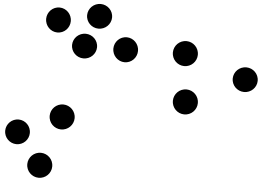
Klaszteranalízis
k-közép módszer

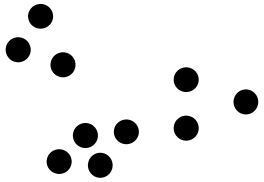
Mit nevezünk klaszteranalízisnek?

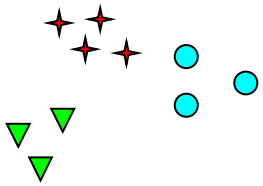
- A klaszteranalízis adatobjektumokat csoportosít kizárólag azon információk alapján, amelyeket azokban az adatokban talál, melyek az objektumokat valamint a köztük fennálló kapcsolatokat írják le.
- A cél az, hogy az egy csoporton belüli objektumok hasonlóak legyenek egymáshoz (vagy kapcsolat legyen közöttük), és különbözőek legyenek más csoportok objektumaitól (vagy ne álljanak ezekkel kapcsolatban).
- Minél nagyobb a hasonlóság (vagy homogenitás) a csoportokon belül és minél nagyobb a különbség az egyes csoportok között, annál jobb vagy pontosabb a klaszterezés.

Klaszteranalízis - Példa

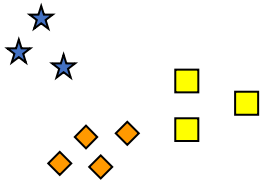


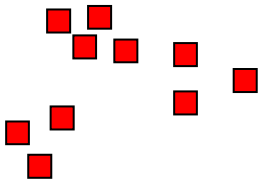
Eredeti pontok



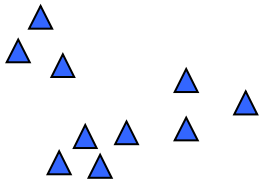


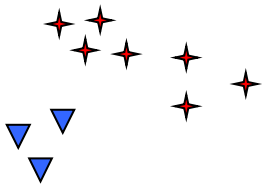
Hat klaszter



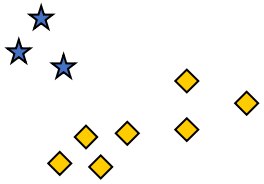


Két klaszter





Négy klaszter



Klaszteranalízis

- Számos alkalmazásban nincs jól meghatározva a klaszter fogalma.
- Azért, hogy jobban megértsük annak a döntésnek a nehézségét, hogy mi alkot egy klasztert, tekintsük az előző példában szereplő húsz pontot, valamint azok klaszterekre osztásának három különböző módját.
- Ez az ábra azt szemlélteti, hogy a klaszterek definíciója pontatlan, és hogy a legjobb definíció az adatok természetétől és a kívánt eredménytől függ.

A klaszterezés különböző típusai

- A klaszterezés klaszterek halmazát adja
- **Partícionáló (felosztó) klaszterezés**
- Az adathalmazban szereplő adatok besorolása nem-átfedő részhalmazokba (klaszterekbe), ahol minden adatelem pontosan egy részhalmazba kerül
- **Hierarchikus klaszterezés**
- a klaszteréknek alklasztereik lehetnek,
- egymásba ágyazott klaszterek, hierarchikus rendszerbe (egy fába) szervezve, ahol a fa minden csúcsa (klasztere), a levélcsúcsokat kivéve, a gyermekei (alklaszterei) uniója, és a fa gyökere az összes objektumot tartalmazó klaszter
- a fa levelei gyakran egyetlen adatobjektumot tartalmazó egyelemű klaszterek.

A klaszterezés különböző típusai

- **kizáró (exclusive) klaszterezés**: mindegyik objektumot csak egyetlen klaszterhez rendelik hozzá.
- **átfedő (overlapping), vagy nem-kizáró (non-exclusive) klaszterezés**: egy objektum egyszerre egynél több csoporthoz (osztályhoz) is tartozhat.
- **Fuzzy klaszterezés:** minden objektum minden klaszterbe beletartozik egy tagsági súly erejéig, melynek értéke 0 (egyáltalán nem tartozik bele) és 1 (teljesen beletartozik) közé esik. (valószínűségi klaszterező módszerek)
- gyakran kizáró klaszterezéssé konvertálják: mindegyik objektumot abba a klaszterbe sorolják be, amelynél a legnagyobb a tagsági súlya vagy a valószínűsége.

