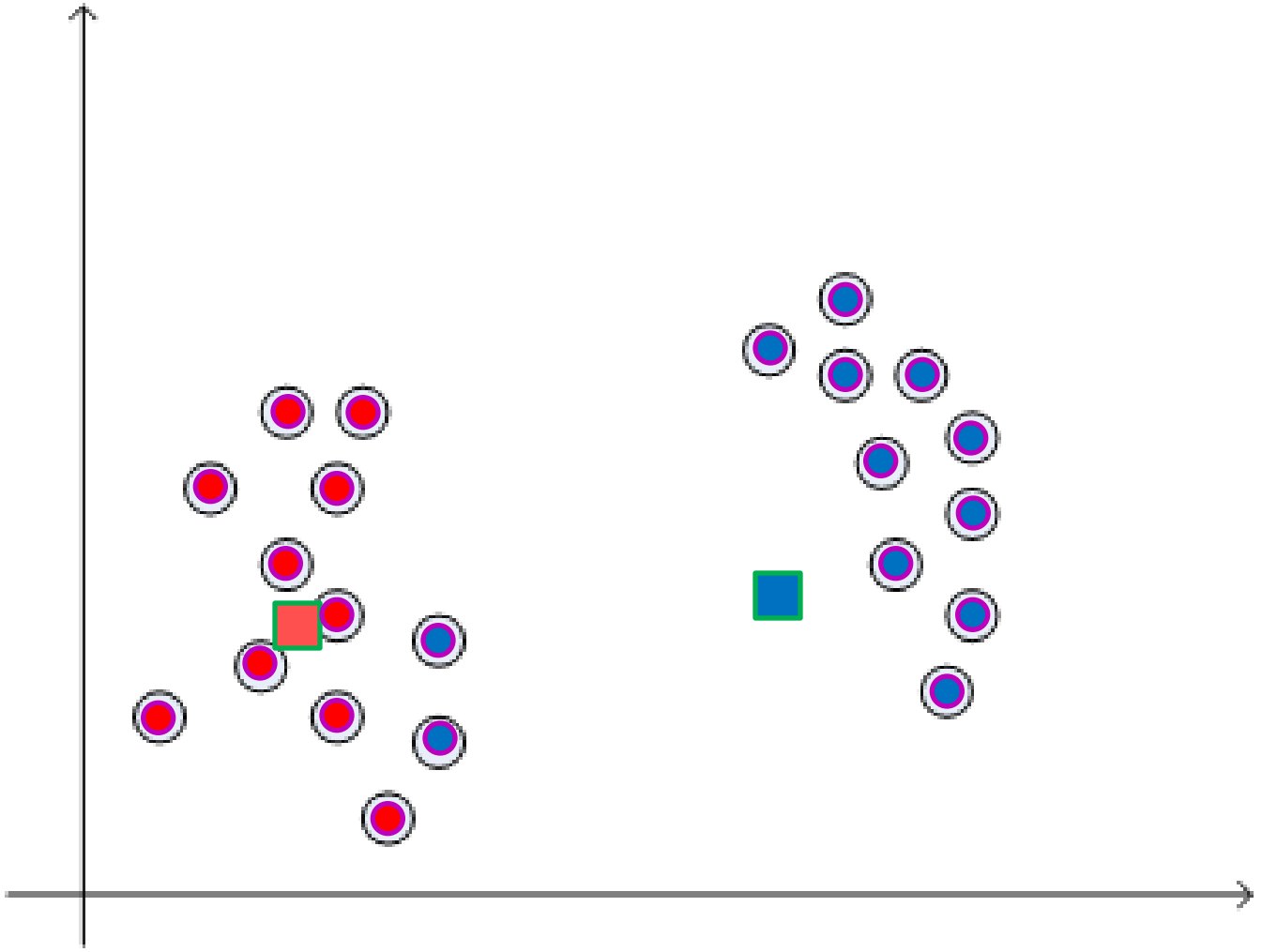
K-Means algoritam



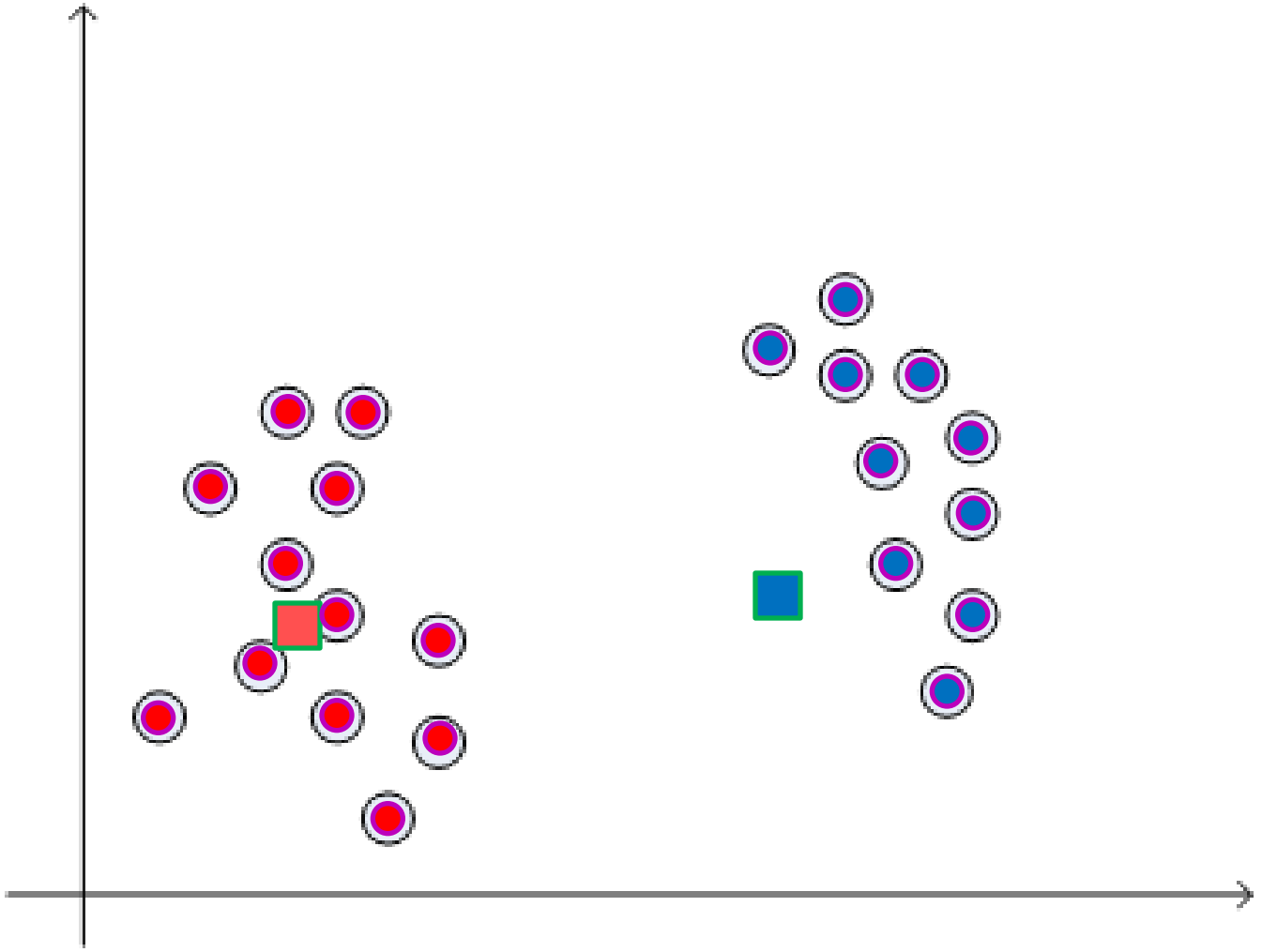
K-Means algoritam



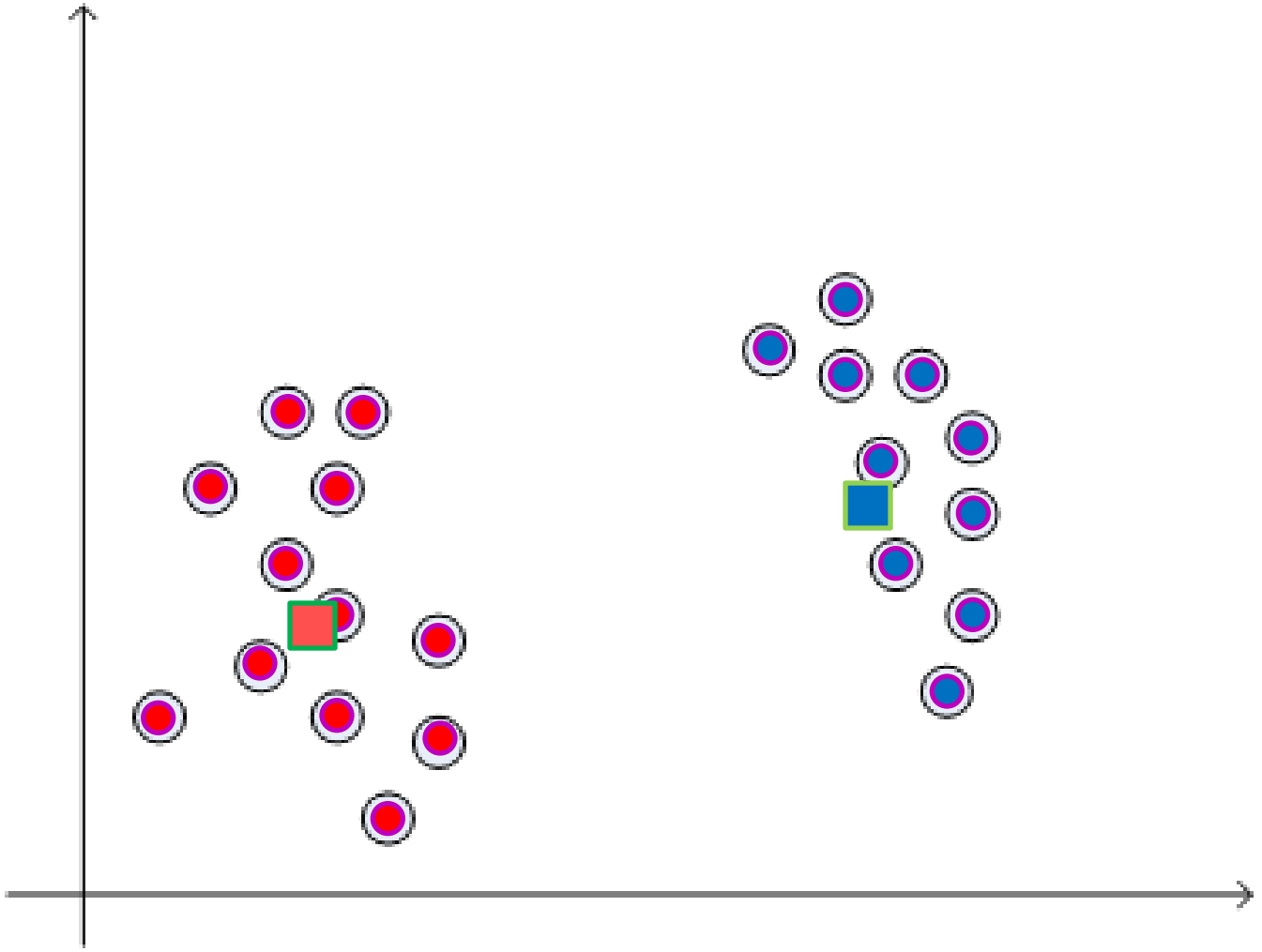
K-Means algoritam



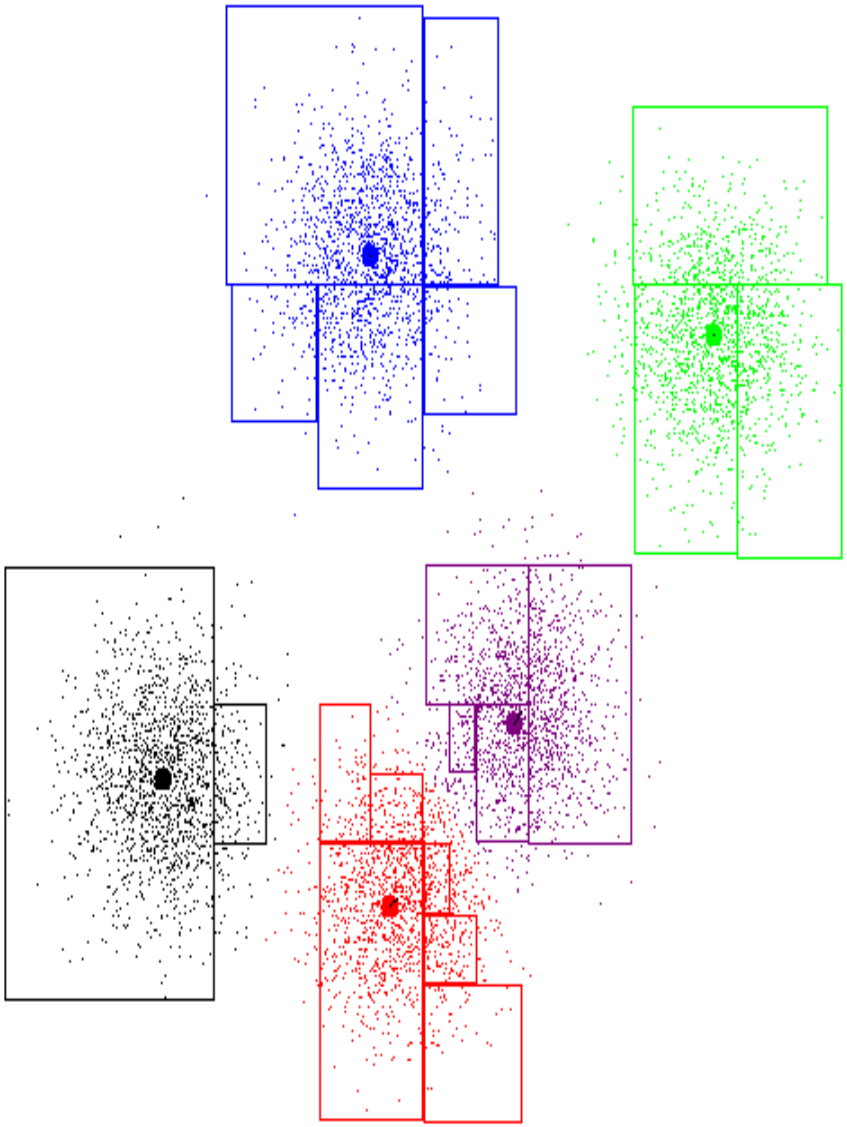
K-Means algoritam



K-Means algoritam



K-Means Primer 2



K-Means kao optimizacioni problem

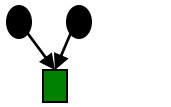
- Pogledajmo ukupan zbir rastojanja tačaka do centara:

