

Klaszterezés

A klaszterezés egy adathalmaz pontjainak, rekordjainak hasonlóság alapján való csoportosítása

- felügyelet nélküli(unsupervised), különben a feladat egy N osztályos klasszifikáció
- Bell-számnyi klaszterezés lehetséges egy n elemű halmaz esetében $O(e^{n \lg n})$
- épp ezért a klaszterezés egyik fő paramétere a klaszterek száma

Amennyiben a klaszterezés végeredménye diszjunkt csoportok hard klaszterezéséről beszélünk (pl. k-közép (k-means)). Szoft (vagy gyenge) klaszterezés esetében csak azt várjuk el, hogy minden (pont, klaszter) párra egy a klaszterba tartozásl függő mértéket rendeljünk.

Klaszterezés

Klaszterezési algoritmus kiválasztása előtt érdemes az adatot is megfigyelni:

1. **Jól szeparált** csoportok (a legtöbb módszer jó)
Minden egy csoportba tartozó elem közelebb van a csoport többi eleméhez mint bármilyen más csoportba tartozó elem
2. **Középpont** alapú csoportok (pl. kmeans)
Minden csoportnak meghatározhatunk egy középpontot, melyhez minden adott csoportba tartozó elem közelebb van, mint más csoportok középpontjaihoz
3. **Sűrűség** alapú csoportok (jó klaszterezés: pl. DBSCAN, OPTICS)
A csoportokat az adott terület sűrűsége határozza meg.
4. **Szomszédossági** vagy kapcsolati háló alapú csoportok
(jó klaszterezés: pl. single-link)
Egy adott pontból elérhető minden más a csoportba tartozó elem.
5. **Szabály** alapú csoportok (feladattól függ)

1. **Jól szeparált** csoportok (a legtöbb módszer jó)

Minden egy csoportba tartozó elem közelebb van a csoport többi eleméhez mint bármilyen más csoportba tartozó elem

2. **Középpont** alapú csoportok (pl. kmeans)

Minden csoportnak meghatározhatunk egy középpontot, melyhez minden adott csoportba tartozó elem közelebb van, mint más csoportok középpontjaihoz

3. **Sűrűség** alapú csoportok (jó klaszterezés: pl. DBSCAN, OPTICS)

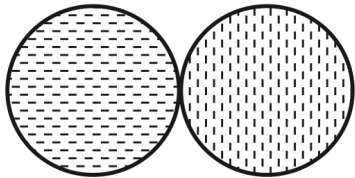
A csoportokat az adott terület sűrűsége határozza meg.

4. **Szomszédossági** vagy kapcsolati háló alapú csoportok

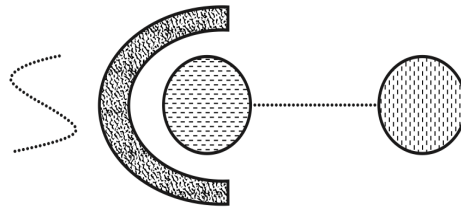
(jó klaszterezés: pl. single-link)

Egy adott pontból elérhető minden más a csoportba tartozó elem.

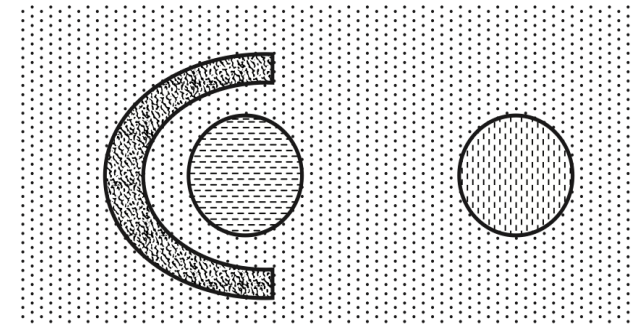
5. **Szabály** alapú csoportok (feladattól függ)



a) ?