Módszerek

- **k-közép módszer:** Ez egy prototípus-alapú, felosztó klaszterező módszer, amely megpróbálja megkeresni a felhasználó által megadott számú (k) klasztert, amelyeket a középpontjaik képviselnek.
- **összevonó (agglomeratív) hierarchikus klaszterezés**: kezdetben minden elem egy külön klaszterbe tartozik, és ezután minden lépésben a két legközelebbi klaszter kerül összevonásra egészen addig, amíg már csak egyetlen, minden elemet magába foglaló klaszter nem marad.

k-közép és k-medoid módszer

- A prototípus-alapú klaszterező eljárások az adatobjektumok egyszintű particionálását állítják elő, a két legfontosabb módszer a k-közép és a k-medoid.
- A k-közép módszer egy középpontot (centroidot) választ ki prototípusnak, amely általában pontok egy csoportjának az átlaga, és jellemzően csak folytonos, n-dimenziós térben elhelyezkedő pontokra alkalmazható.

k-közép és k-medoid módszer

- Ezzel szemben a k-medoid módszer a prototípust a medoiddal definiálja, amely pontok egy csoportjának a legjellemzőbb pontja.
- Ez a módszer adatok szélesebb körében használható, hiszen csak az objektumpárok közötti szomszédsági mértékre van szükség.
- Míg a centroid szinte sosem felel meg egy valós adatpontnak, addig a medoid -- definíciójából adódóan -- mindig egy konkrét adatpont.

k-közép módszer

- az egyik legrégebbi és legszélesebb körben használt klaszterező algoritmus
- Alapvető k-közép algoritmus

  1: Válasszunk ki k kezdeti középpontot

  2:**repeat**

  3: Hozzunk létre k klasztert mindegyik adatpontnak a legközelebbi középponthoz rendelésével

  4: Számoljuk újra mindegyik klaszter középpontját

  5: **until** a középpontok nem változnak

k-közép módszer

- Első lépésként kiválasztunk k darab kezdő közép-pontot, ahol k a felhasználó által megadott paraméter, nevezetesen a klaszterek kívánt száma.
- Minden adatpontot a hozzá legközelebb eső középponthoz rendelünk, és az így képzett csoportok lesznek a kiinduló klaszterek.
- Mindegyik klaszter középpontját a klaszterhez rendelt pontok alapján frissítjük.
- A hozzárendelési és frissítési lépéseket felváltva folytatjuk addig, amíg egyetlen pont sem vált klasztert, vagy ezzel egyenértékűen, míg a középpontok ugyanazok nem maradnak.

k-közép módszer lépései

- **Pontok legközelebbi középponthoz rendelése**
- ehhez szükségünk van egy olyan közelségi (távolsági) mértékre, amely meghatározza a ''közelség'' fogalmát. Az euklideszi távolságot gyakran használjuk euklideszi térben elhelyezkedő pontoknál, míg dokumentumoknál alkalmasabb a koszinusz hasonlóság.
- más közelségi mértékek, melyek hasznosak lehetnek bizonyos típusú adatok esetén: Manhattan távolság, amelyet euklideszi térben elhelyezkedő adatokra lehet alkalmazni, valamint a Jaccard-mérték, melyet dokumentumok esetén használnak előszeretettel.

k-közép módszer lépései

- **Pontok legközelebbi középponthoz rendelése**
- A k-közép algoritmus általában egyszerű hasonlósági mértékeket használ, mivel az algoritmus újra meg újra kiszámolja a pontok és a középpontok közötti hasonlóságot. Néhány esetben, mint például az alacsony dimenziójú euklideszi terekben elhelyezkedő adatok esetén, azonban lehetőség van arra, hogy elkerüljük sok hasonlóság kiszámítását, így gyorsítva fel jelentősen a k-közép algoritmust. (kettéosztó k-közép módszer)

k-közép módszer lépései

- **Középpontok és célfüggvények**
- Az algoritmus negyedik lépése: ''újraszámoljuk a klaszterek középpontjait'', ahol a középpont számítá-sának módja változhat a közelségi mértéktől és a klaszterezés céljától függően.
- A klaszterezés célját jellemzően egy célfüggvény fejezi ki, a pontok egymás közötti vagy a pontok és a közép-pont közötti közelségtől (távolságtól) függően, például minimalizáljuk a pontok és a hozzájuk legközelebbi középpont négyzetes távolságának az összegét.
- adott közelségi mérték és célfüggvény esetén a válasz-tandó középpont gyakran meg is határozható.

k-közép módszer lépései

- **Négyzetes távolság (Sum of Squared Error, SSE)**
- minden pontra a legközelebbi középponttól való távolság
- **SSE definíciója:**

