Klaszterezés

Csima Judit

BME, VIK,
Számítástudományi és Információelméleti Tanszék

2015. április 8. és 9.

Klaszterezés

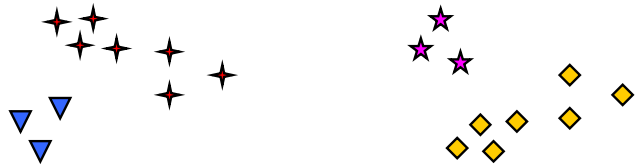
- cél: dolgokat úgy csoportokba osztani, hogy a hasonlók kerüljenek egy csoportba
- dolog: n hosszú vektorok, az egyes koordináták az attribútumoknak felelnek meg (mint eddig)
- unsupervised learning: nincs címkézés, ami segít, az attribútumértékek egymáshoz való viszonya alapján kell csoportosítani
- fő elvek:
  - csoporton belüli max. távolság minél kisebb legyen
  - csoportok közti min. távolság minél nagyobb legyen
- nem precíz a feladat

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters



Four Clusters

Példák klaszterezésre

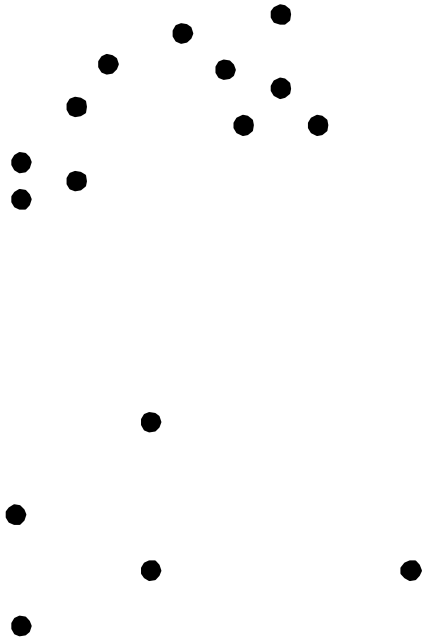
- dokumentumok csoportosítása hasonlóság (közös téma) alapján
- market segmentation
- social network analysis
- exploratory analysis része is lehet, találjunk valami mintát az adatban
- hagyományos algoritmusok első része is lehet: utazóügynök feladat megoldása

Klaszterezés fajtái

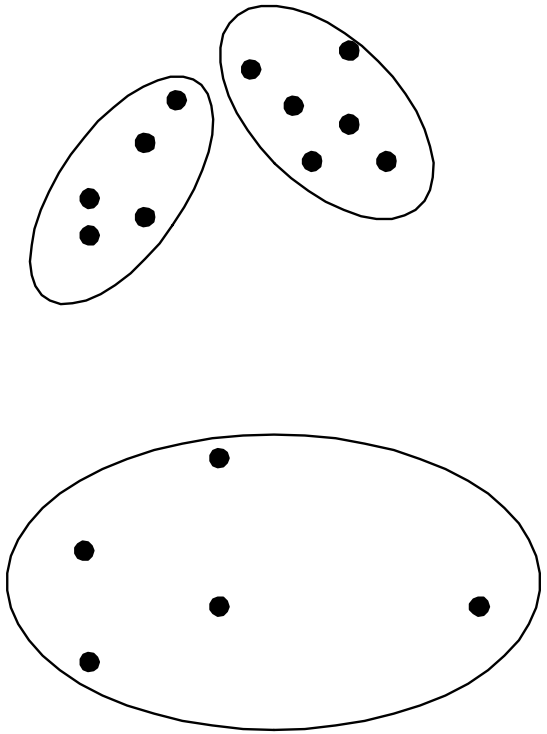
hierarchikus vs. partitional:

- partitional: valahogyan felosztjuk a pontokat részekre, egy pont pontosan egy halmazba kerül
- hierarchical: a klaszterek egymásba ágyazottak, egy csúcs több, egyre nagyobb klaszterbe tartozhat

Partitional Clustering

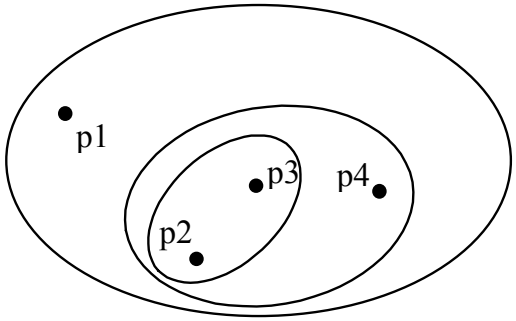


Original Points

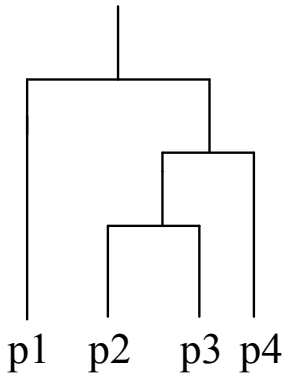


A Partitional Clustering

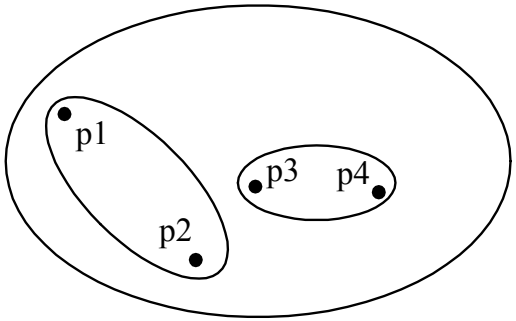
Hierarchical Clustering



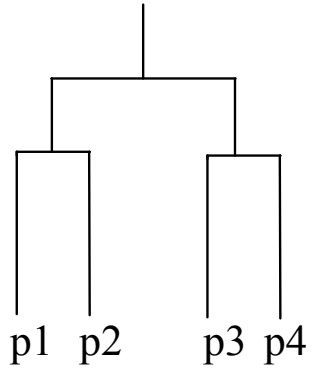
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Klaszterezés fajtái még (de ezekről nem lesz szó)

exclusive vs. overlapping vs. fuzzy

- exclusive: egy csúcs csak egy helyre tartozik
- overlapping: egy csúcs tartozhat több klaszterbe is (market segmentation esetén lehet olyan vevő, aki krimit és gyerekkönyvet is vesz)
- fuzzy: egy pont egy adott valószínűséggel tartozik az egyes klaszterekbe