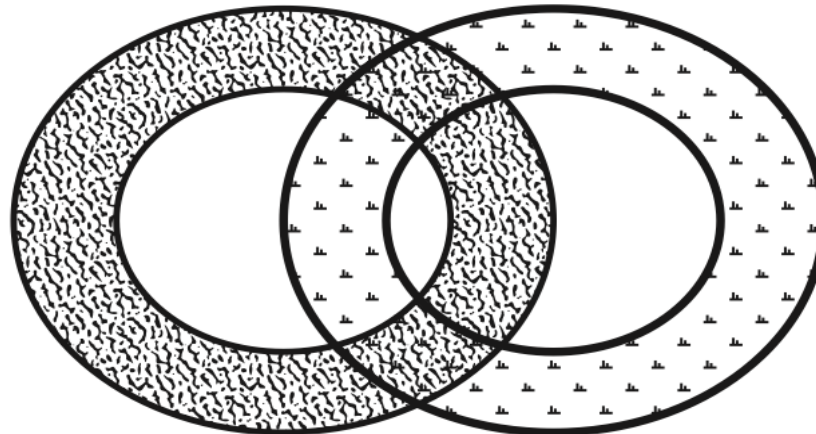
b) ?



c) ?



d) ?



e) ?

K-means

Feltesszük, hogy az adatpontjaink egy vektortérben helyezkednek el.

A klasztereket a **középpontjuk** (súlypontjuk) határozza meg.

K-means (D; k)

```
1  r(C1), r(C2),... , r(Ck) reprezentánsok tetszőleges k elemű kezdeti
   halmaza
2  while az r(Ci) reprezentánsok rendszere változik
3  do for i 1 to k
4      do r(Ci) = Ci-be tartozó elemek átlaga
5  for minden u ∈ D
6      do u legyen az  $\operatorname{argmin}_i d(u; r(C_i))$  indexű klaszterben
7  C az új klaszterezés
8  return C
```

$$Err_{squared}(D, k) = \sum_{i=1}^D (x_i - c_i)^2$$

A kezdőpontok meghatározása lehet:

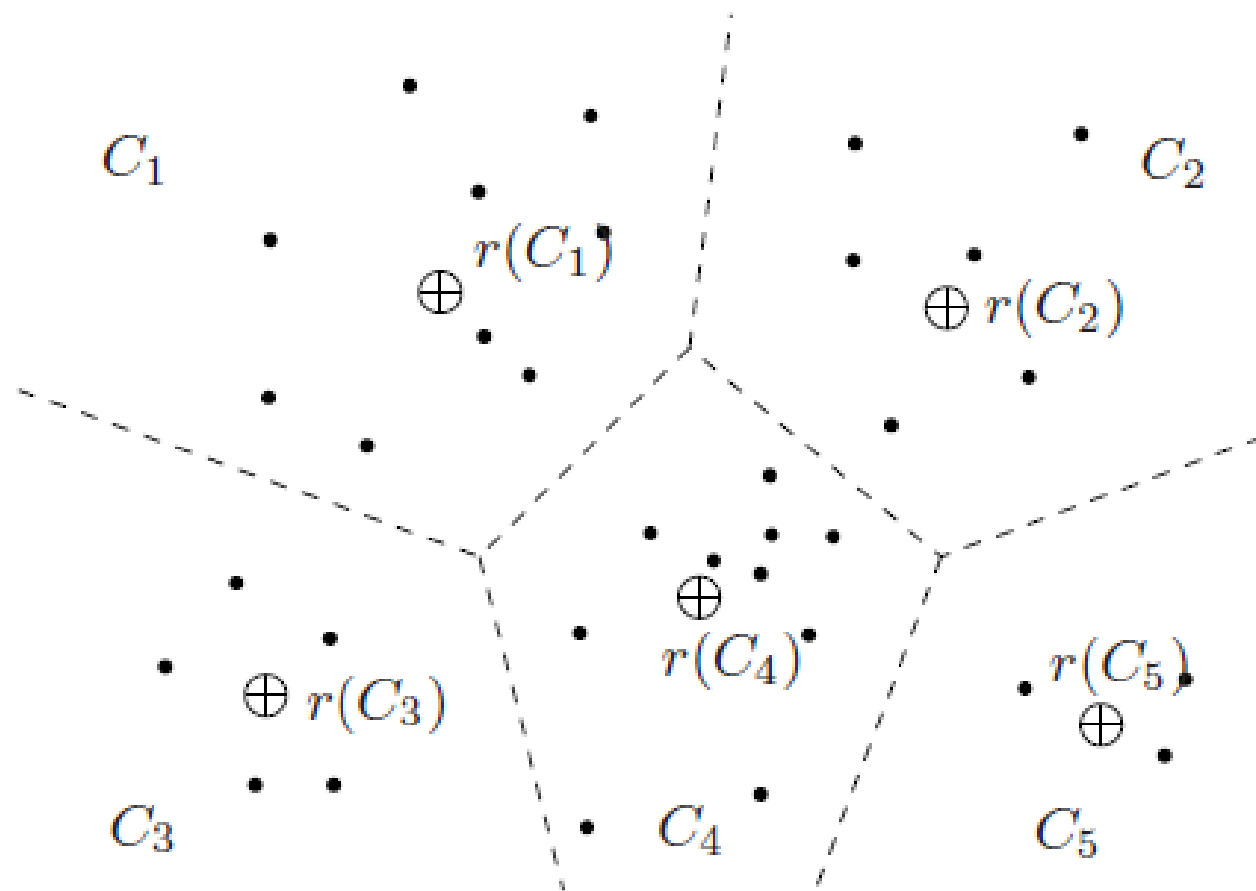
- a) véletlen pontokból
- b) véletlen választott tanulópontokból