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

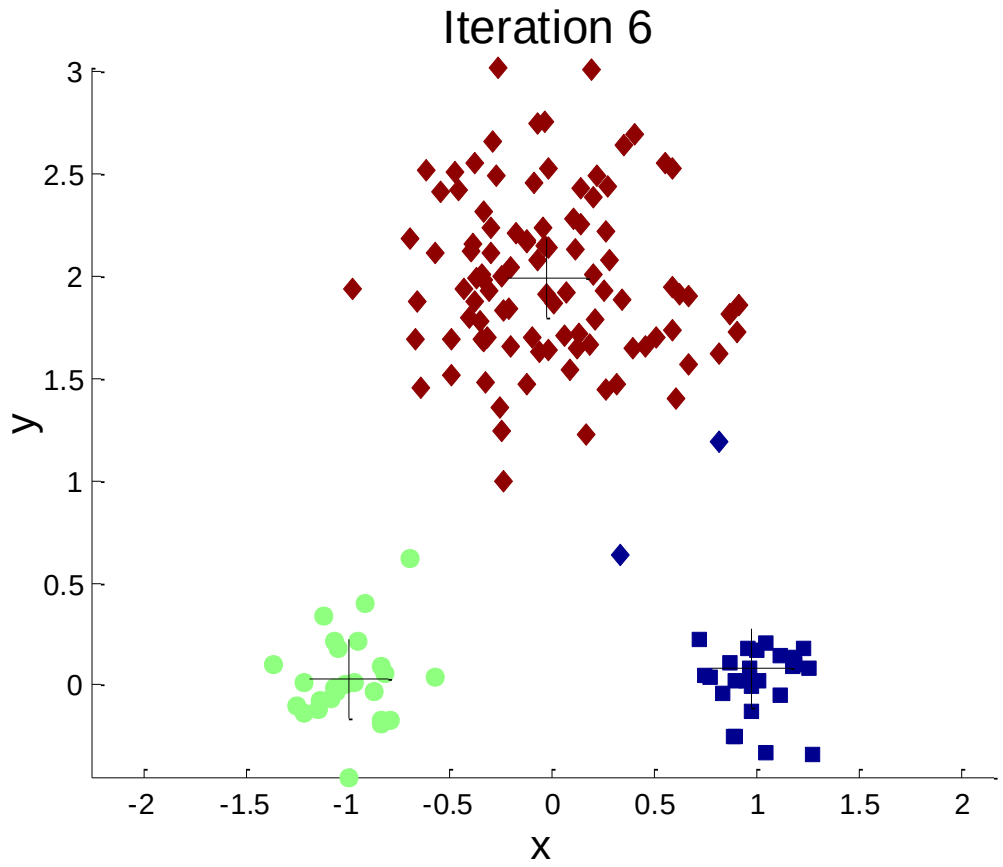
- x a C_i klaszterben van, m_i a C_i klasztert reprezentáló pont
- Ha adott két klaszterezésünk, a kisebb hibájút választjuk
- Az SSE általában csökken k növelésével

k-közép módszer lépései

- **Középpontok és célfüggvények**
- A k-közép algoritmus 3. és 4. lépésénél az SSE (a célfüggvény) minimalizálása a cél. A 3. lépésben klasztereket hozunk létre úgy, hogy a pontokat a hozzájuk legközelebb eső középponthoz rendeljük, amely minimalizálja az SSE-t középpontok egy adott halmazára. A 4. lépésben újraszámoljuk a középpontokat, így csökkentve tovább az SSE-t. Az algoritmus 3. és 4. lépése azonban csak a lokális minimum megtalálását garantálja az SSE-re nézve, mivel csak a középpontok és a klaszterek speciális megválasztása mellett optimalizálja az SSE-t, nem minden lehetséges esetre.