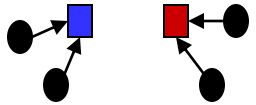
$$\phi(\{x_i\},\{a_i\},\{c_k\}) = \sum_i \text{dist}(x_i, c_{a_i})$$

tačke

dodele

centri





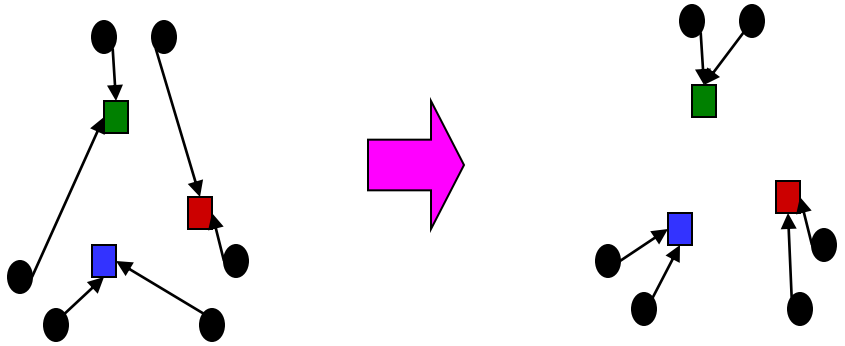
- Svaka iteracija smanjuje fukciju ϕ

- Dve faze u svakoj iteraciji:
  - Dodela klasterima: fiksiramo centre **c**, menjamo dodele **a**
  - Promena centara: fiksiramo **a**, menjamo centre **c**

Faza I: Dodela Klasterima

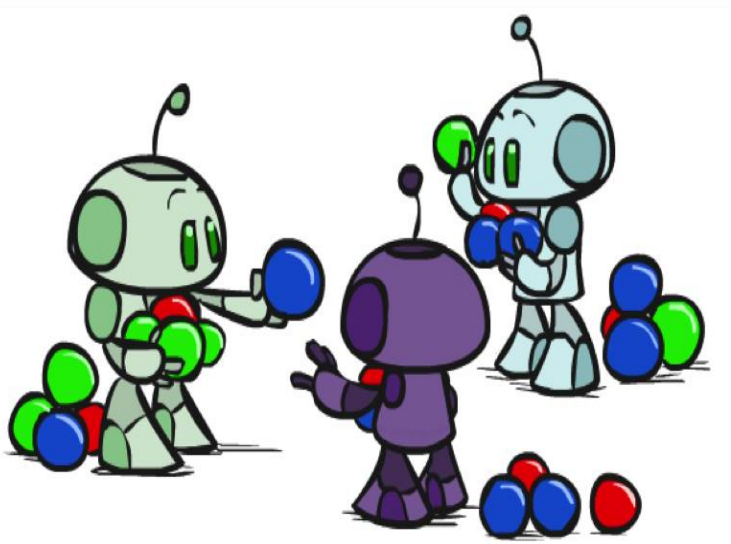
- Dodaj svaku tačku centru koji joj je najbliži:

$$a_i = \underset{k}{\operatorname{argmin}}\, \mathsf{dist}(x_i, c_k)$$



- Ova faza može samo da smanji fukciju ϕ!

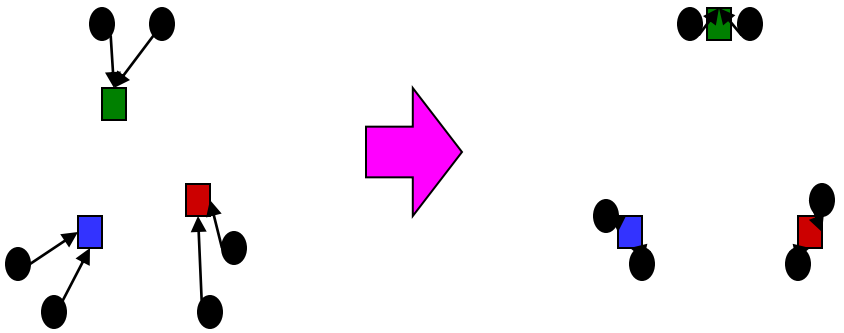
$$\phi(\{x_i\}, \{a_i\}, \{c_k\}) = \sum_i \mathsf{dist}(x_i, c_{a_i})$$



Faza II: Promena Centara

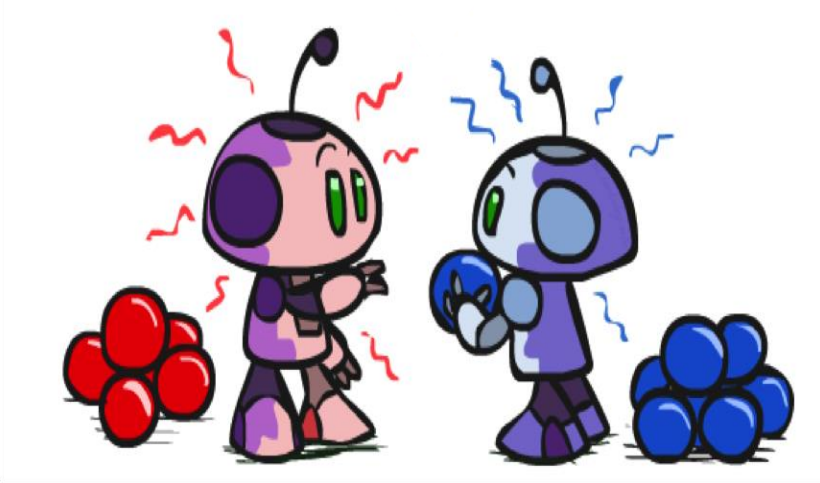
- Pomeramo svaki centar ka proseku tačaka koje su mu dodeljene:

$$c_k = \frac{1}{|\{i : a_i = k\}|} \sum_{i:a_i=k} x_i$$



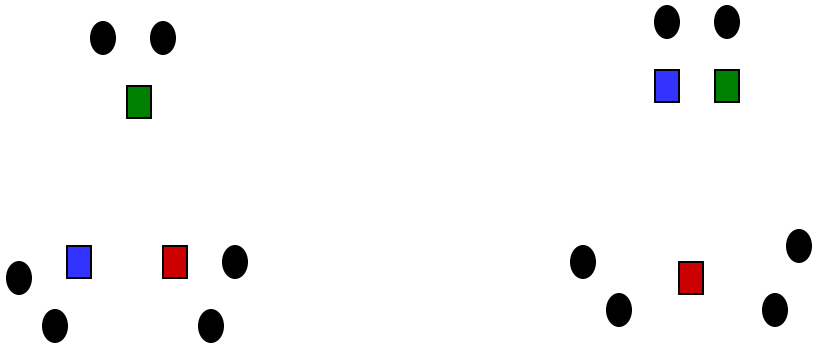
- Takođe samo smanjuje fukciju ϕ.

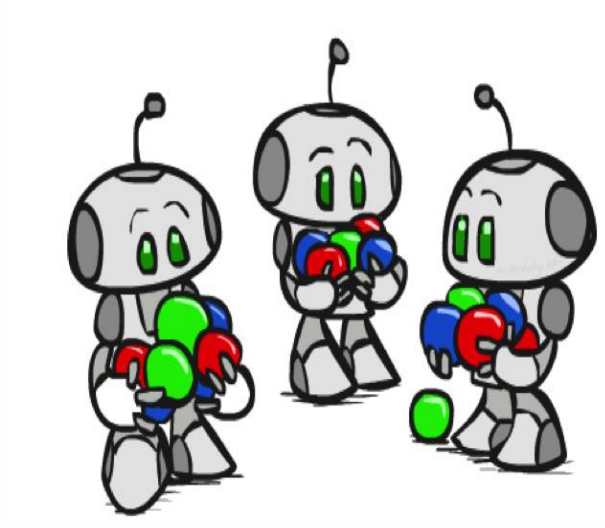
- Uzećemo bez dokaza: tačka koja ima najmanju kvadratnu euklidsku udaljenost ka tačkama {x} u nekom skupu je baš centar tih tačaka.



Inicijalizacija

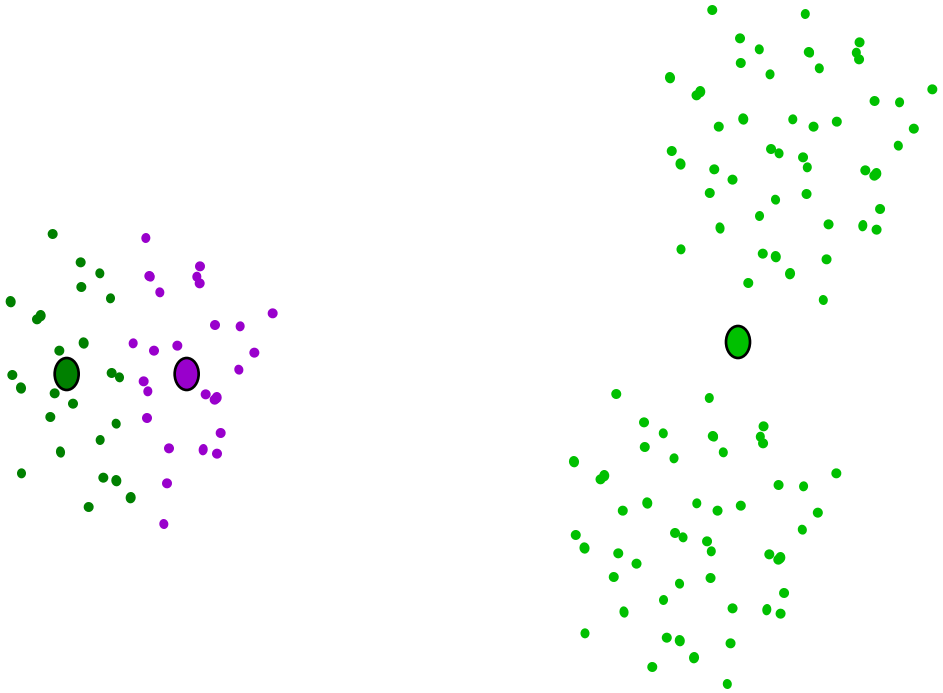
- K-means ne daje uvek isti rezultat za više pokretanja
  - Zahteva inicijalne centre
  - Vrlo je značajno kako su odabrani!
  - Postoji puno metoda za rešavanje ovog problema. Jedan od njih ćemo raditi danas.



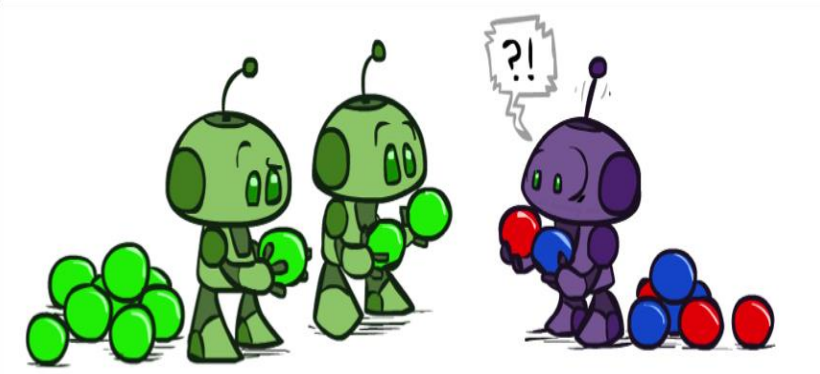


K-Means može da se zaglavi

- Lokalni optimum:

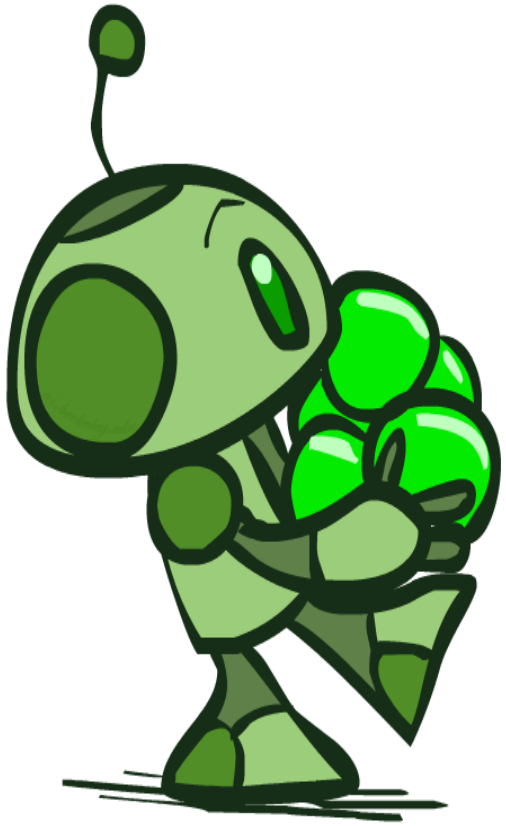


Šta je problem kod ovog primera?



K-Means Pitanja

- Da li konvergira?
  - Ka globalnom optimumu?
- Da li će uvek pronaći stvarne šablone koji postoje u podacima?
  - Samo ako su ti šabloni stvarno jasni?
- Da li će uvek naći nešto interesantno?
- Da li se stvarno koristi?
- Koliko klastera odabrati?