Az algoritmust megállítjuk, ha:

- a) a klaszterezés nem változik
- b) a küszöbhiba meghaladja a négyzetes hiba mértékét
- c) elértük a maximális iterációk számát

K-means

Sajnos lokális minimumban is megállhat (gyakran)!



K-means

Előnye, hogy a futásidő $N \cdot K \cdot it$ ahol N a pontok, K a klaszterek, it pedig az iterációk száma

Zajos adatokon nem hatékony.

Csak “gömb”-szerű klasztereket képes megtalálni.

A fenti algoritmus esetében a négyzetes hiba konvergál

Lehet-e előre meghatározni K -t?

- mivel a legtöbb esetben véletlen klaszterekből indulunk ki, lehetséges több K -t kipróbálni, s a legjobb K -t választani a feladatainknak megfelelően
- kifinomultabb, ha kiindulunk egy maximális K - klaszterből, majd finomítjuk összevonásokkal
- vagy fordítva, elindulunk egy klaszterből s folyamatosan új középpontokat határozunk meg amennyiben az új struktúra jobbnak bizonyul mint az előző

Mindezt egy maximális K klaszterszámig számoljuk, vagy megállunk, ha nem érdemes tovább bontani a klasztereket.

K-medoid

Amennyiben a klaszterközéppontok a tanulóhalmaz pontjaiból kerülnek ki k-medoid algoritmust használunk:

Medoid körüli particionálás:

Adott: K , N elemű X tanulóhalmaz

1. K véletlen középpont kiválasztása
2. Minden pontot a hozzá legközelebbi klaszterba soroljuk , kiszámoljuk a középponttól vett osztávolságot (ez a költség)
3. Minden klaszter és nem a klaszterba tartozó pont párra kiszámoljuk mi lenne a költség, ha kicserélnénk őket
4. Az új medoidok legyenek azok melyeknél a költség a legkisebb volt
5. Ismételjük amíg vagy nem változik a konfiguráció vagy elértük a maximális iterációk számát

Kezdőklaszterek kiszámítása k-meansnél?

Sűrűség alapú módszerek

A K-means egyik legnagyobb hibája, hogy akkor is K klasztert fog létrehozni ha nincs rá szükség.

Olyan adatokon, melyek struktúrája sűrűség típusú, általában Hasztalan.

Ezekben az esetekben jól látható hogy az X-means sem lesz megfelelő.

Nem egy adott középponttól vett távolságra van szükség, hanem strukturális modellre.