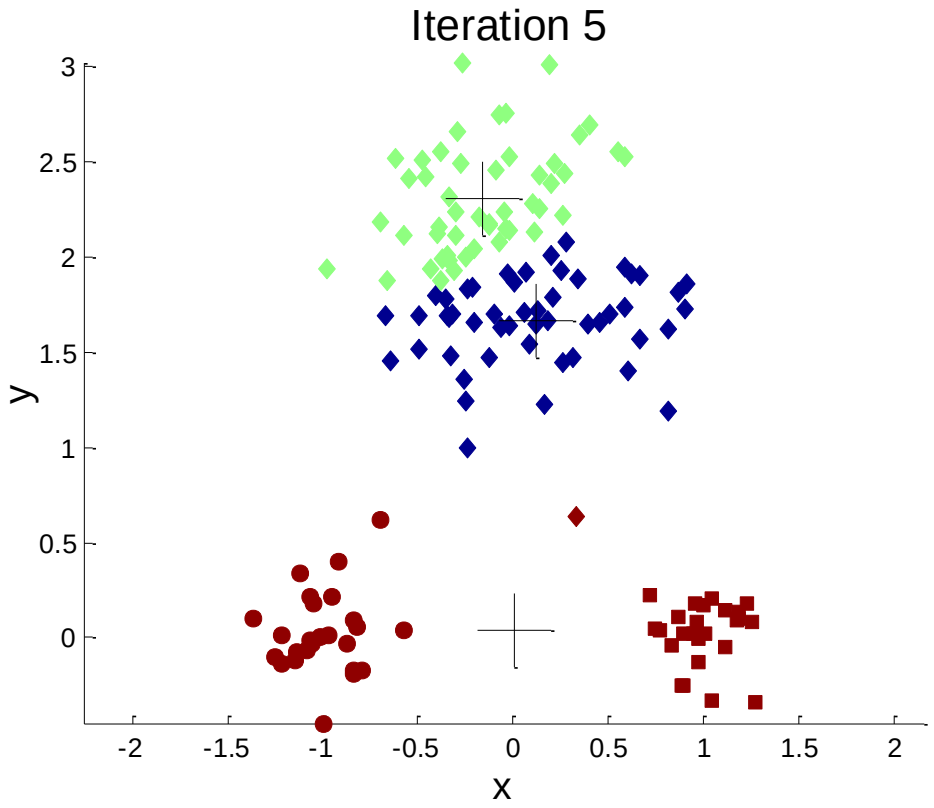
k-közép módszer lépései

Kezdeti középpontok jó választása



k-közép módszer lépései

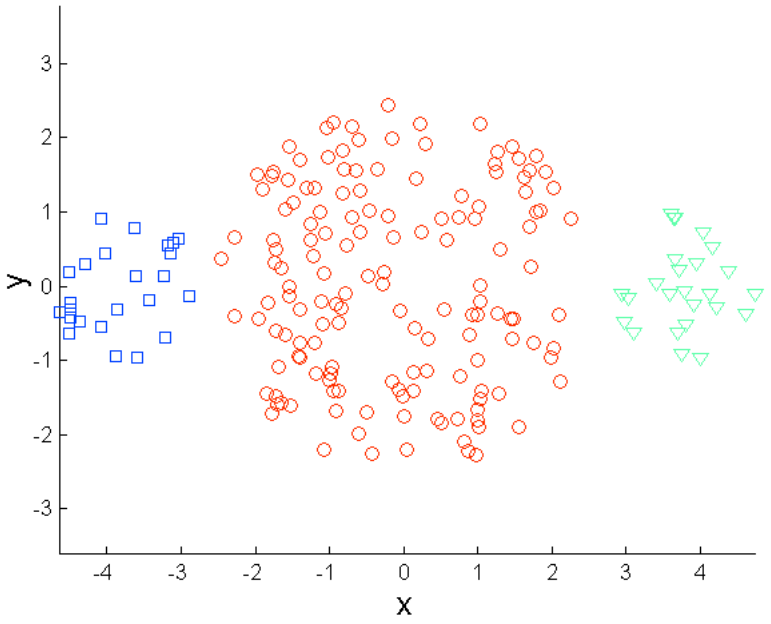
Kezdeti középpontok rossz választása



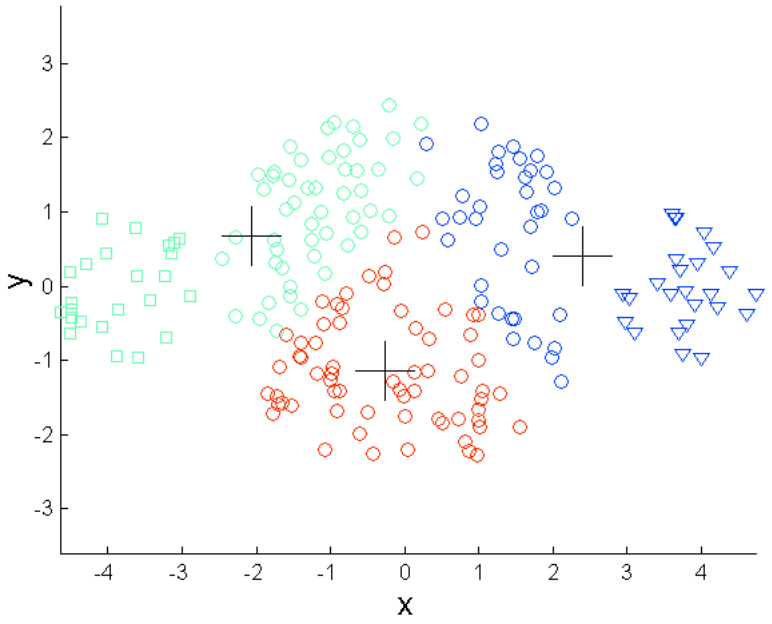
k-közép módszer korlátai