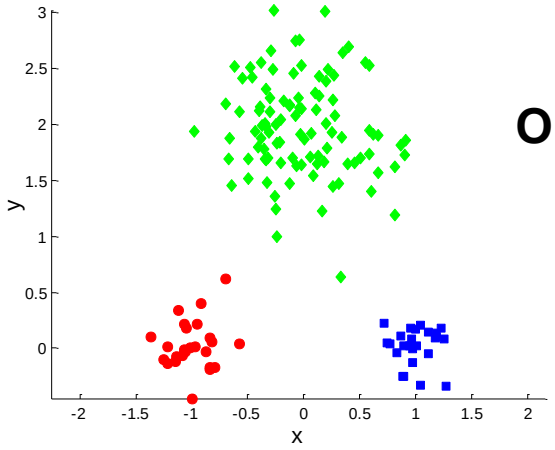


Кластеровање K-средина – Детаљи

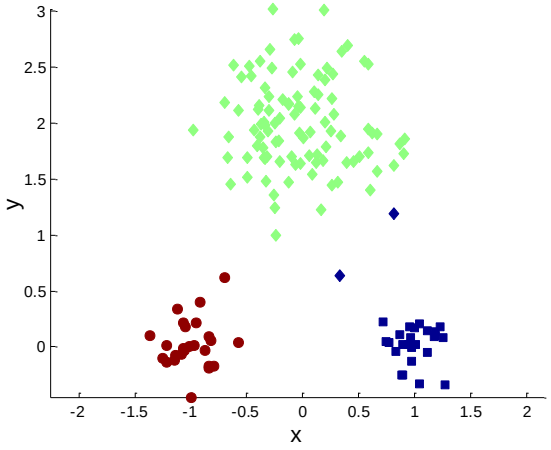
- Иницијални центроиди се често случајно бирају.
  - Добијају се различити кластери за различите случајне секвенце.
- Центроид је (обично) средња вредност тачака из кластера.
- ‘Близина’ се мери Еуклидским растојањем, косинусном сличношћу, корелацијом, итд.

- K-средине конвергирају за уобичајене (поменуте) мере сличности.
- Најбржа је конвергенција у првих неколико итерација.
  - Критеријум заустављања у пракси је најчешће ‘док релативно мало података мења кластер’

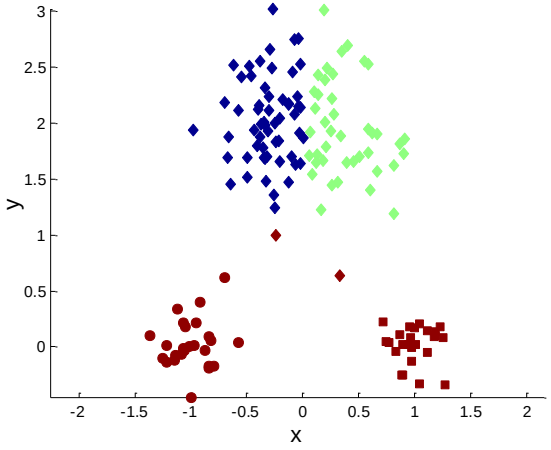
Два кластеринга K-средина



Оригиналне тачке

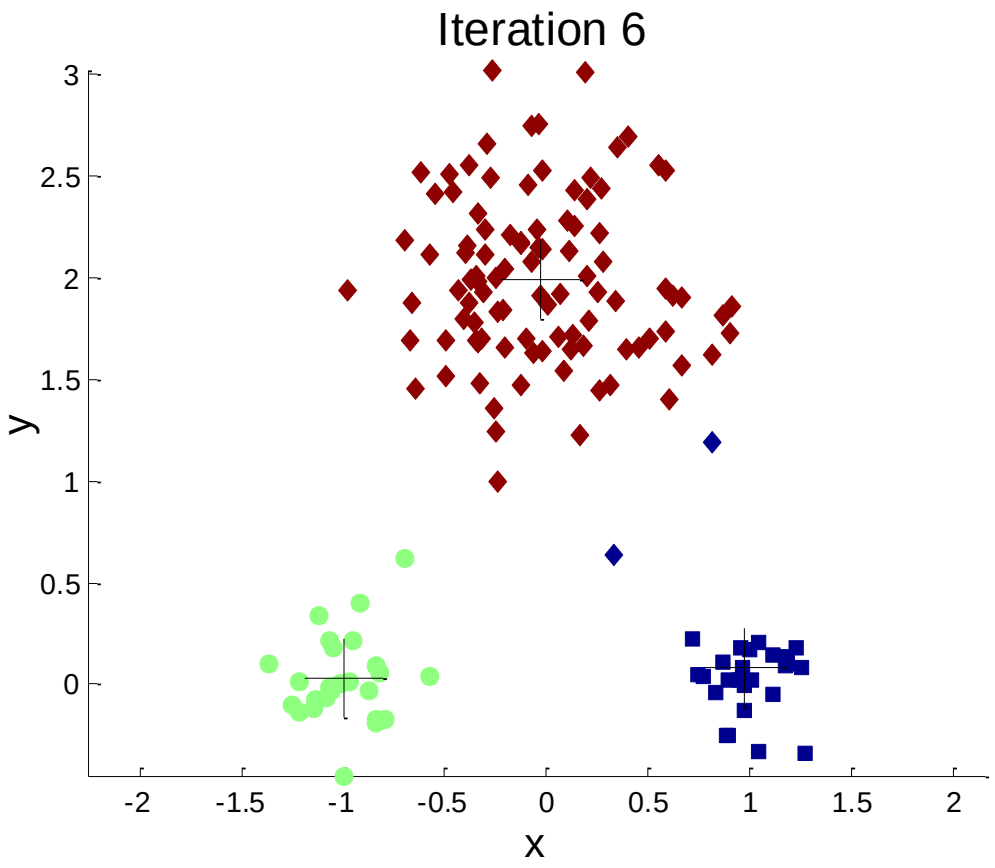


Оптимални кластеринг

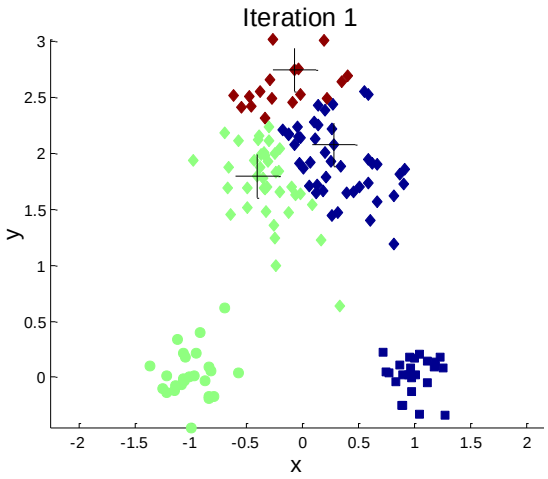


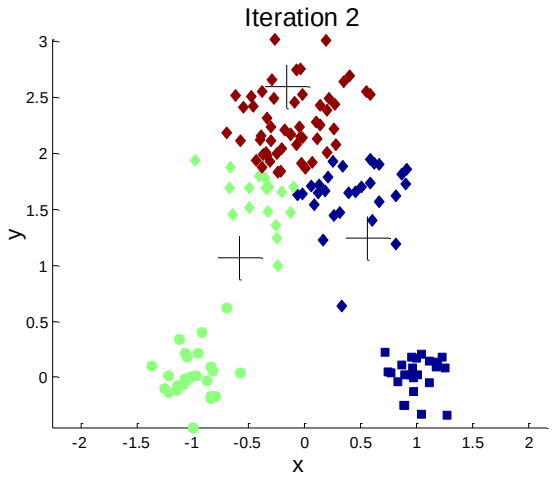
Субоптимални кластеринг

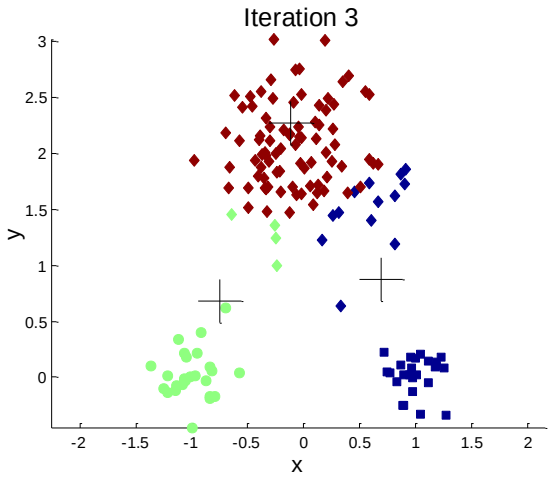
Важност избора иницијалних центроида

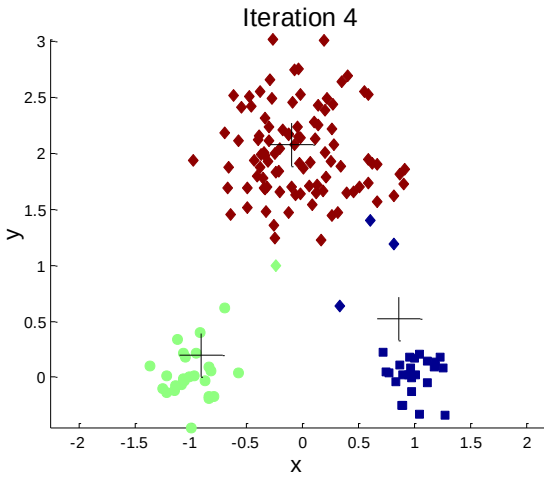


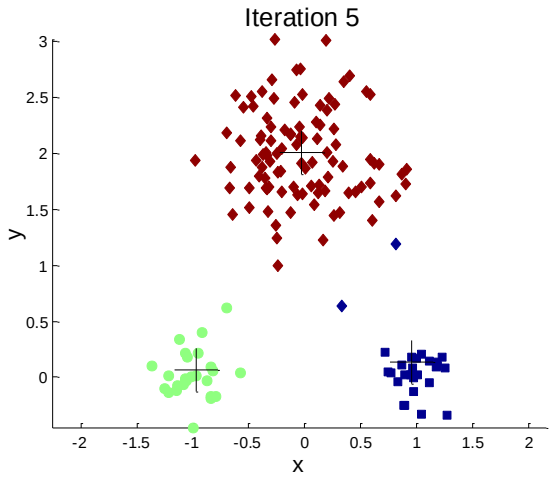
Важност избора иницијалних центроида

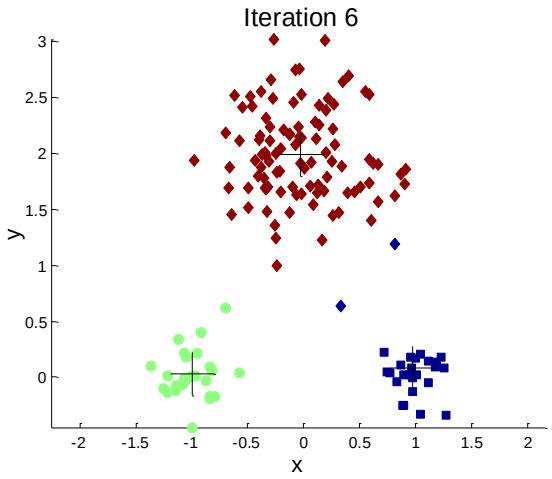




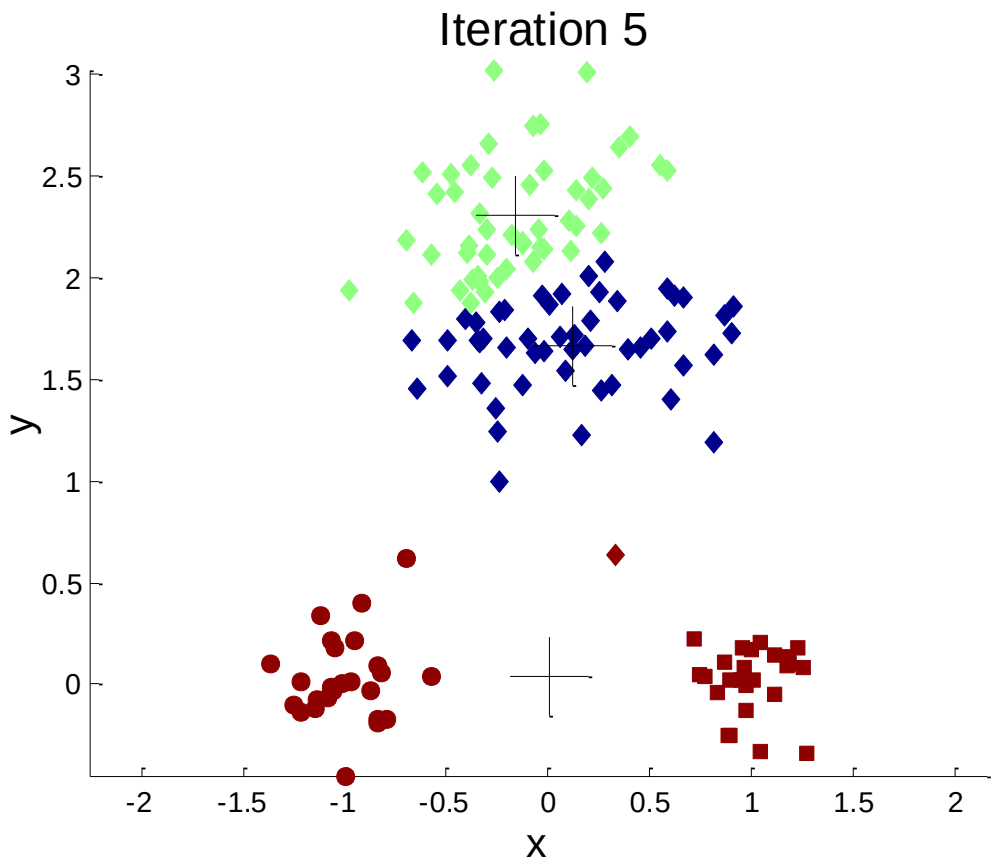




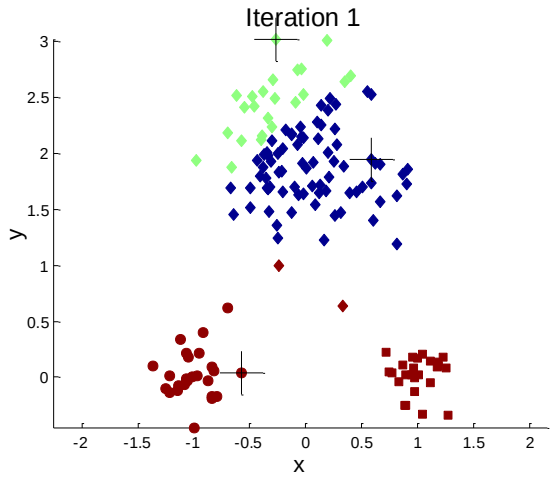


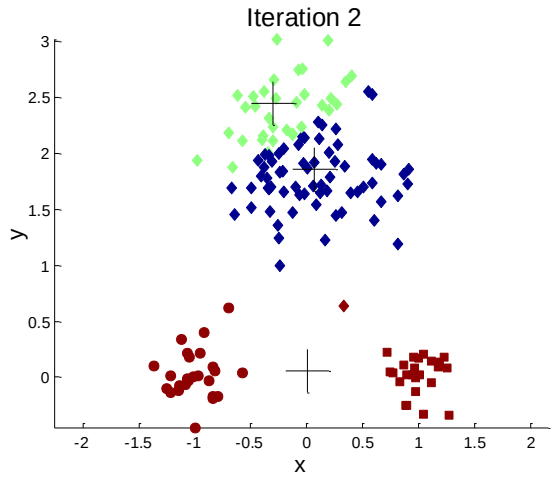


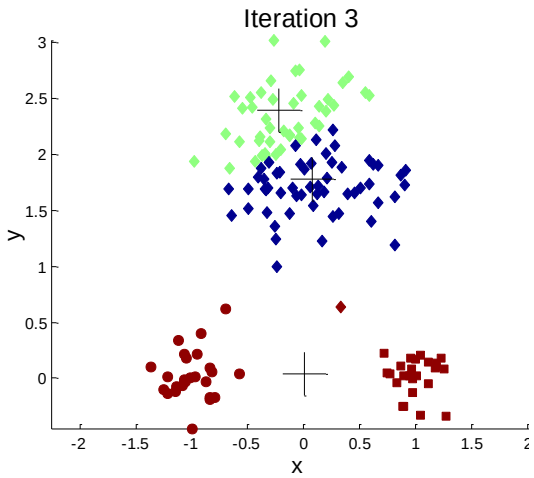
Важност избора иницијалних центроида ...