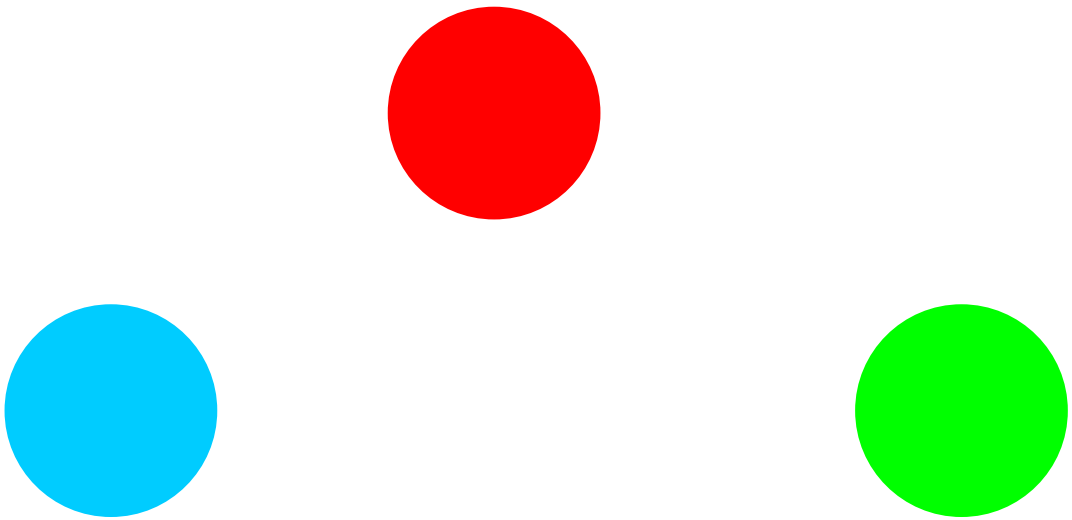
Klaszterek fajtái

Fő elv: úgy csoportosítani, hogy hasznos csoprtok jöjjenek létre.
Kérdés: mi a hasznos, mi definiálja az egyes klasztereket?

- well- separated clusters
  - a csoporton belül bármely két pont hasonlóbb egymáshoz, mint akármelyik két, külön csoportban levő csúcs
  - ez nem mindig lehetséges

Types of Clusters: Well-Separated

- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

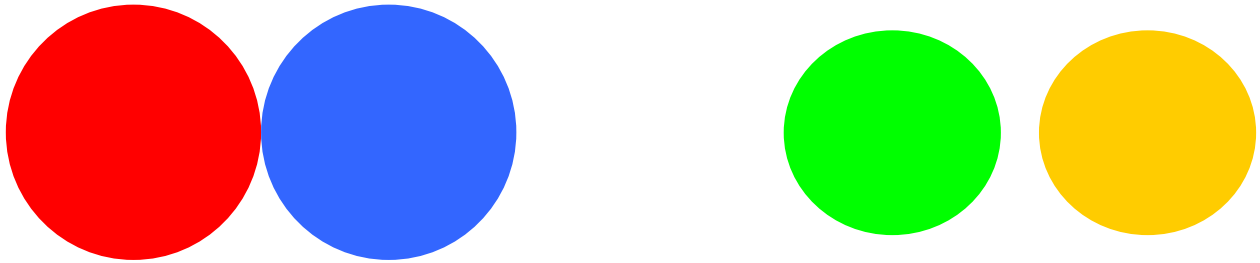
Klaszterek fajtái még

prototype-based or center-based clusters:

- minden klaszternek van egy reprezentánsa
- minden csúcs abba a klaszterbe kerül, aminek a reprezentánsához legközelebb van
- folytonos attribútumok esetén általában centroid: átlag
- kategorikus attribútum esetén: medoid, többségi címke

Types of Clusters: Center-Based

- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most “representative” point of a cluster



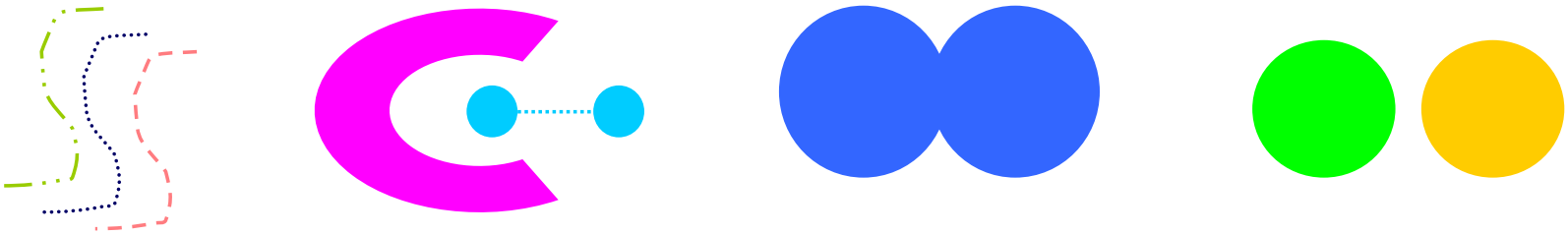
4 center-based clusters

klaszterek fajtái még

- gráf alapú
  - a pontok a csúcsok és él van bizonyos esetben két csúcs között
  - pl. él van, ha a távolságuk egy küszöbnél kisebb
  - klaszter: összefüggő komponensek
  - ekkor (ha van legalább két csúcs a klaszterben) minden csúcshoz van egy vele egy csoportban levő másik, aki közelebb van hozzá, mint bármelyik, más csoportba eső csúcs
  - az ilyen csoportosítás neve: contiguity-based clustering