- k-középnek akkor vannak problémái, ha a klaszterek nagyon különböznek
  - Méretben
  - Sűrűségben
  - Nem gömbszerű alakúak vagy ha sok kiugró érték van

k-közép módszer korlátai

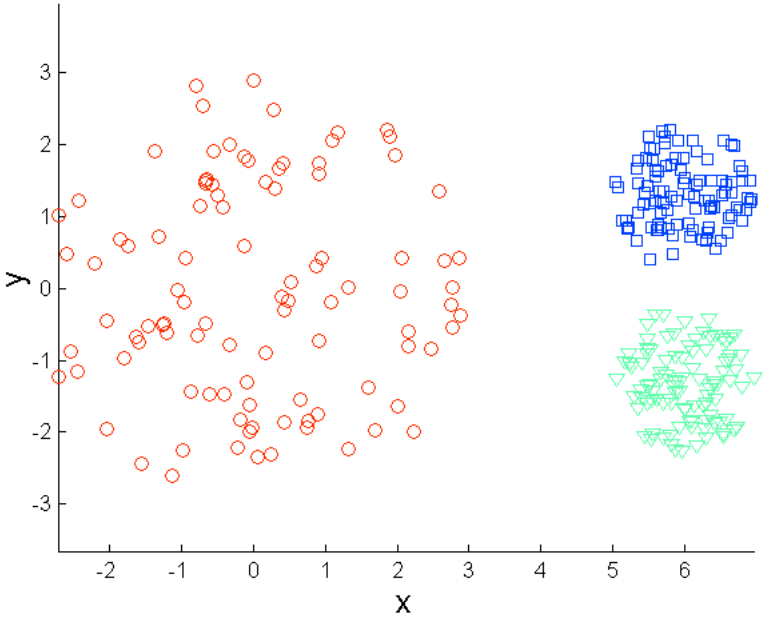


Eredeti pontok

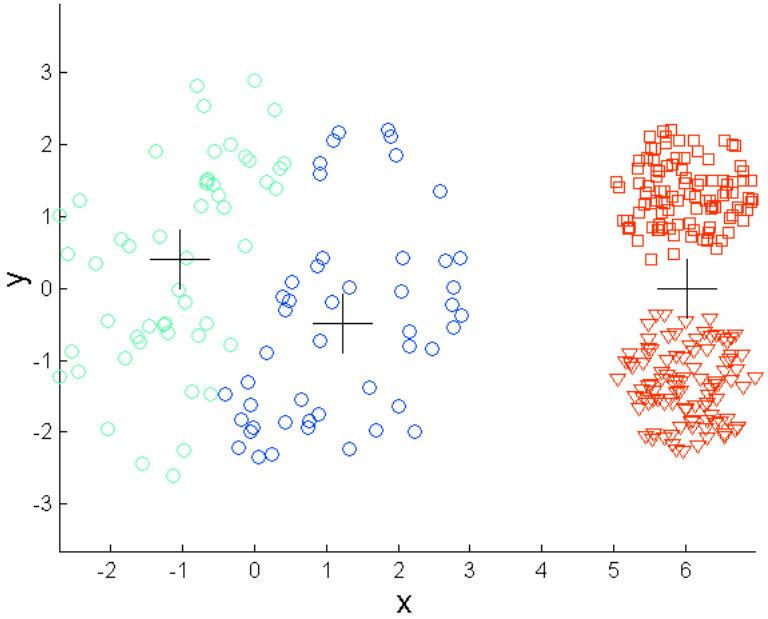


k-közép (3 klaszter)