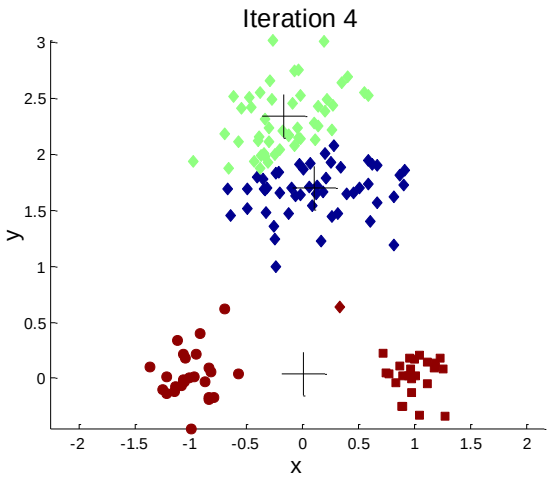


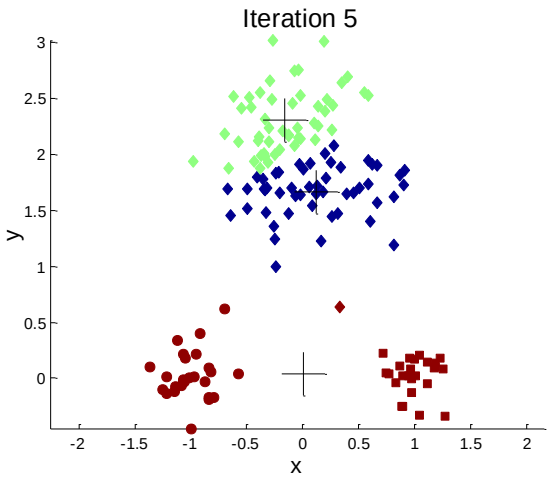
Важност избора иницијалних центроида...











Проблеми при избору иницијалних тачака

- Ако постоји K ‘стварних’ кластера, шанса да се изабере један центроид за сваки кластер је мала.
  - Шанса је релтивно мала када је K велико
  - Ако су кластери исте димензије, n, тада је шанса

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

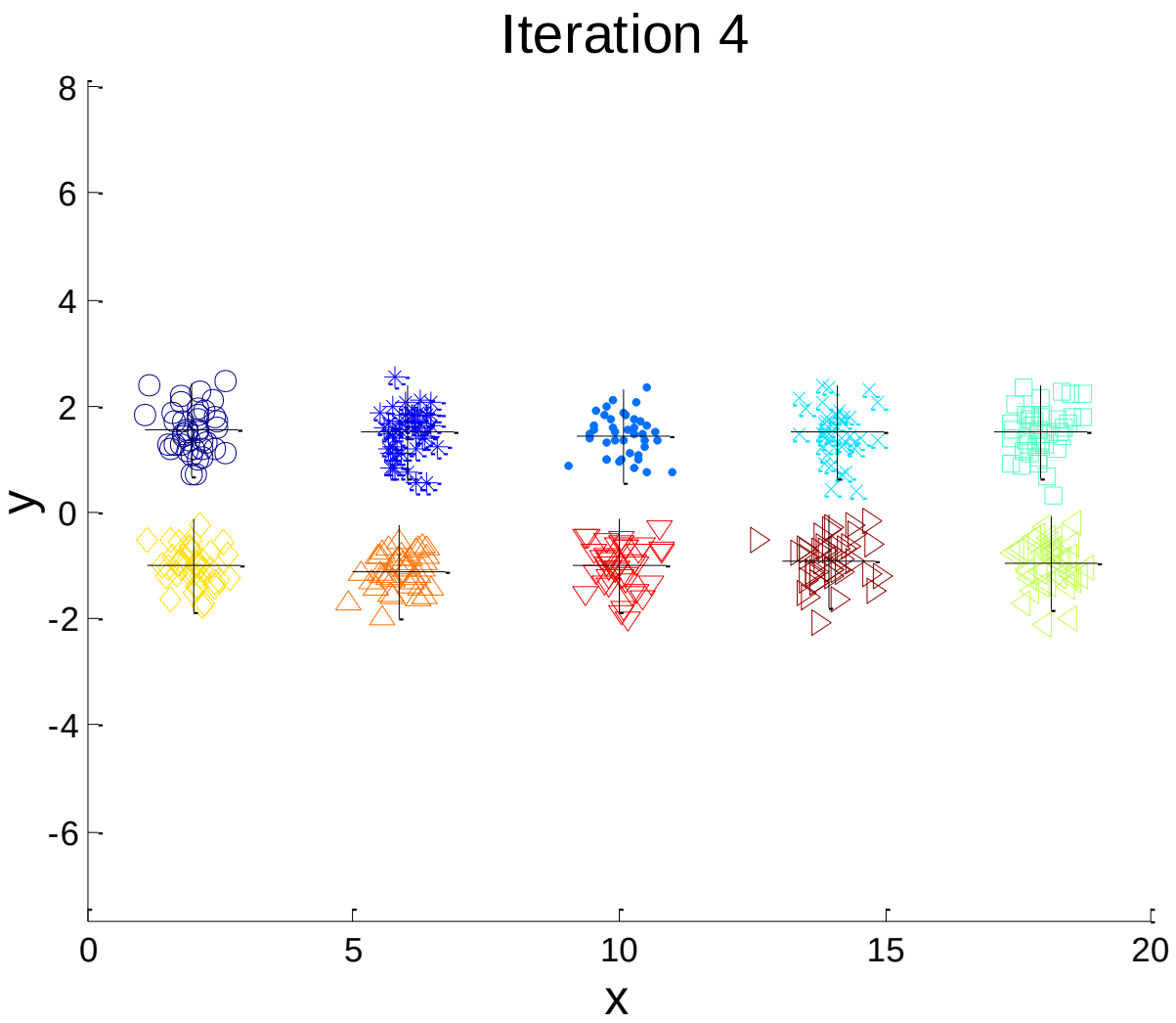
бира се кластер

од *n* тачака у кластеру, бира се тачка која ће бити цетроид

$$= \frac{K*n*(K-1)*n*(K-2)*n\ldots.*2*n*1*n}{K*n*K*n*K*n\ldots.*K*n*K*n} =$$

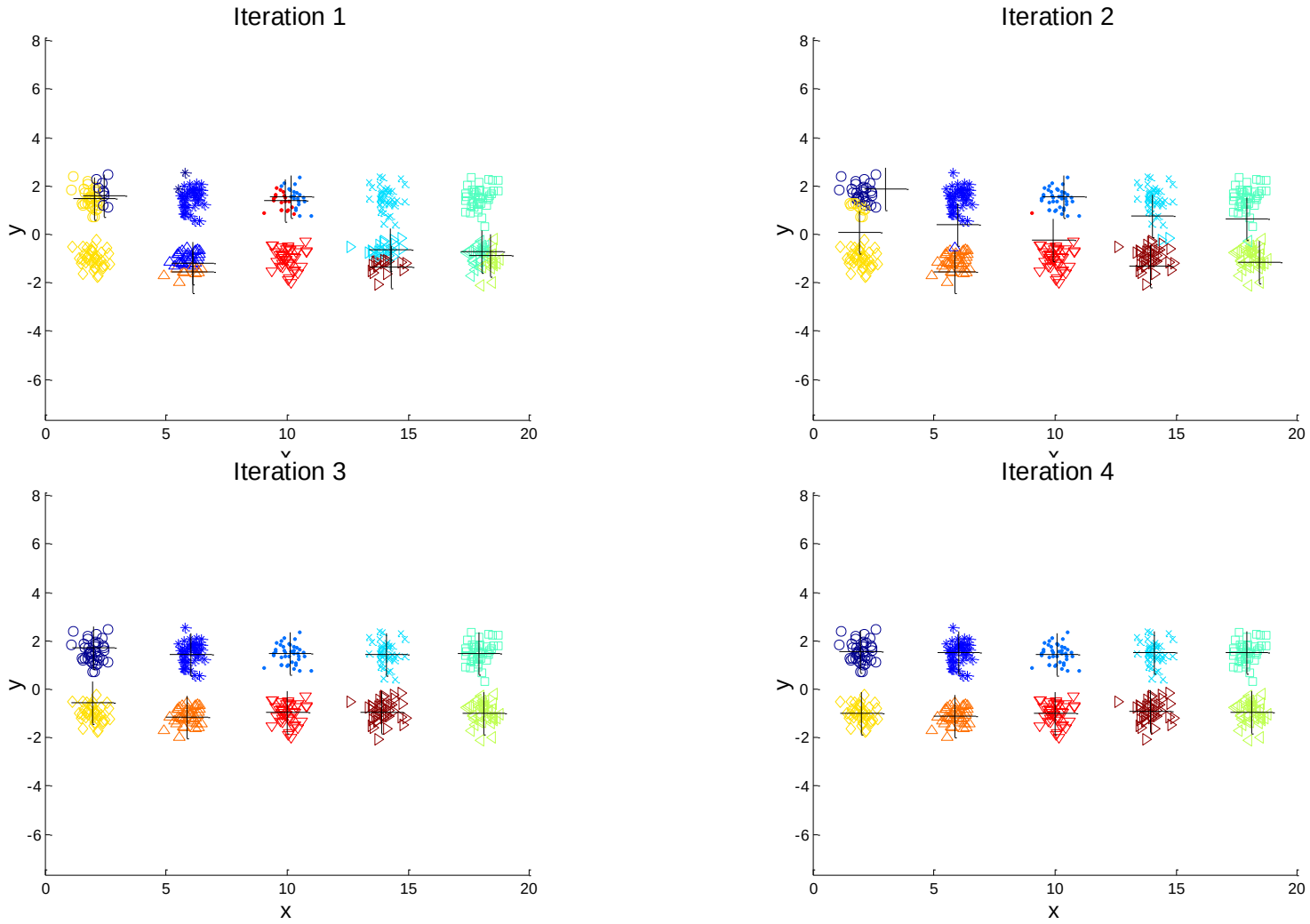
  - На пример, за K = 10, вероватноћа = $10!/10^{10} = 0.00036$
  - У неким случајевима центроиди ће се модификовати на ‘добар’ начин, а у некима баш и неће
  - Посмтраћемо пет парова кластера

Пример 10 кластера



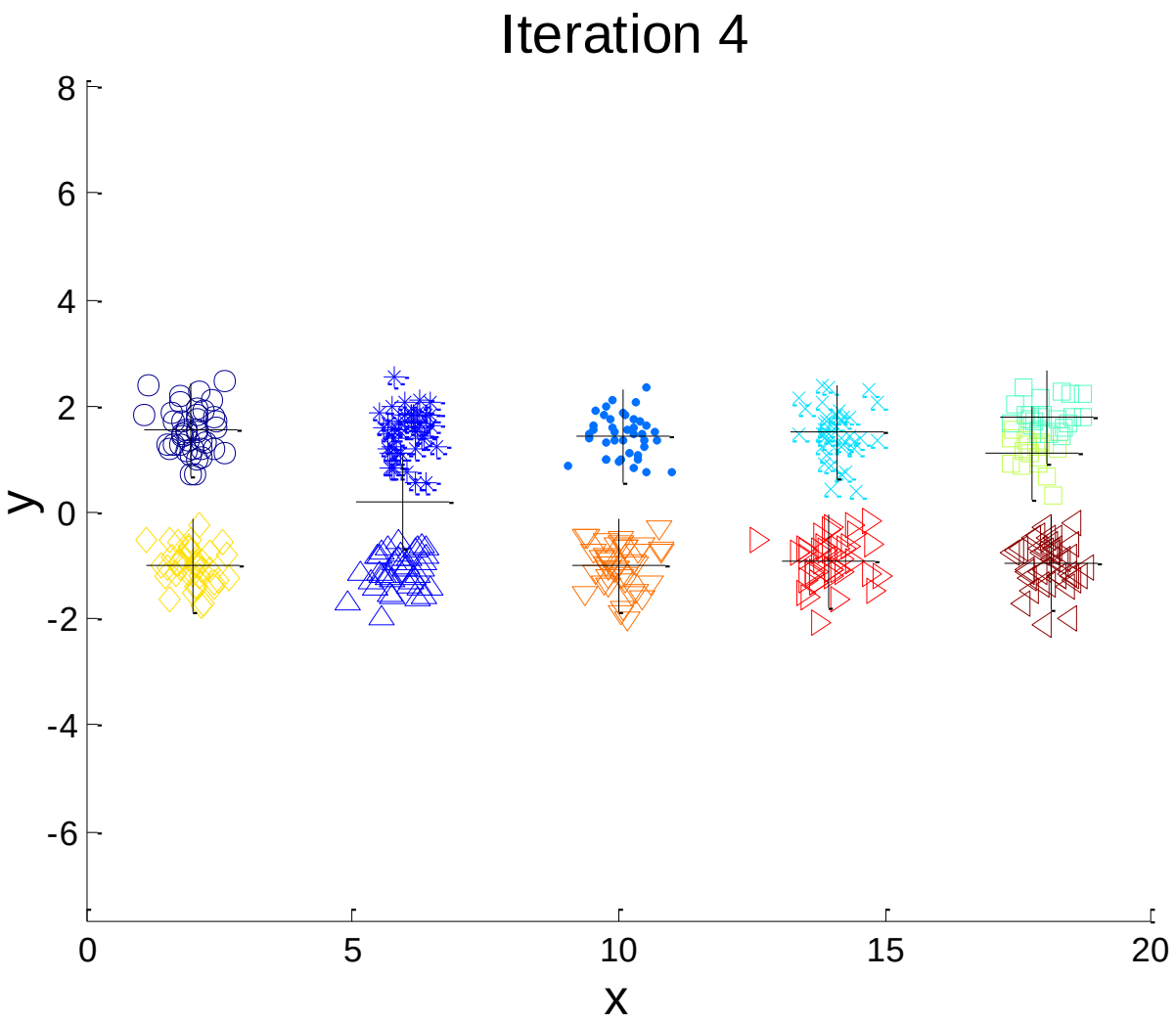
Почиње се са два иницијална центроида у једном кластеру сваког пара кластера