Types of Clusters: Contiguity-Based

- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



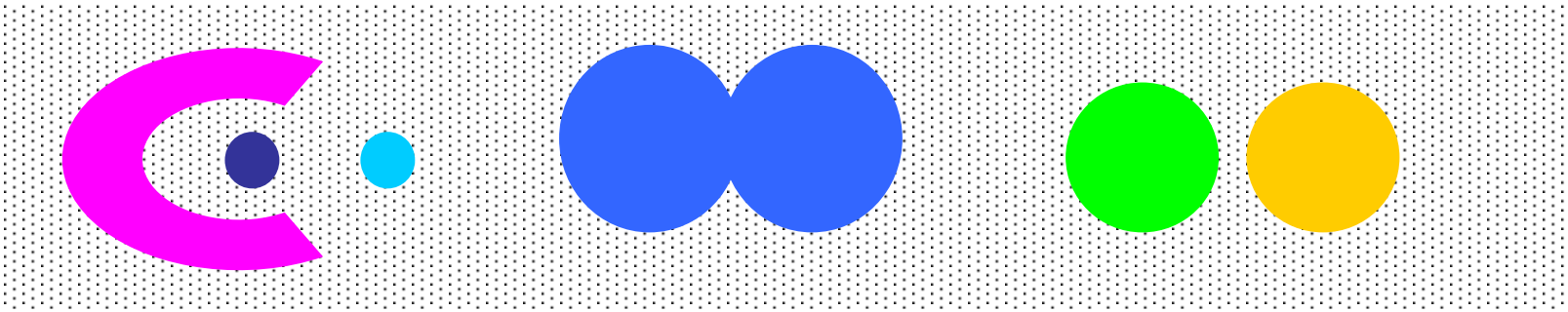
8 contiguous clusters

Klaszterek fajtái még

- sűrűség alapú klaszterezés
  - a klaszterek nagy pontsűrűségű részek, amiket kisebb sűrűségű részek választanak el
- célfüggvénnyel definiált klaszterezés
  - valami célfüggvény van, ami minden felosztásra ad egy értéket
  - keressük azt a felosztást, amire ez az érték a legkisebb vagy legnagyobb
  - pl. lehető legnagyobb, klaszterek közti legkisebb távolság, lehető legkisebb, klaszteren belüli legnagyobb távolság

Types of Clusters: Density-Based

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.



6 density-based clusters

Mit tanulunk mi?

- K-means: partitional, prototype-based, adott darabszámú csoportot csinál (K)
- hierarchikus klaszterezés: összevonásokkla csinál egyre nagyobb elemszámú csoportokat

K-means algo

Adott egy K szám, ennyi csoportot akarok

1. Választok K darab kezdő centroidot az n dimenziós térben (n darab attribútumból áll egy pont), nem kell adat-pontnak lennie
2. Minden adatpontot hozzácsatolok a legközelebbi centroidhoz
3. A kapott csoportokra újraszámolom a centroidokat

2. és 3. pontot iterálom, amíg már nincs változás

K-means kérdések

- Mi a közeli? Mi a távolság, amit használok?
- Hogyan számolom ki az új centroidokat?
- Ez mindig konvergálni fog?

Távolság

- sok mindent lehet használni
- szokásos az L_2, de lehet L_1, cosine, Jaccard is, attól függ, hogy milyen típusú az adathalmaz

Centroidok számolása L_2 távolság esetén

- általában SSE-t minimalizáló felosztást keresek
- $\text{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} dist(x, c_i)^2$, ahol
  C_i az i. csoport, ennek centroidja c_i
- azaz K centroidot akarok találni és egy ezekhez való hozzárendelést
- úgy, hogy a pontok saját centroidjaiktól vett távolságnégyzetek összege minimális legyen
- ez meghatározza, hogy egy adott csoportosításra mi lesz az optimális centroid választás
- ez a mean lesz ebben az esetben

Centroidok számolása általában

- SSE-t vagy valami ehhez hasonló mennyiséget minimalizáló felosztást keresek
- pl. SAE $= \sum_{i=1}^{K} \sum_{x \in C_i} dist(x, c_i)$, ahol $dist$ az L_1 távolság
- tehát most a pontok saját centroidjaiktól vett távolságösszege legyen minimális
- ekkor egy adott csoportosításra az optimális centroid választás a median lesz

Konvergencia különböző távolság és centroidszámolás esetén

Az alábbi esetekben bizonyítottan konvergál az algo

- L_1 és median
- L_2 és mean
- cosine és mean (SAE-szerű objective function)

Általában olyan gyorsan konvergál, hogy elég azt mondani, hogy álljuk le I darab kör után vagy akkor, ha már csak kis százaléka vándorol a pontoknak