k-közép módszer korlátai

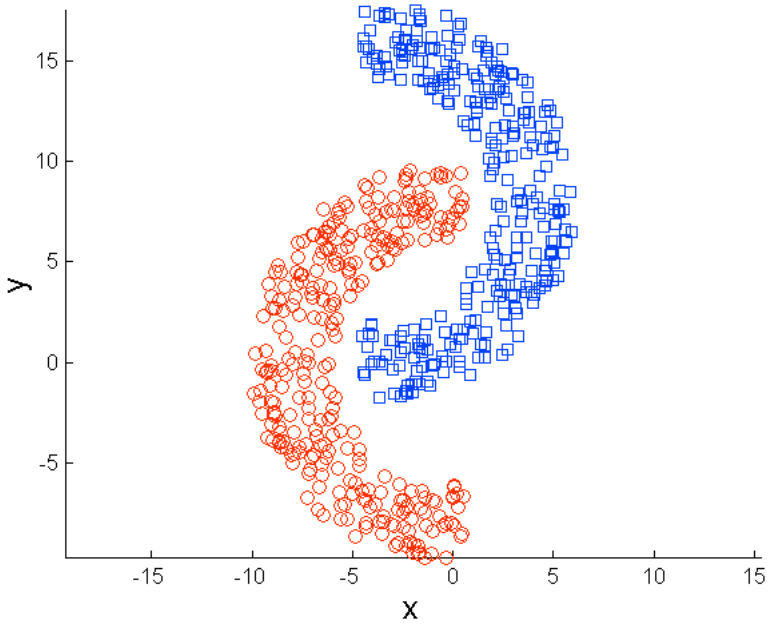


Eredeti pontok

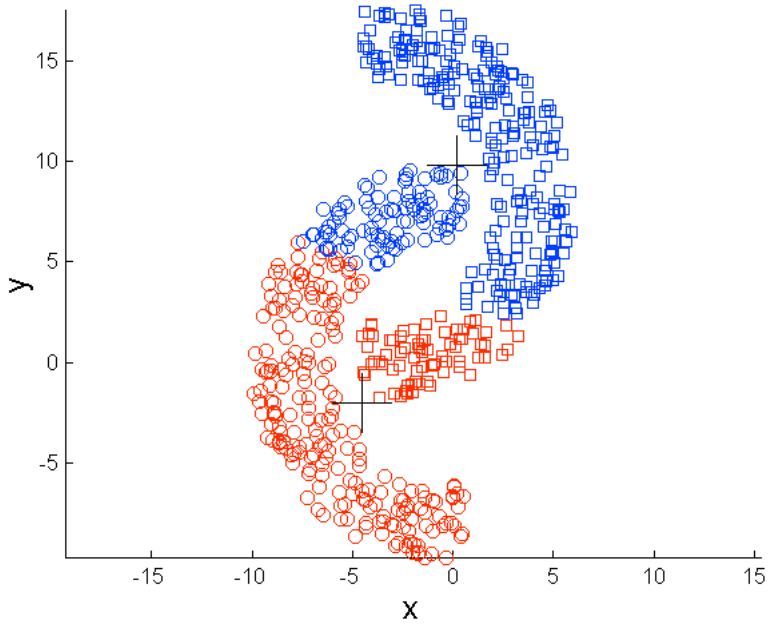


k-közép (3 klaszter)