Пример 10 кластера



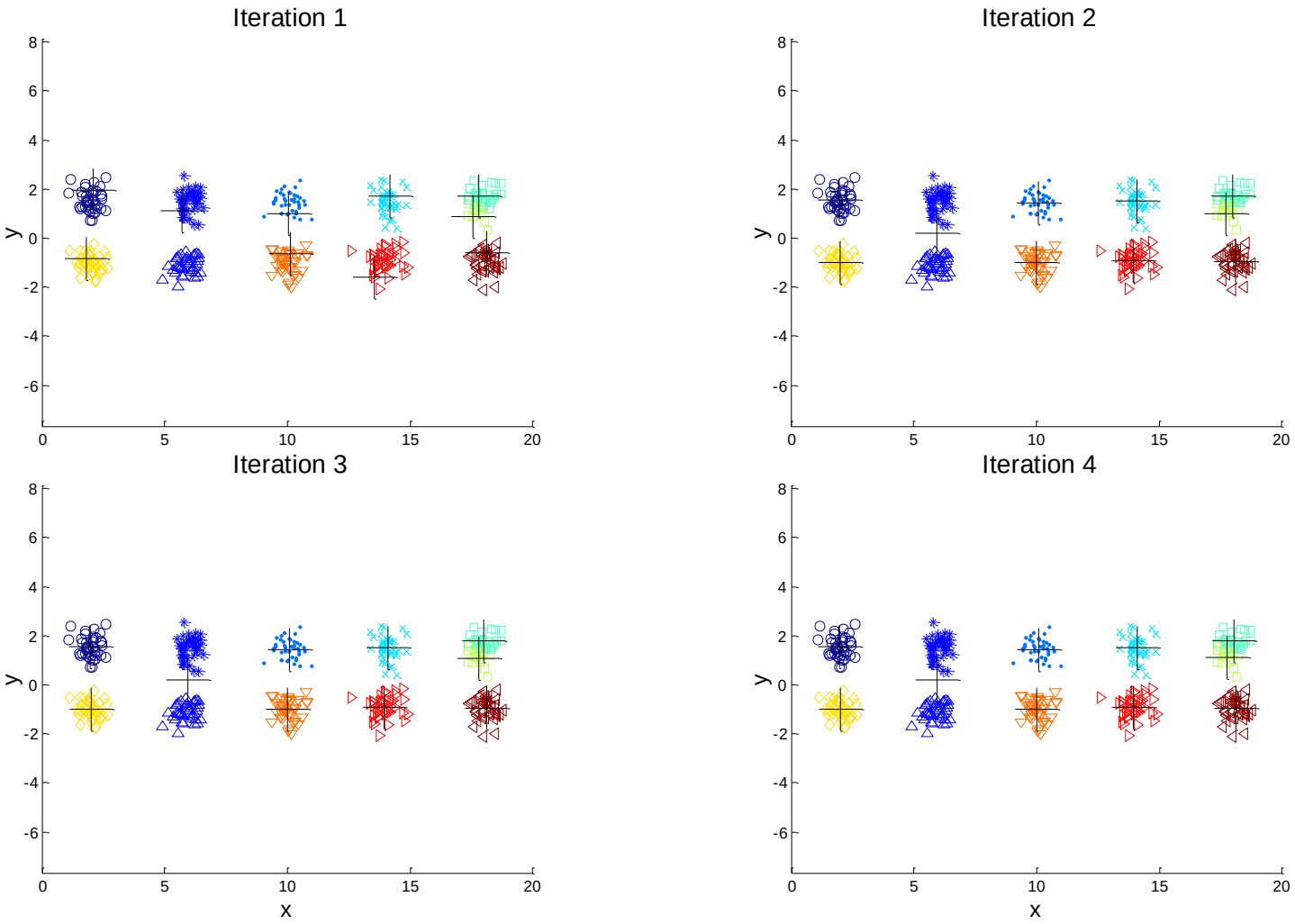
Почиње се са два иницијална центроида у једном кластеру сваког пара кластера

Пример 10 кластера



Почиње се са неколико парова кластера који имају три иницијална центроида, док остали имају по један.

Пример 10 кластера



Почиње се са неколико парова кластера који имају три иницијална центроида, док остали имају по један.

Решење проблема иницијалних центроида

- Вишеструка извршавања
  - Помаже, али вероватноћа није на вашој страни
- Коришћење хијерархијског кластеринга за одређивање иницијалних центроида
- Генерисање више од *к* иницијалних центроида и затим избор међу тим центроидима
  - Бирају се они који су најбоље раздвојени
- Пост-процесинг (спајање или разбијање добијених кластера)
- Бисекција K-средина (биће приказан на неком од наредних курсева)
  - Није јако осетљиво на питања иницијализације

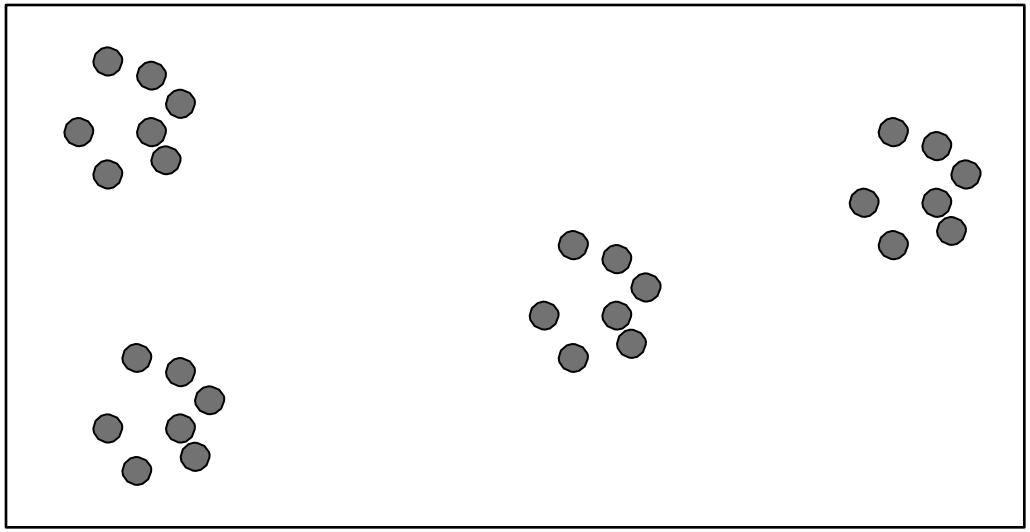
K-means++ [Arthur et al. ’07]

- Предлог решења проблема иницијализације центара
- Идеја алгоритма: раширити центре што више
- Алгоритам:
- Одабрати први центар, c_1, на случајан начин из униформне расподеле целог скупа података
- Понављати за $2 \leq i \leq k$: (к је број кластера)
  - Одабрати c_i тако да буде тачка из података x_0 бирана из дистрибуције: $\frac{D_i}{\sum_j D_j}$

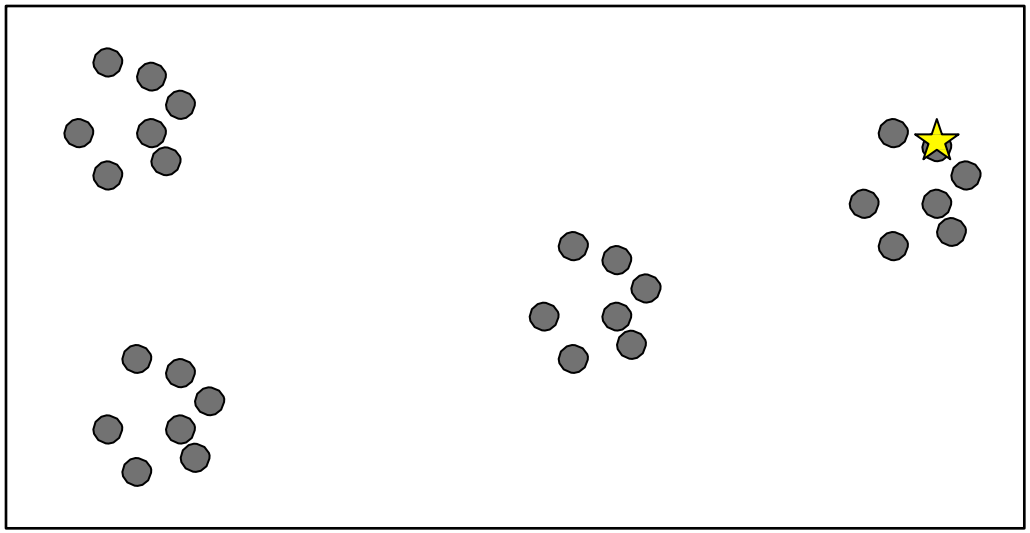
$$D_i = \min(\|x_i - c_1\|^2, \|x_i - c_2\|^2, ..., \|x_i - c_n\|^2)$$

- Идеја је да се следећи центар бира тако да тачке које су удаљеније од већ одабраних центара имају већу вероватноћу

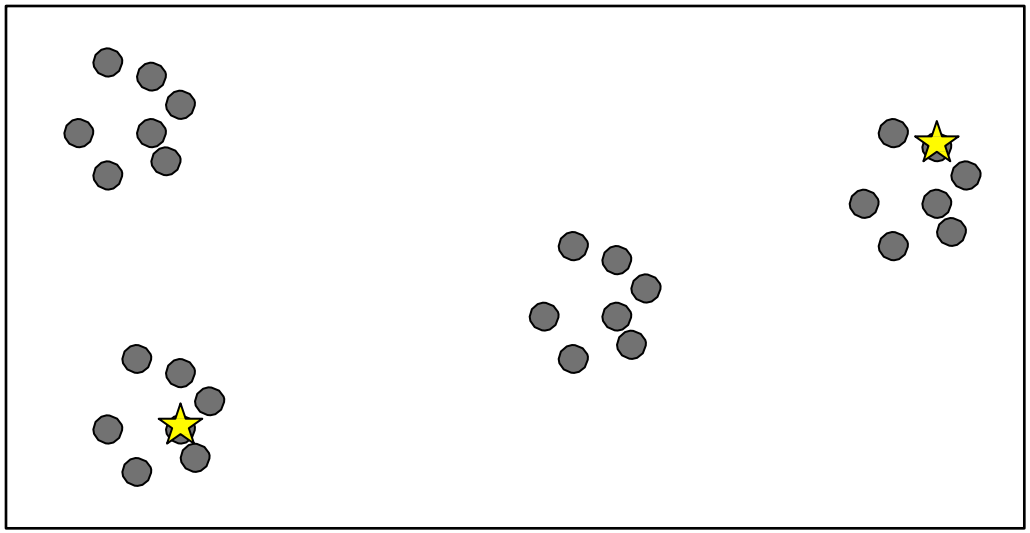
K-means++ Пример



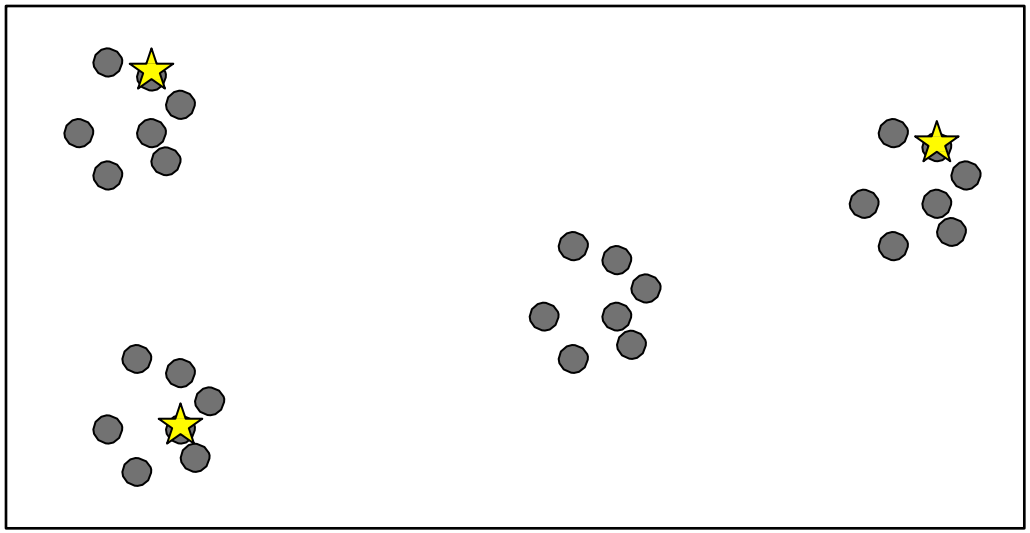
K-means++ Пример



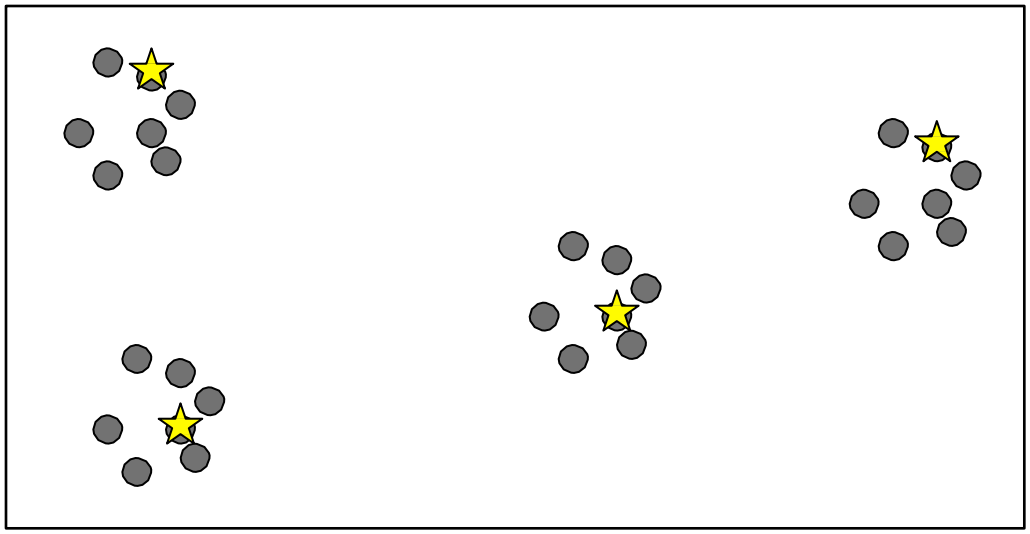
K-means++ Пример



K-means++ Пример



K-means++ Пример



Проблеми K-means++

- K пролаза кроз податке
- За велике колекције података обично је и K велико па је алгоритам јако спор.
- Предлог убрзања: “Scalable k-means++”, Bahmani et al. 2012