K-means felépítése SSE minimalizás esetén

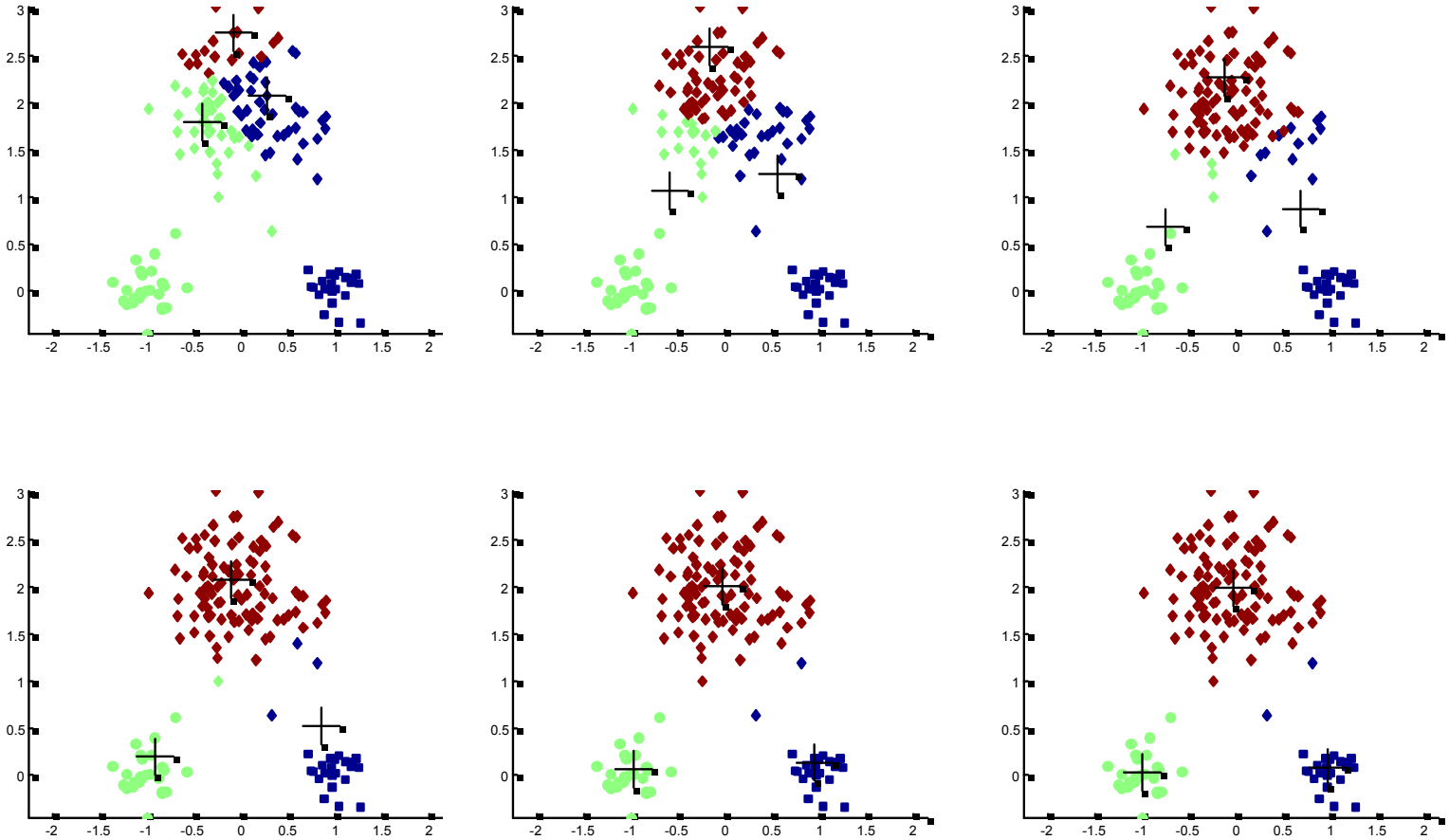
Két lépés váltogatásával keresi a legjobb megoldást:

- egyik lépésben adott centroidhoz keres csoportosítést
- másik lépésben adott csoportosításhoz keres centroidot