k-közép módszer korlátai



Eredeti pontok

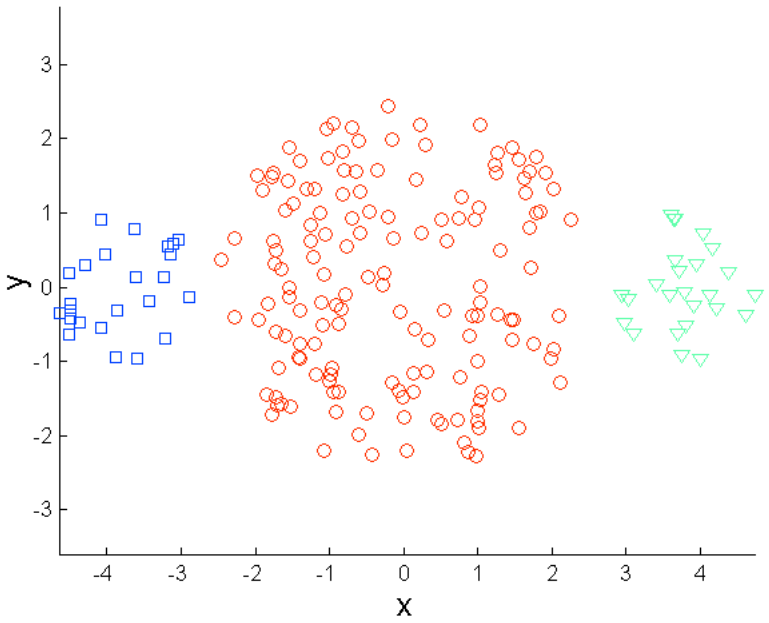


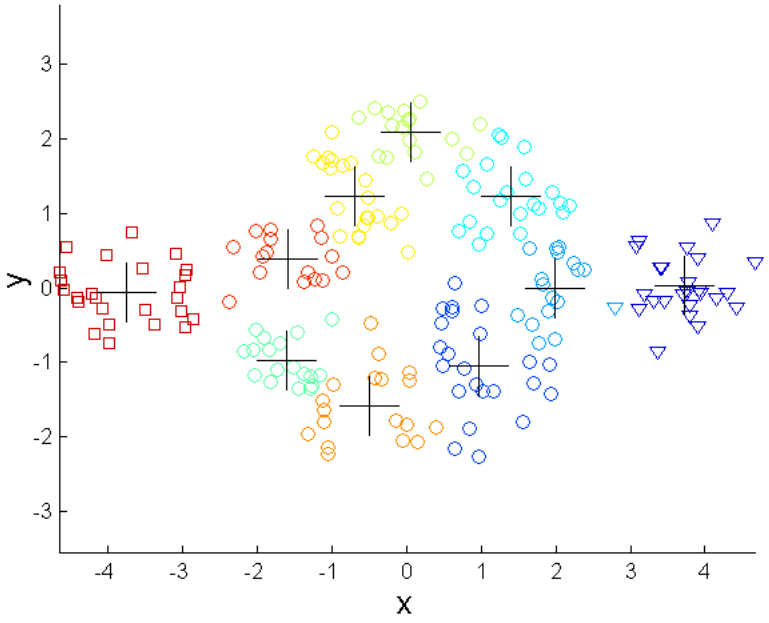
k-közép (2 klaszter)

Elő- és utófeldolgozás

- Előfeldolgozás
  - Normalizálás
  - Kiugró értékek kiszűrése
- Utófeldolgozás
  - Kis klaszterek kiszűrése (kiugró értékek?)
  - A laza klaszterek felosztása (nagy SSE értékek mellettiek)
  - Fésüljük össze a közeli, kis SSE-vel rendelkező klasztereket