Оцена кластера K-средина

- Најчешћа мера је сума квадрата грешака (Sum of Squared Error - SSE)
  - За сваку тачку, грешка је растојање до најближег кластера
  - SSE се добија сабирањем кавдрата ових грешака.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

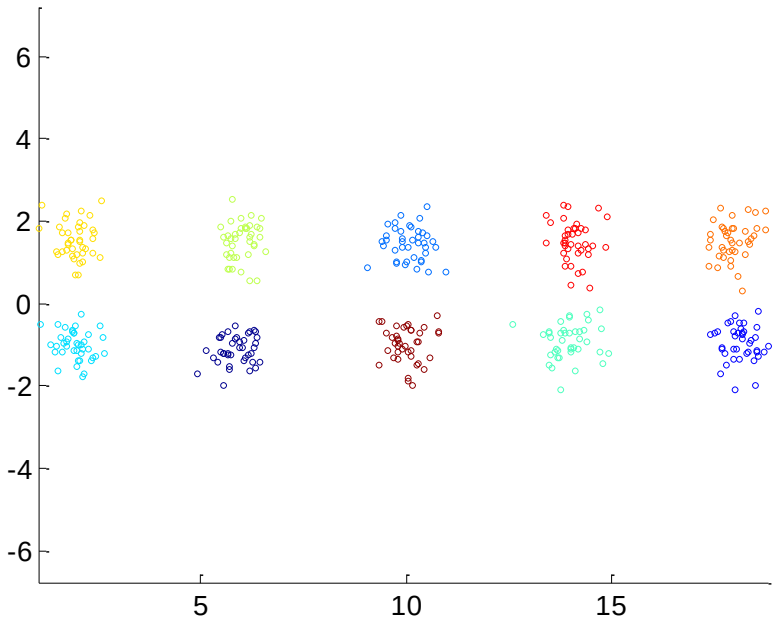
  - x је податак (тачка) из кластера C_i а m_i је репрезентативна тачка за кластер C_i
    - m_i одговара центру (средини) кластера
  - Од два дата кластеринга можемо одабрати онај са мањом грешком
  - Један једноставан начин за смањење SSE је повећање K, броја кластера
    - Али, добро кластеровање са мањим K може да има нижу SSE него лоше кластеровање са већим K

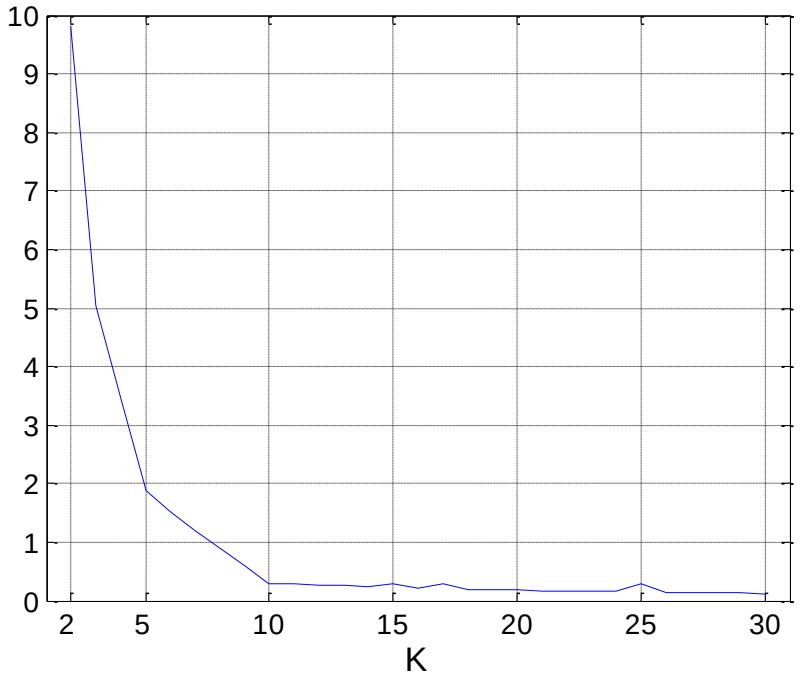
Одабир броја кластера

- Није лако унапред знати у колико кластера треба кластеровати скуп података
- Код 2д скупа можемо визуализовати податке и видети природне групе
- Код више-димензионих можемо пројектовати податке на 2д, али тиме потенцијално губимо информације
- Два алтернативна метода у наставку

Одабир броја кластера

- Оба метода су занована на “расутости (густини)” кластера
- Први метод
  - Тражимо нагли прелаз (“лакат”) у графику SSE по броју кластера





Одабир броја кластера

- Други метод је статистички:
  - Gap statistic: Tibshirani, Walther & Hastie (2000)

- Заснива се на разлици (*Gap*) дисперзије кластера добијених помоћу К-средина за дати скуп података и дисперзије кластера случајно генерисаних скупова података

- Разлика се мери итеративно од неког датог броја кластера

- Број кластера који произведе највећи размак је предлог за број кластера за К-средина

Одабир броја кластера

- Заснива се на разлици (Gap) дисперзије кластера добијених помоћу К-средина за дати скуп података и дисперзије кластера случајно генерисаних скупова података.
- Идеја је у томе да наши подаци имају природне групе тј. да нису скроз случајно генерисани.

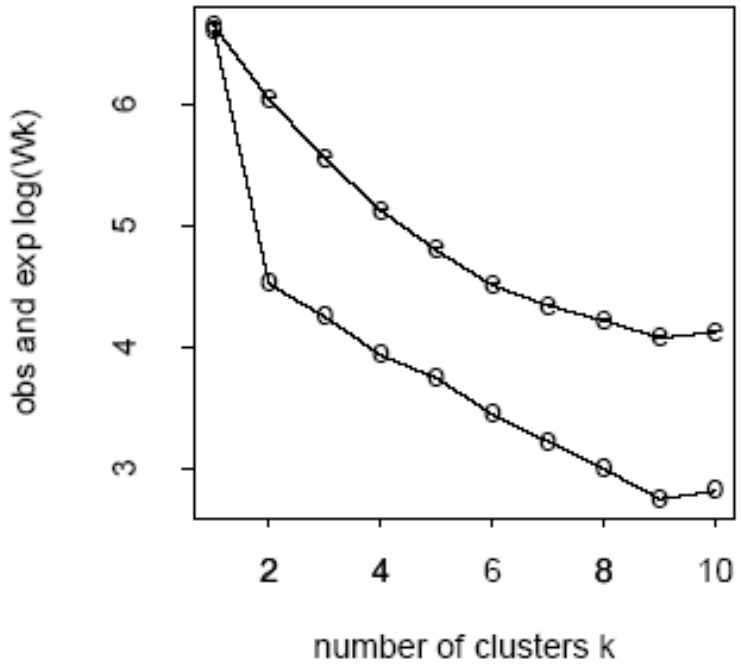
- Постављамо питање колико има тих група тј. у колико кластера треба да кластерујемо?
- Идеја је да ће дисперзија (расутост) података око центара кластера бити мала кад пронађемо баш тај природан број група.
- Како ћемо знати шта је мала дисперзија?
- Тако што ћемо видети колика је дисперзија случајно генерисаних података (оних који немају природне групе) и онда је упоредити са оном коју смо добили.
- К за које је разлика дисперзија у односу на случајно генерисане податке највећа је оно које бирамо.

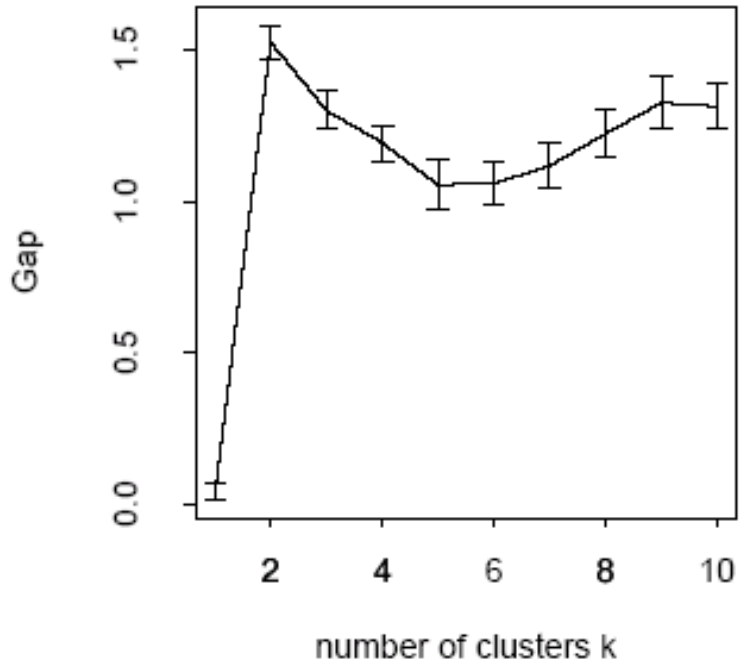
Одабир броја кластера

- Заснива се на разлици (Gap) дисперзије кластера добијених помоћу К-средина за дати скуп података и дисперзије кластера случајно генерисаних скупова података

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \quad D_r = 2n_r \sum_{i \in C_r} \|x_i - \bar{x}\|^2 \leftarrow \text{SSE}$$

$$\max Gap_n(k) = E_n^*(\log(W(k))) - \log(W(k))$$





Одабир броја кластера

- Заснива се на разлици (Gap) дисперзије кластера добијених помоћу К-средина за дати скуп података и дисперзије кластера случајно генерисаних скупова података

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \qquad D_r = 2n_r \sum_{i \in C_r} \left\| x_i - \bar{x} \right\|^2 \leftarrow \text{SSE}$$

$$\max Gap_n(k) = E_n^*(\log(W(k))) - \log(W(k))$$

Ово је дисперзија података коју очекујемо за случајно генерисане податке.

Добијамо је вишеструким генерисањем случајних скупова података и одређивањем дисперзије за сваки.

Те дисперзије се онда упросече.

Одабир броја кластера

- Заснива се на разлици (Gap) дисперзије кластера добијених помоћу К-средина за дати скуп података и дисперзије кластера случајно генерисаних скупова података

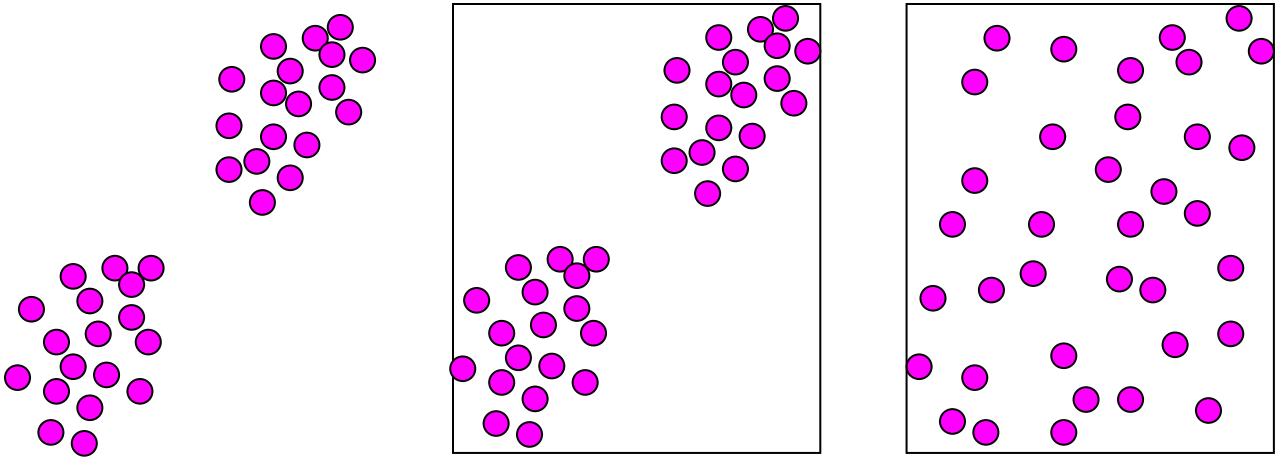
$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \qquad D_r = 2n_r \sum_{i \in C_r} \left\| x_i - \bar{x} \right\|^2 \leftarrow \text{SSE}$$

$$\max Gap_n(k) = E_n^*(\log(W(k))) - \log(W(k))$$

Ово је дисперзија добијена за к кластера помоћу К-средина.

Број кластера к код којега је ова разлика највећа је онај који најбоље групише тачке, тачније онај који је успео да пронађе природно груписање нашег скупа података, ако оно постоји.