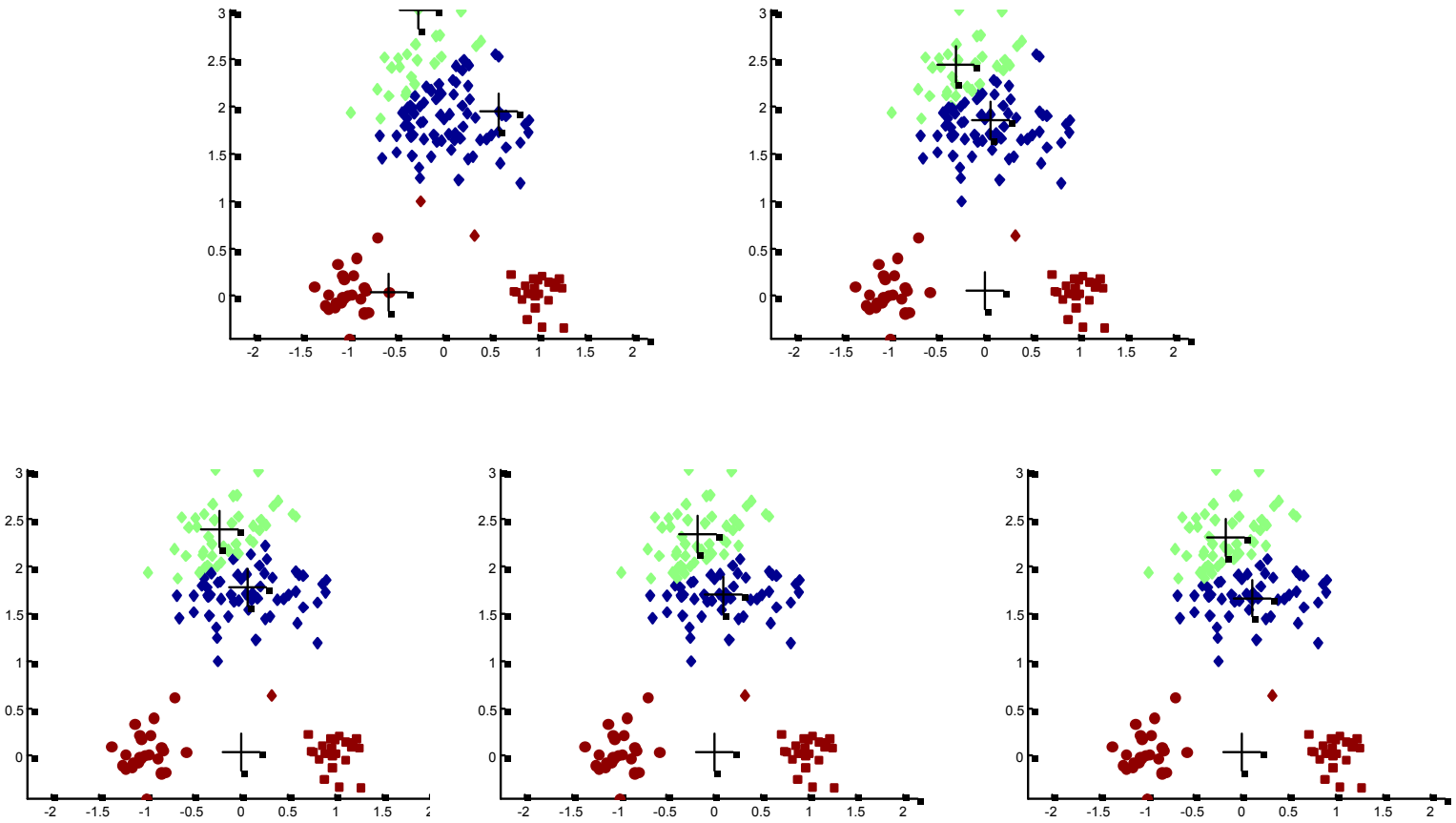
Kezdeti centroidok megválasztása

- a végén kialakuló klaszterek attól is függenek, hogy honnan indítjuk az algot
- lehet, hogy egy béna kezdő választással teljesen rossz klasztereket kapunk, még akkor is, ha vannak szép, természetes csoportok

Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids ...



Kezdő-centroidokra megoldás

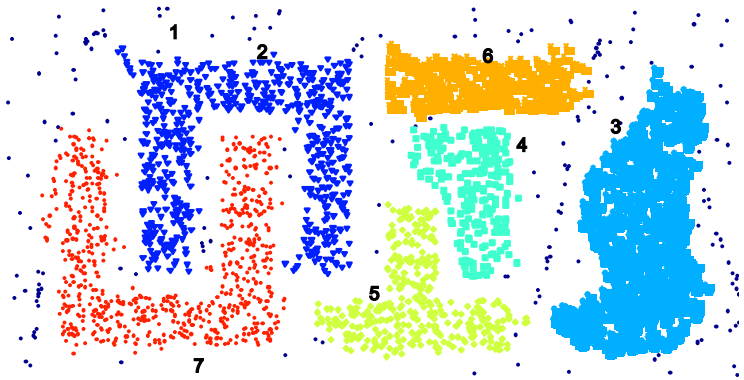
- randomizálás: sokszor megcsinálom az algot, különféle véletlen kezdőpontokból indítva
- a kezdőpontok training pontok lesznek, azaz választok K training pontot és innen indítok
- Kérdés: melyik klaszterezés lesz a nyerő a végén?
- Ahol a minimalizálni kívánt SSE a legkisebb.
- Ez nem mindig megoldás, alternatív megoldás: bisecting K-means (erről később)

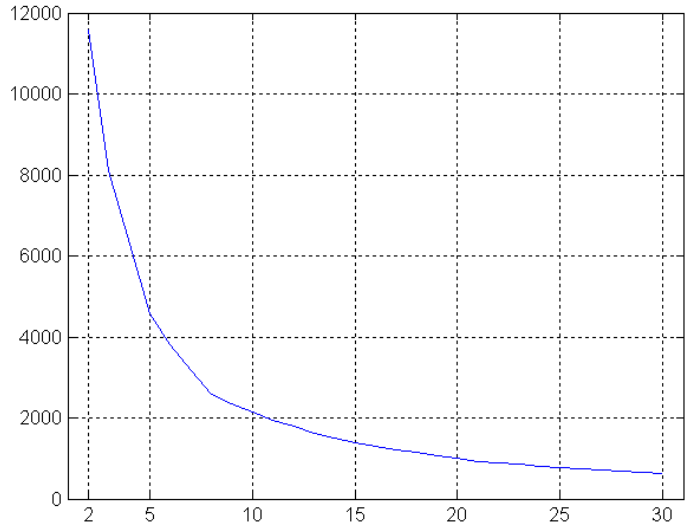
K megválasztása

- attól függ, hogy mire kell a klaszterezés
- több K-t kipróbálunk (mindet sok random kezdéssel)
- ha később fel akarjuk használni valamire a csoportokat, akkor azt a K-t választjuk, aminél a klaszterezést használó alkalmazás a legkívánatosabb eredményt adja
- ha pusztán az SSE a jóság mérése: könyök-szabályt próbálom használni (egyre nagyobb K-k, ahol lelassul a hiba csökkenése, ott állok meg)

Internal Measures: SSE

- SSE curve for a more complicated data set





SSE of clusters found using K-means

Megjegyzések a K-meansről: üres klaszter

Lehet, hogy üres klaszter keletkezik (minden pont elpártol az egyik centroidtól).