A k-közép korlátainak feloldása



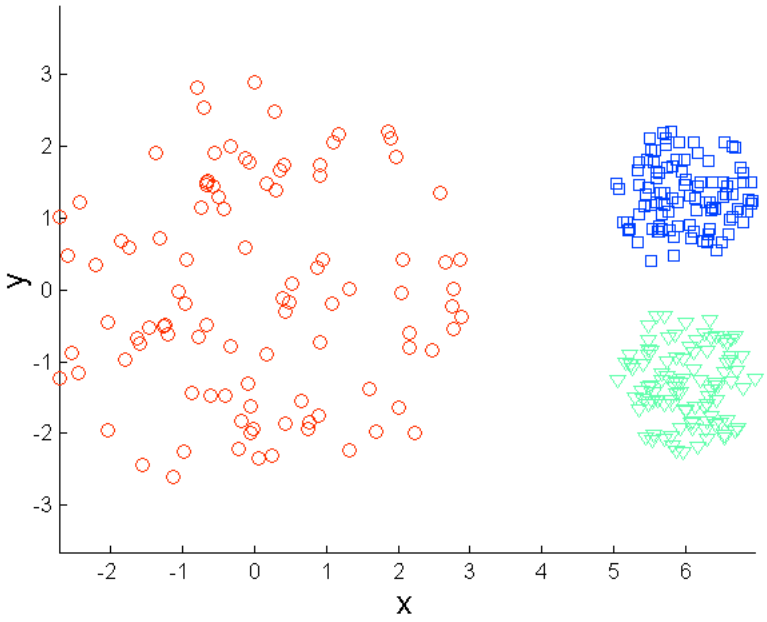


Eredeti pontok

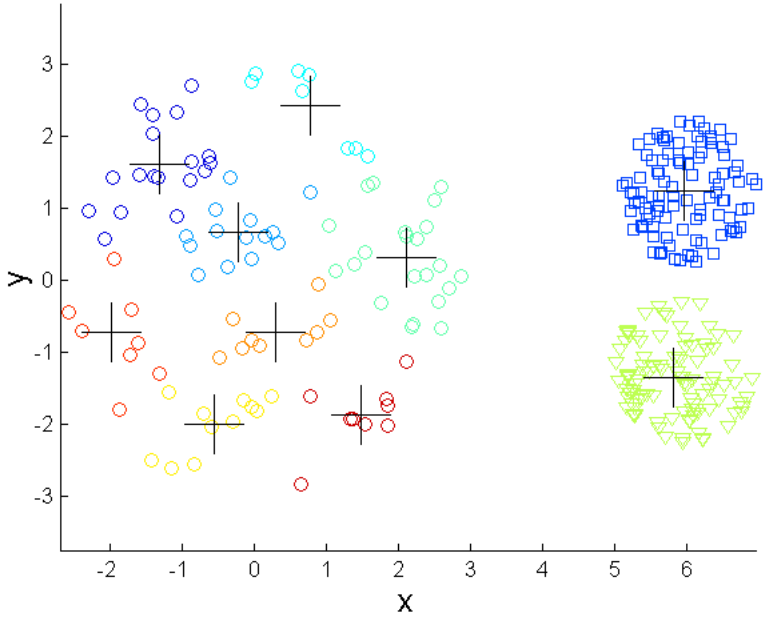
K-közép klaszterek

Egy megoldás: sok klaszter keresése, majd a végén össze kell vonni őket.

A k-közép korlátainak feloldása

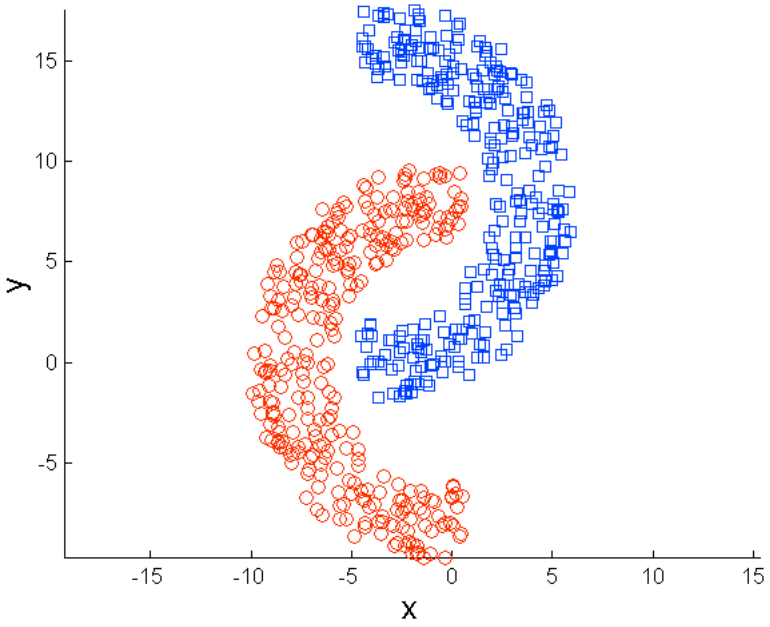


Eredeti pontok

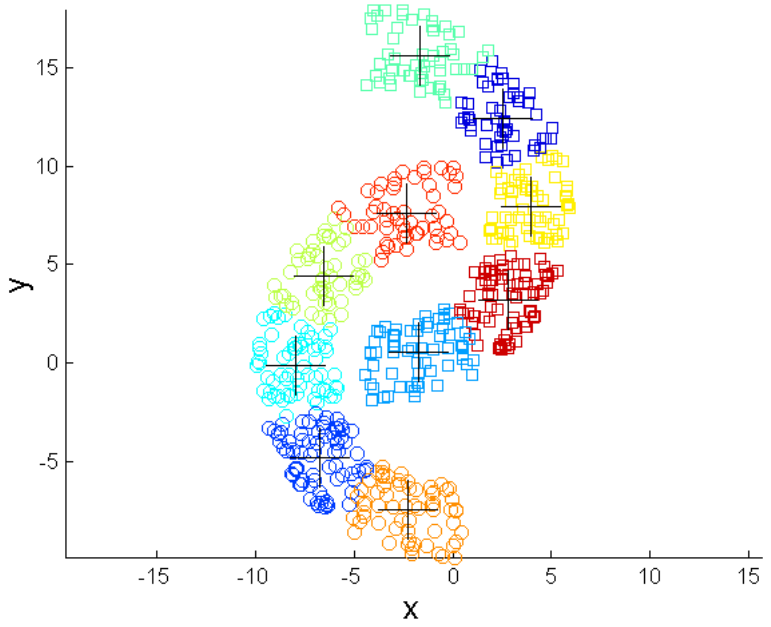


K-közép klaszterek

A k-közép korlátainak feloldása



Eredeti pontok



K-közép klaszterek

Klaszterezés (Szegedi Tudományegyetem) című tananyaga alapján készült (részben)