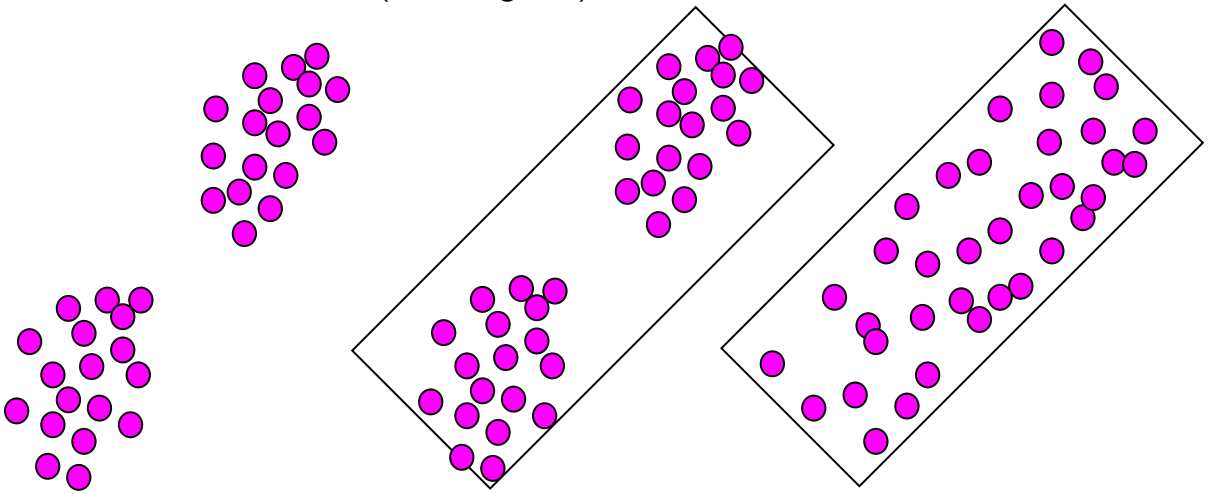
Како добијамо случајно генерисане скупове података?



Подаци

Ограничавајући
Правоугаоник
(*Bounding Box*)

Monte Carlo
симулације



Подаци

Ограничавајући
Правоугаоник
(*Bounding Box*)

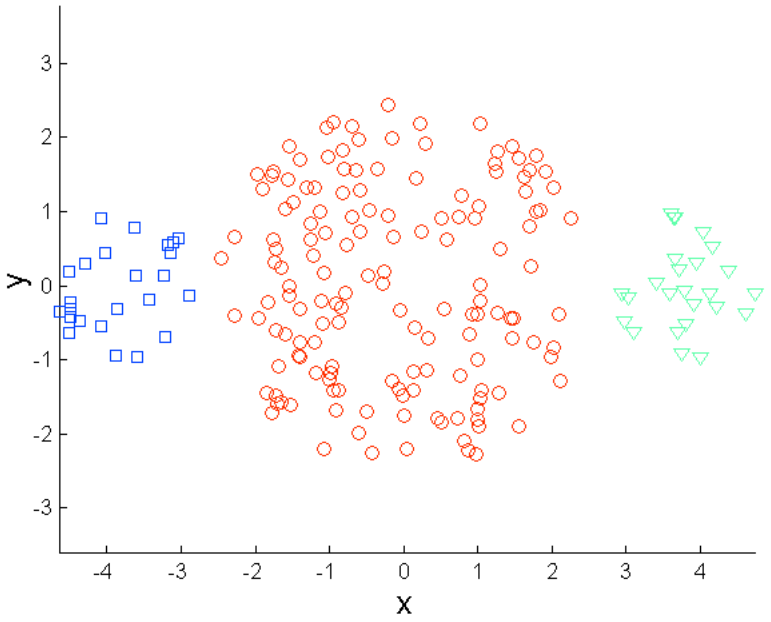
Monte Carlo
симулације

Ограничења K-средина

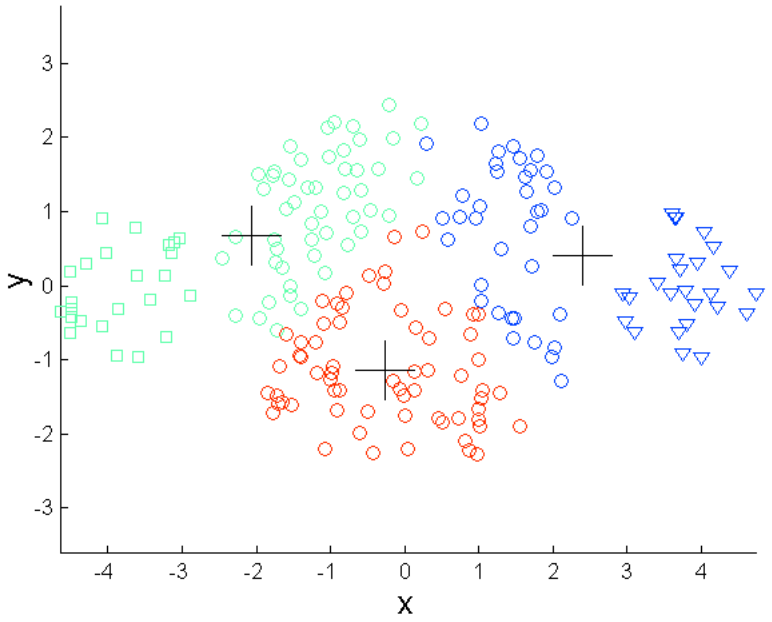
- K-средина има проблеме када се разликују кластери
  - величина
  - густина
  - несферични облици

- K-средина има проблем у случају присуства страних података.

Ограничења K-средина: Различите величине кластера

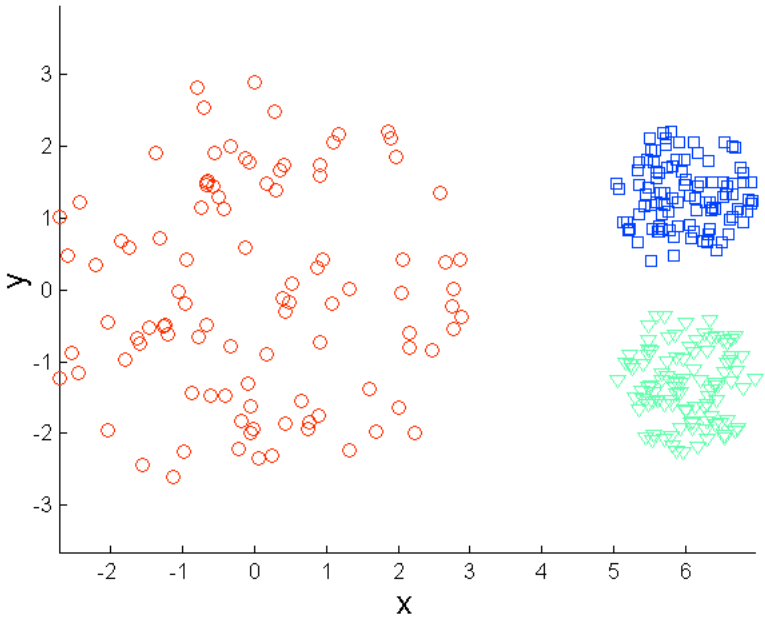


Оригиналне тачке

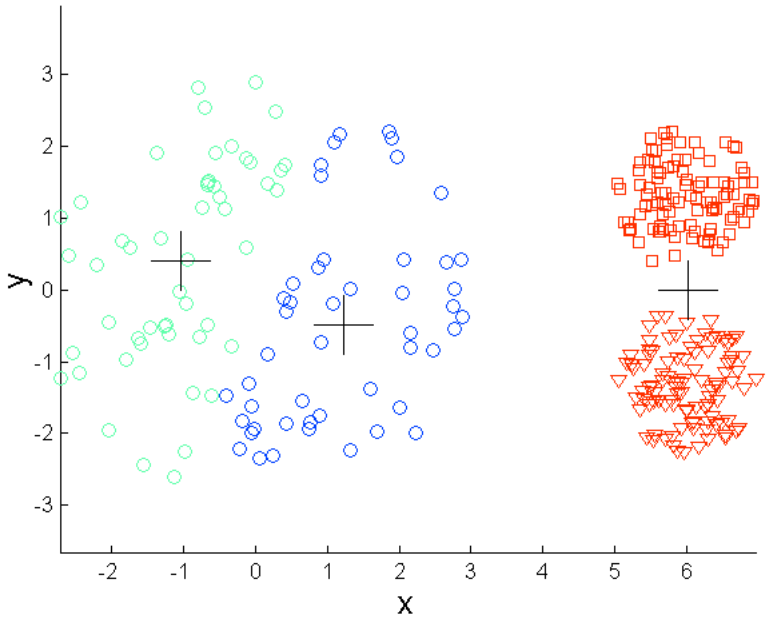


K-средине (3 кластера)

Ограничења K-средина: Различите густине

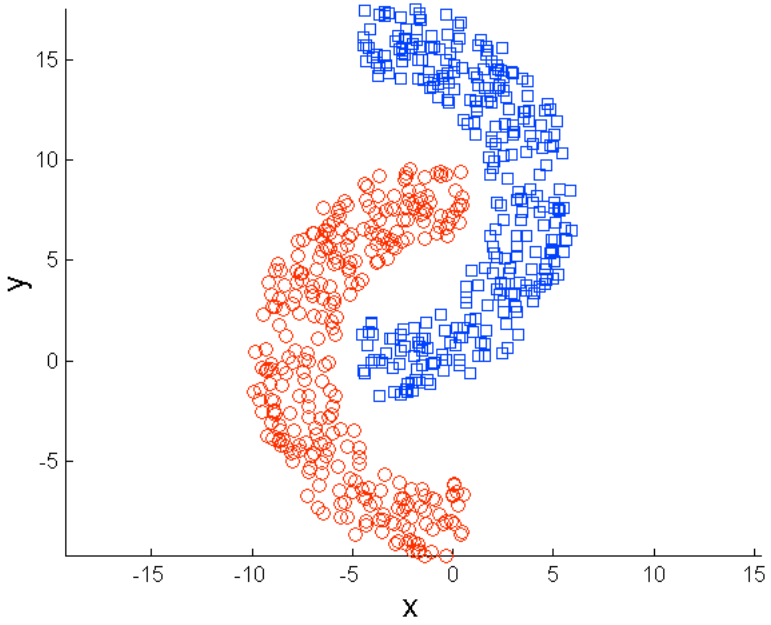


Оригиналне тачке

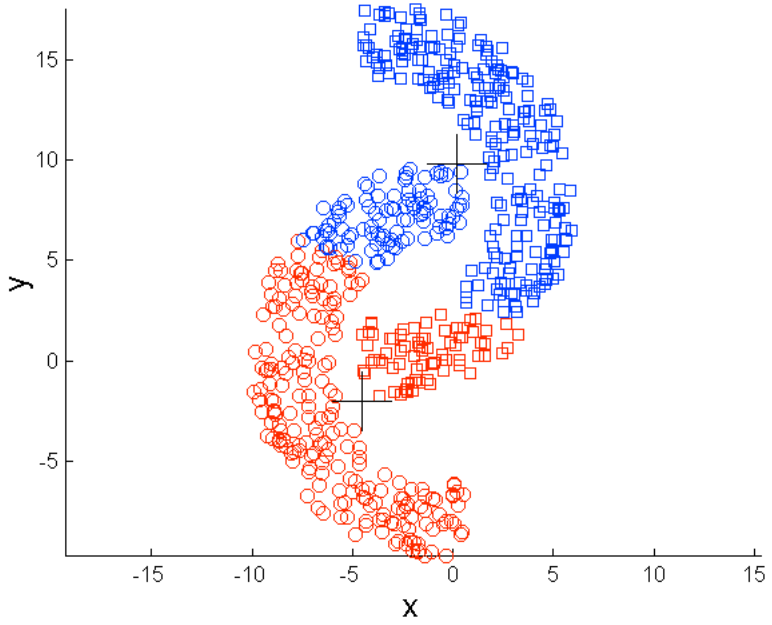


K-средине (3 кластера)

Ограничења K-средина: Несферични облици



Оригиналне тачке

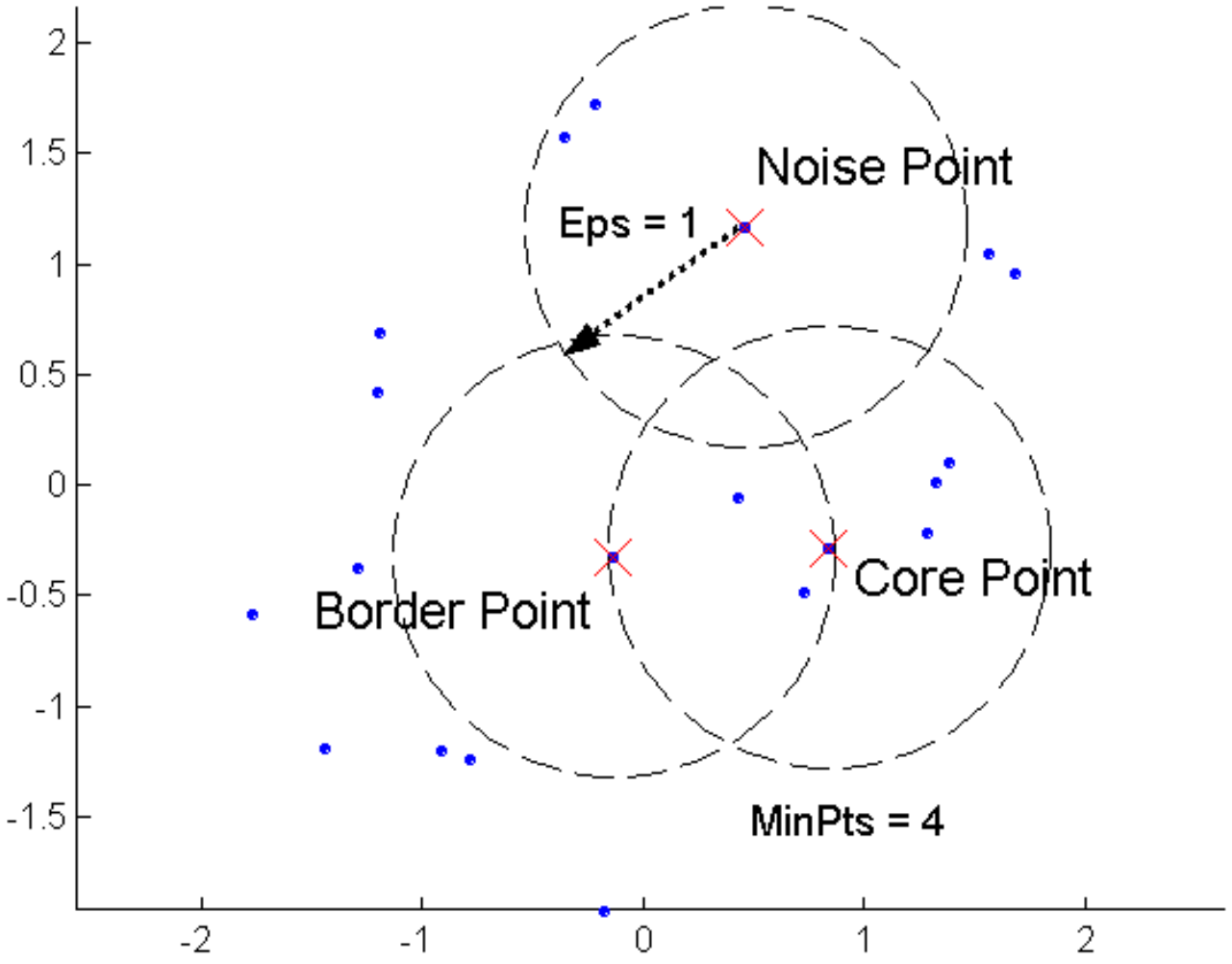


K-средине (2 кластера)