Ilyen esetben használható pl. [single-link](#), [DBSCAN](#), [OPTICS](#)



Sűrűség alapú módszerek

DBSCAN

DBSCAN: Density-Based Spatial Clustering of Applications with Noise

A pontok halmaza P
Szomszédossági reláció:

$N_{\epsilon} = (\{(p, q) \in P \times P \mid \text{sim}(p, q) > \epsilon\})$ vagy $N_{\epsilon}(p) = (\{(p, q) \in P \times P \mid \text{dist}(p, q) < \epsilon\})$

Azon pontok halmaza melyek p -ből elérhetőek ϵ távolságon belül, vagy a hasonlóságuk nagyobb mint ϵ :

p szomszédjai: $N_{\epsilon}(p) = \{q \in P \mid q \in N_{\epsilon}\}$

p sűrűsége: $|N_{\epsilon}(p)|$

Sűrűség alapú módszerek

DBSCAN

DBSCAN:

1. p magpont amennyiben $|N_{\epsilon}(p)| \geq \text{MinPts}$
2. a C klaszter azon pontjai melyek nem magpontok , a klaszter határ vagy keretpontjai

A p pont közvetlenül elérhető a q pontból, ha

1. p q szomszédja és
2. q magpont

A p pont elérhető q pontból, ha

Létezik olyan $q=p_1, p_2, \dots, p_n, p_{n+1}=p$ sorozat, ahol minden i -re p_{i+1} közvetlenül elérhető p_i -ből

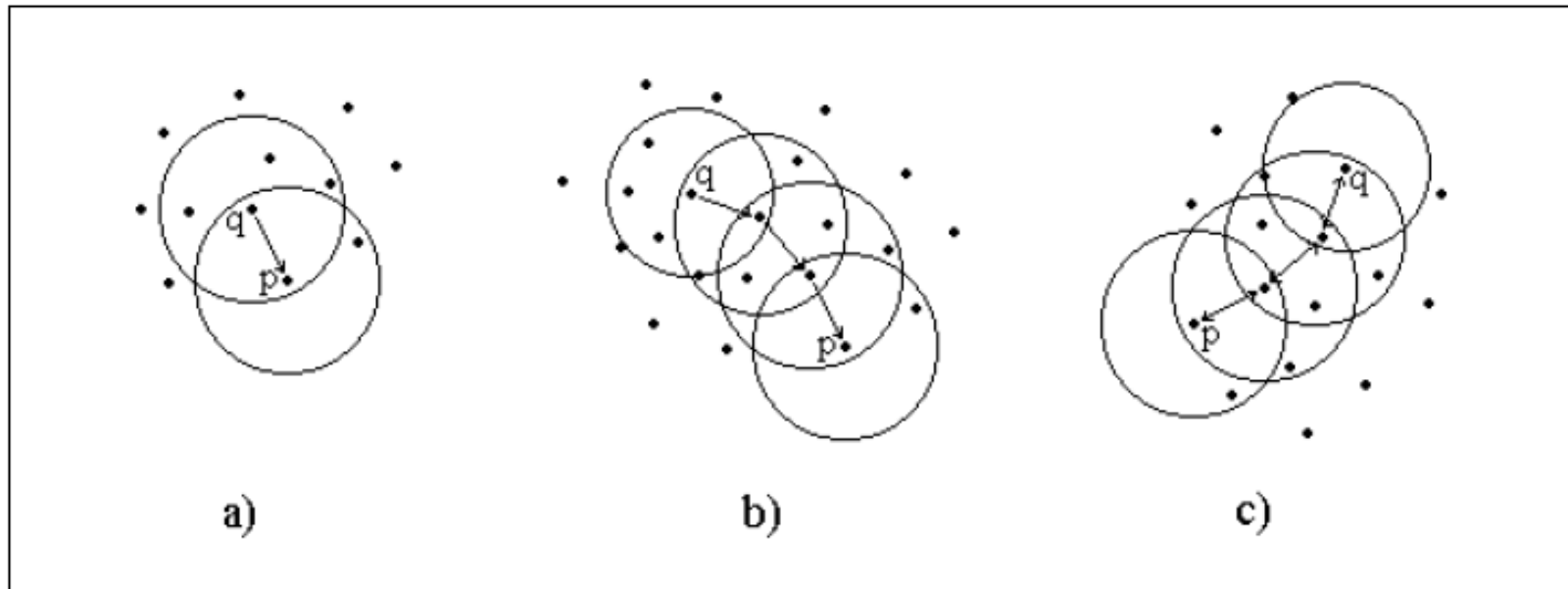
A p, q összekapcsoltak, ha létezik $r \in P$, s p és q is elérhető r -ből