- eggyel kevesebb klaszter lesz, ez baj lehet
- új centroidot kellene ez helyett választani a következő iterációra
- több megoldás szokásos:
  - az a pont, aki legtávolabb van minden centroidtól
  - a legnagyobb SSE-jű klaszterből valaki

Megjegyzések a K-meansről: postprocessing

- miután van egy klaszterezésünk K csoportra, megpróbáunk rajta javítani (esetleg K változhat)
- K (kis) növelésével (nagyon) csökkenthető-e az SSE
  - nagy SSE-jű klaszter kettévágása
  - új centroid bevezetése, pl. a minden eddigi centroidtól legtávolabbi pont
- K kis csökkentése nem romlik sokat az SSE
  - kis elemszámú klaszter (vagy kicsit rontó) klaszter centroidját kidobjuk
  - két közeli centroid klaszterét összevonjuk

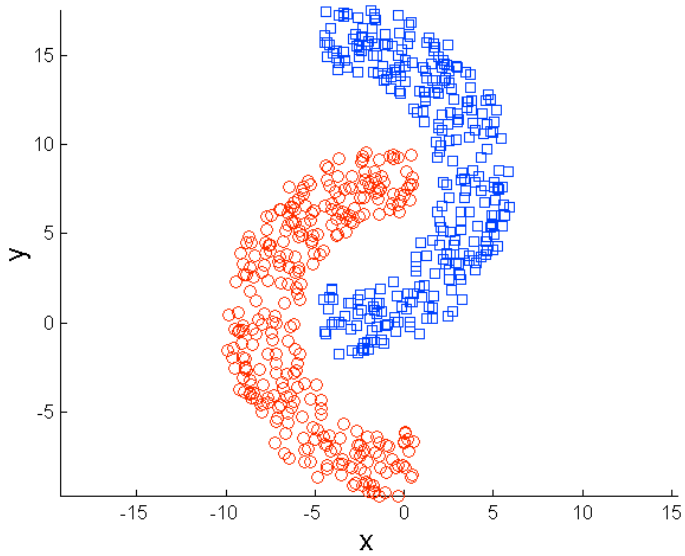
Bisecting K-means

- egy klaszterből indulunk és minden lépésben egy kiválasztott klasztert kettévágunk, amíg K klaszter nem lesz
- a kettévágás a hagyományos K-means algoval történik, $K = 2$ választással és persze sok random indítással
- arra is használatos, hogy a végén kapott K darab centroiddal indítunk egy szokásos K-means algot

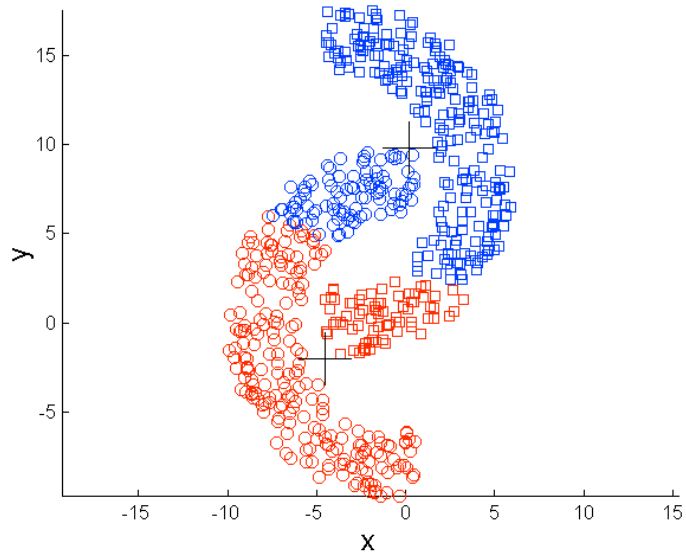
Záró megjegyzések a K-means algoritmusról: gondok

- K-means nem túl ügyes, ha a természetes klaszterek (amiket jó lenne megtalálni) nem gömbszerűek
- gond lehet még: eltérő nagyságú természetes klaszterek
- gond még: eltérő sűrűségű természetes klaszterek

Limitations of K-means: Non-globular Shapes

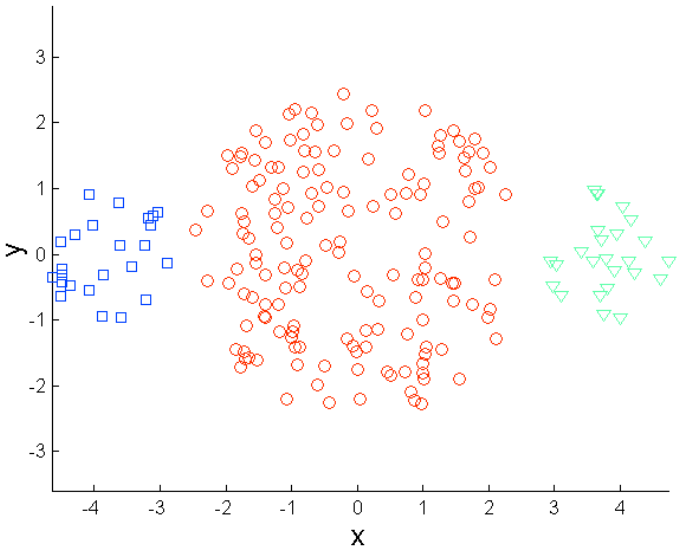


Original Points

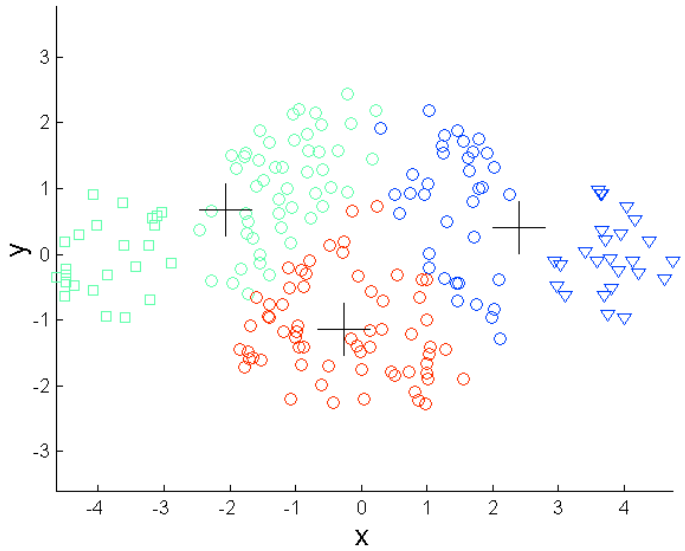


K-means (2 Clusters)