DBSCAN

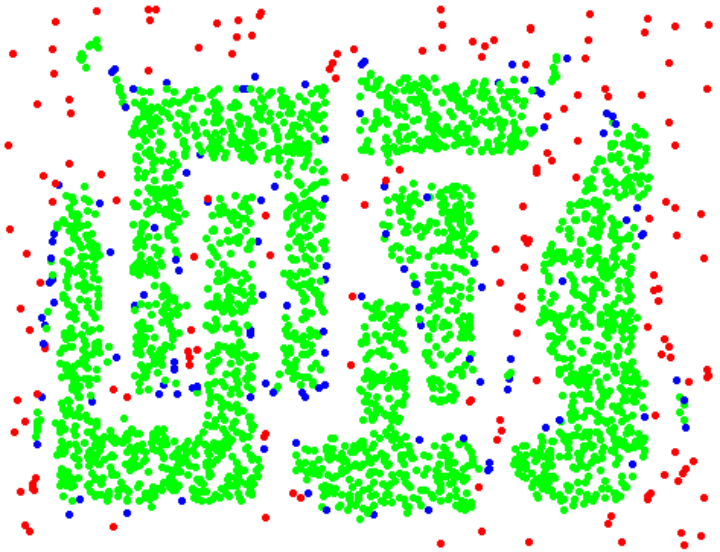
- DBSCAN је алгоритам базиран на густини.
  - Густина = број тачака унутар задатог пречника (Eps)
  - Тачка је тачка језгра (core point) ако има више од специфицираног броја тачака (MinPts) унутар Eps
    - То су тачке које се налазе унутар кластера
  - Ивична тачка (border point) има мање од MinPts тачака на растојању Eps, али је суседна са тачком језгра (налази се у Eps "кругу" неке тачке језгра)
  - Тачка шума (noise point) је свака тачка која није ни тачка језгра ни ивична тачка.

DBSCAN: тачка језгра, ивична тачка, тачка шума



DBSCAN: тачке језгра, ивичне тачке, тачке шума

Оригиналне тачке

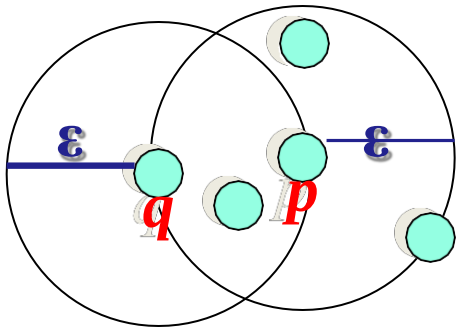


Типови тачака:
језгро, ивица и шум

Eps = 10, MinPts = 4

Dosežnost po gustini (*Density-reachability*)

- **Direktna dosežnost po gustini**
  - **Tačka *q* je direktno dosežna po gustini od tačke *p* ako je *p core* tačka, a *q* je u ε okolini p**

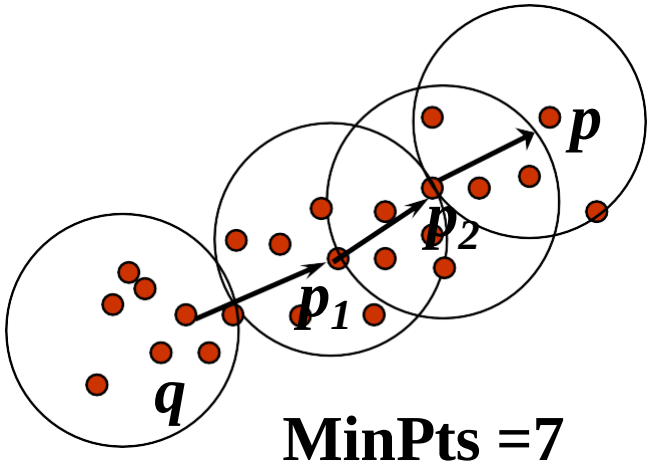


MinPts =4

- ***q je* direktno dosežna po gustini od *p***
- ***p* nije direktno dosežna po gustini od *q* zato što *q* nije *core* tačka**
- **Dosežnost po gustini nije simetrična**

Dosežnost po gustini

- **Dosežnost po gustini (direktna i indirektna):**
  - Tačka *p* je direktno dosežna po gustini od tačke p_2
  - p_2 je direktno dosežna po gustini od tačke p_1
  - p_1 je direktno dosežna po gustini od tačke *q*
  - $p \leftarrow p_2 \leftarrow p_1 \leftarrow q$ lanac direktne dosežnosti po gustini

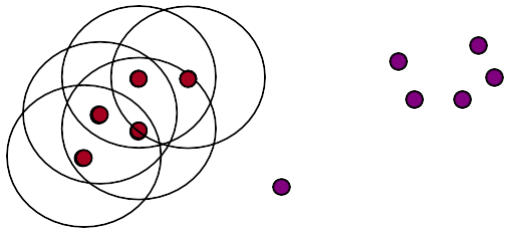


MinPts =7

- ***p* je indirektno dosežna po gustini od tačke *q***
- ***q* nije direktno dosežna po gustini od tačke *p* zato što *p* nije *core* tačka**

DBSCAN Primer

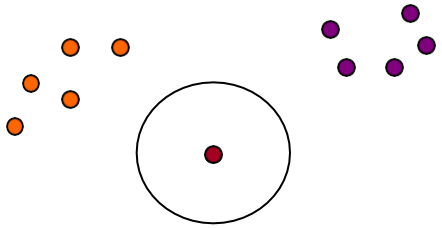
- **Parametri**
  - ε = 2 cm
  - *MinPts* =3



remove all noise points
for each core-object *o* do
if *o* is not yet classified then
assign *o* to a new cluster *C*
collect all objects density-reachable from *o*
and assign them to *C*
end

DBSCAN Primer

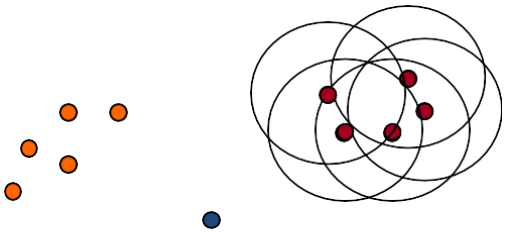
- **Parametri**
  - ε = 2 cm
  - *MinPts* =3



remove all noise points
for each core-object *o* do
if *o* is not yet classified then
assign *o* to a new cluster *C*
collect all objects density-reachable from *o*
and assign them to *C*
end

DBSCAN Primer

- **Parametri**
  - ε = 2 cm
  - *MinPts* =3



remove all noise points
for each core-object *o* do
if *o* is not yet classified then
assign *o* to a new cluster *C*
collect all objects density-reachable from *o*
and assign them to *C*
end

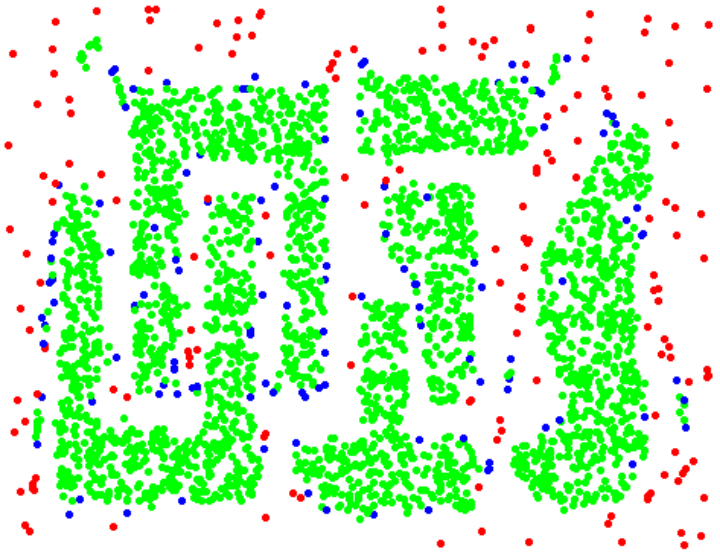
DBSCAN Алгоритам

- Елиминишу се тачке шума
- Кластеринг се изврши над преосталим тачкама

current_cluster_label $\leftarrow 1$
for all core points **do**
  if the core point has no cluster label **then**
    current_cluster_label $\leftarrow$ *current_cluster_label* $+ 1$
    Label the current core point with cluster label *current_cluster_label*
  end if
  for all points in the *Eps*-neighborhood, except i^{th} the point itself **do**
    if the point does not have a cluster label **then**
      Label the point with cluster label *current_cluster_label*
    end if
  end for
end for

DBSCAN: тачке језгра, ивичне тачке, тачке шума

Оригиналне тачке

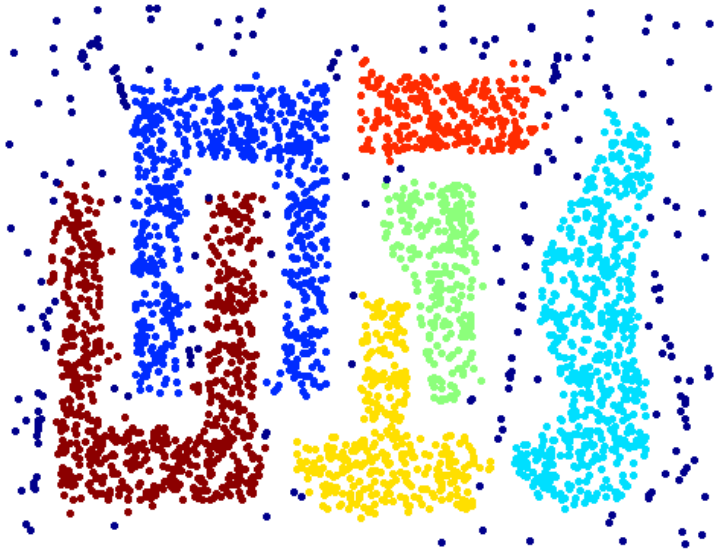


Типови тачака:
језгро, ивица и шум

Eps = 10, MinPts = 4

Када DBSCAN добро ради

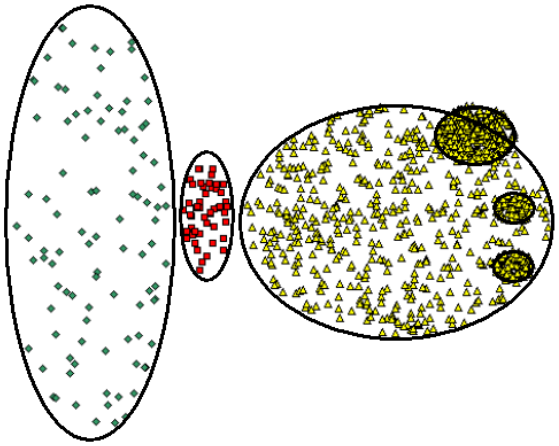
Оригиналне тачке



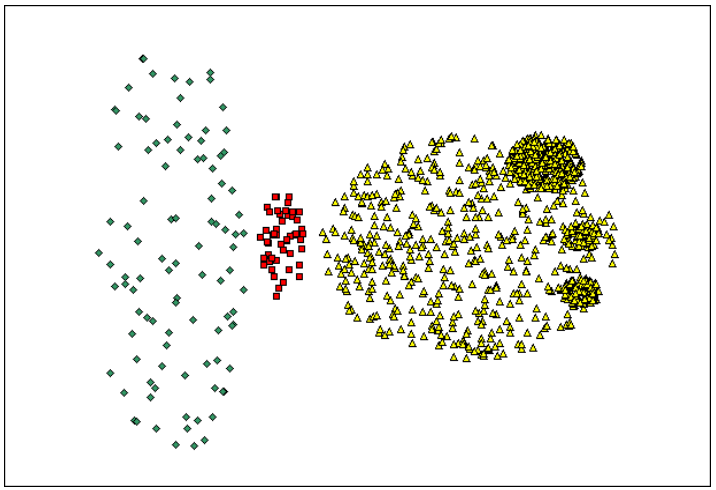
Кластери

- Отпоран на шум
- Може да ради са кластерима различитих облика и величине

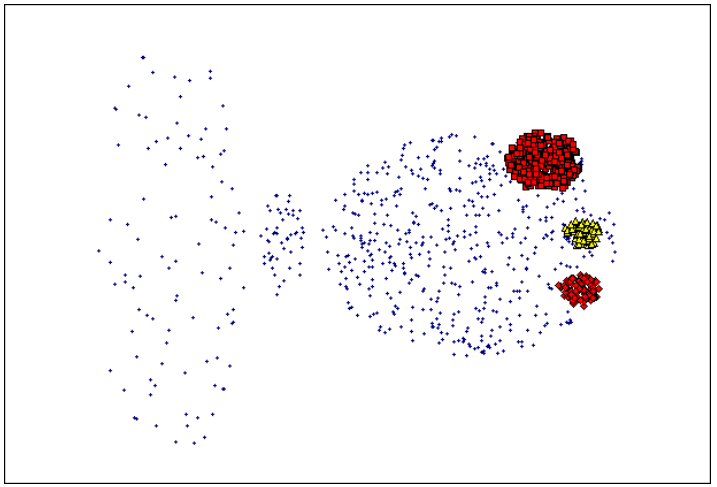
Када DBSCAN НЕ РАДИ добро



Оригиналне тачке



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

- **Варијабилне густине**
- **Високо димензионални подаци**

DBSCAN: Одређивање EPS и MinPts

- Идеја је да за тачке из кластера, k^{te} најближе комшије буду на приближно истом растојању
- Тачке шума за k^{te} најближе комшије су на већем растојању
- Дакле, нацртају се сортиране дистанце сваке тачке до сваког њеног k^{tor} најближег комшије