Limitations of K-means: Differing Sizes

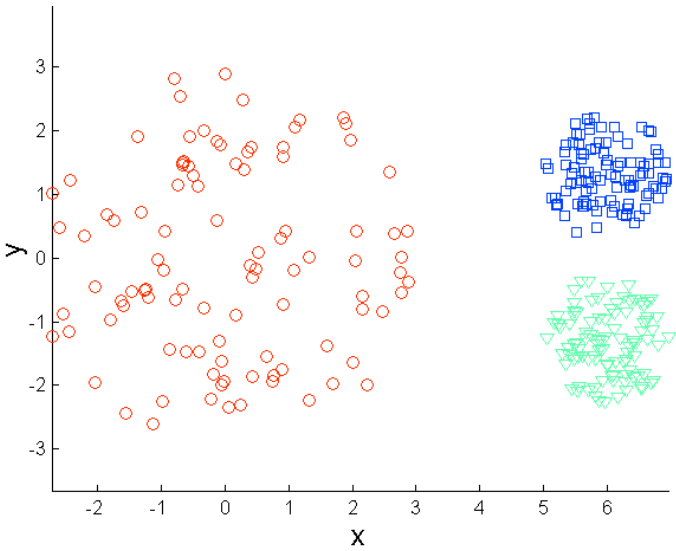


Original Points

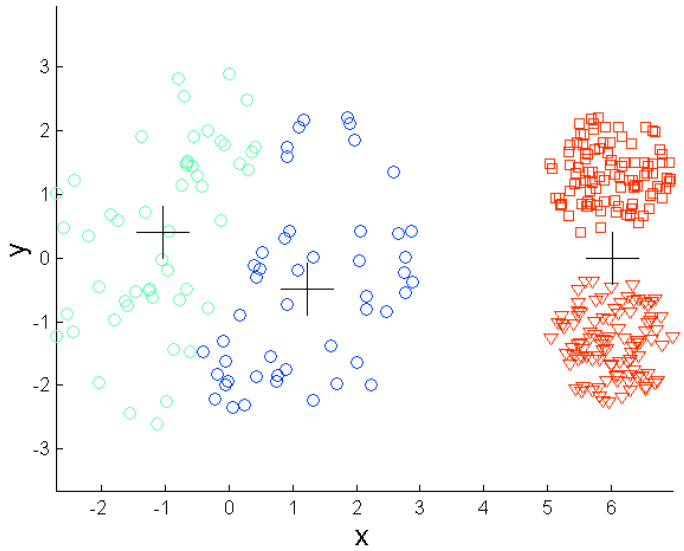


K-means (3 Clusters)

Limitations of K-means: Differing Density



Original Points

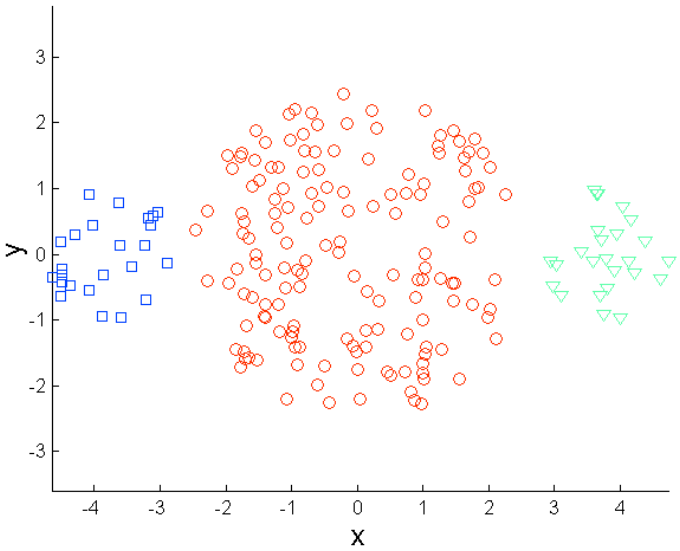


K-means (3 Clusters)

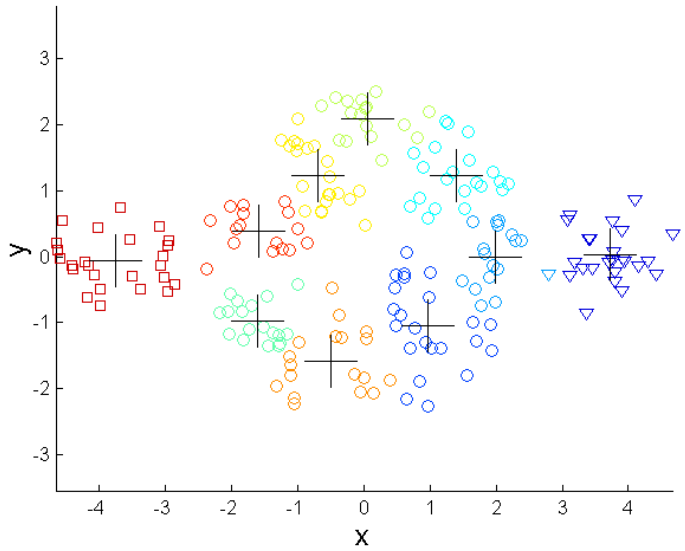
Záró megjegyzések a K-means algoritmusról: megoldások, jóság

- K-means egyszerű, gyors
- az előbbi problémákra megoldás lehet, ha nagyobb K-t használunk és így egy természetes klaszter több megtalált csoport uniója lesz
- R-ben kmeans függvény

Overcoming K-means Limitations



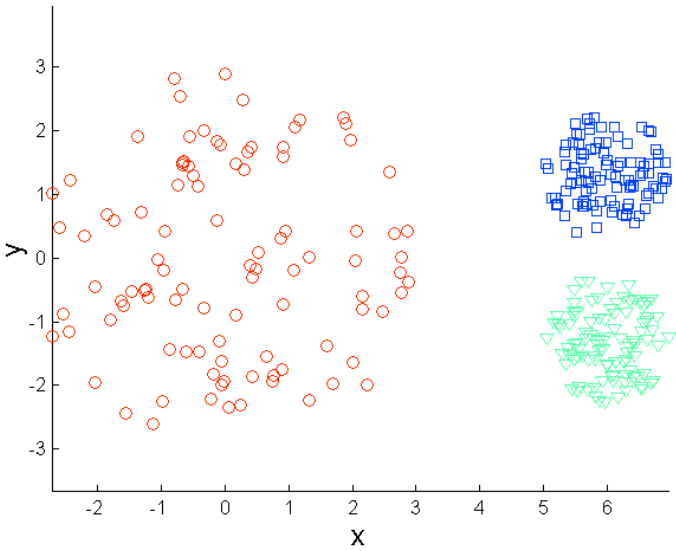
Original Points



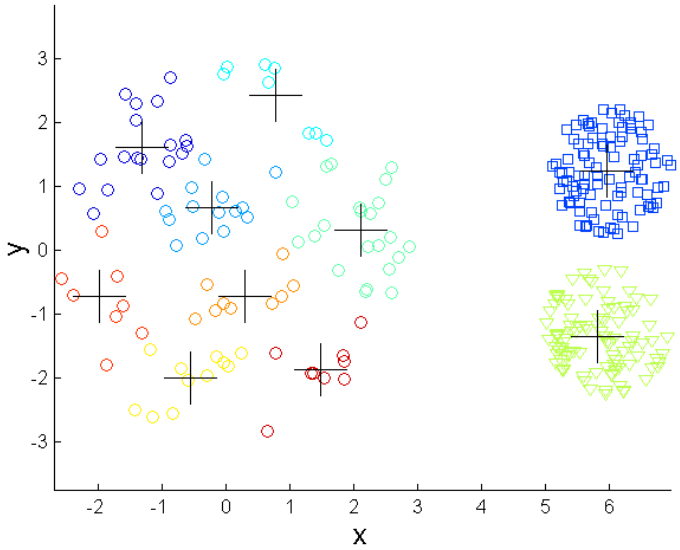
K-means Clusters

One solution is to use many clusters.
Find parts of clusters, but need to put together.

Overcoming K-means Limitations

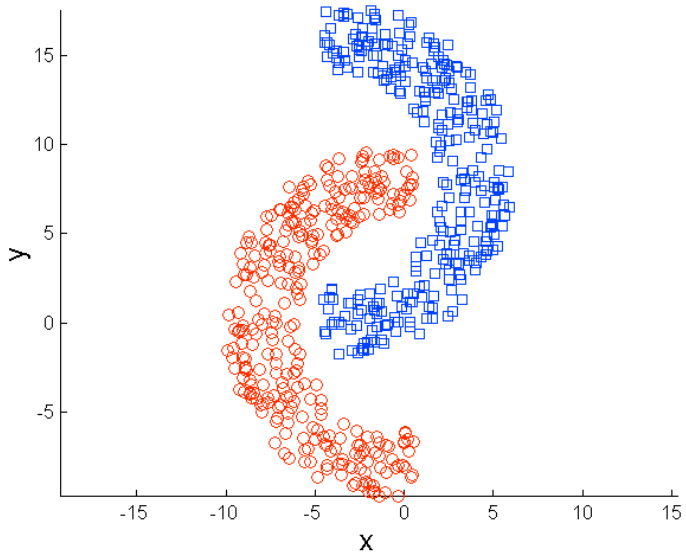


Original Points

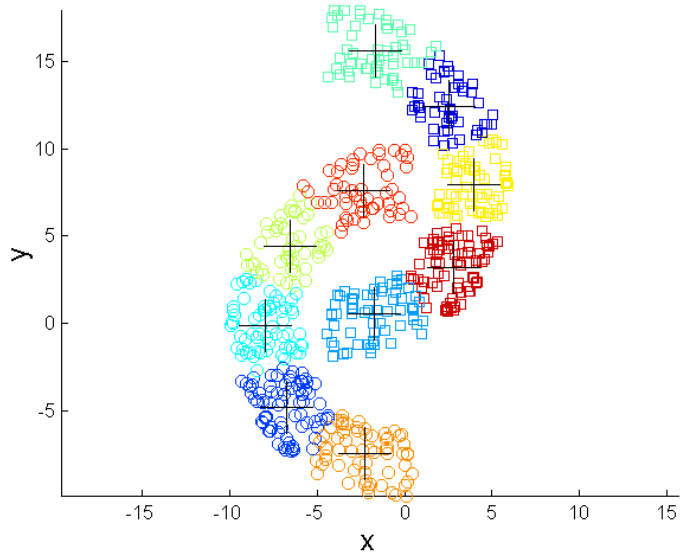


K-means Clusters

Overcoming K-means Limitations



Original Points



K-means Clusters