Sűrűség alapú módszerek

DBSCAN

Ha C részhalmaza P -nek és C nem üres halmaz akkor klaszter amennyiben:

1. (összefüggőségi feltétel) minden $p, q \in C$ összekapcsolt
2. (maximálisság) minden $p \in P$ és $q \in C$, ha p elérhető q -ből akkor $p \in C$



Példa: $\text{MinPts} = 4$

- a) p közvetlenül elérhető q -ből (q magpont, p határpont)
- b) p elérhető q -ből (q magpont, p határpont)
- c) p és q összekapcsolt (mindkettő határpont)

Kapcsolati háló alapú klaszterezés

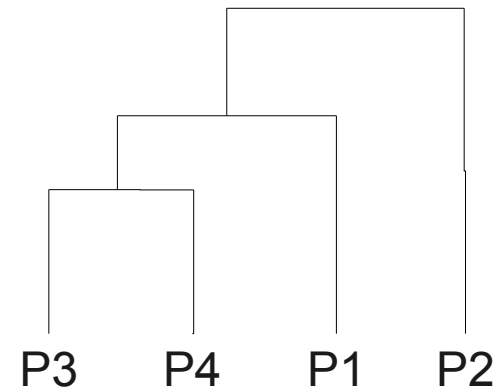
Két klaszter hasonlósága

- a) leghasonlóbb elemeiknek a hasonlósága (single-link)
- b) legkevésbé hasonló elempárjaik hasonlósága (complete-link)
- c) elemeik átlagos hasonlóság (average-link)

1. Minden pont meghatároz egy különálló klasztert
2. Keressük meg a leghasonlóbb klaszter-párt s egyesítsük őket
3. Ismételjük a második lépést amíg van legalább két klaszterünk

Ábrázolás: dendrogram (ábra: single-link)

Hasonlóság	P1	P2	P3	P4
P1	1	0.5	0.2	0.7
P2	0.5	1	0.4	0.6
P3	0.2	0.4	1	0.8
P4	0.7	0.6	0.8	1



Órai feladat 1:

Rajzoljunk dendogram-ot a következő példára!

- a) single-link
- b) complete-link
- c) average-link

Hasonlóság	P1	P2	P3	P4
P1	1	0.8	0.2	0.7
P2	0.8	1	0.9	0.2
P3	0.2	0.9	1	0.1
P4	0.7	0.2	0.1	1

Sűrűség alapú módszerek

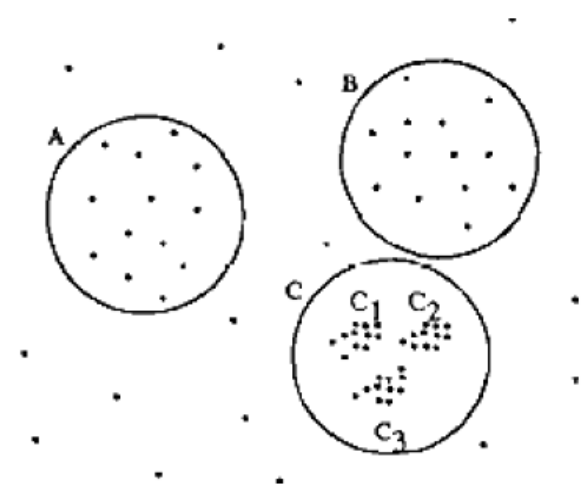
OPTICS

A DBSCAN feltételezi, hogy a sűrűség az egész adatra hasonló. (globális sűrűség)

OPTICS: Ordering Points to Identify the Clustering Structure (M. Ankerst, M. Breunig, H. Krieger, J. Sander)

Az algoritmus rendezi az adatpontokat, a rendezés megjelenítéséből azonosíthatóvá válnak a klaszterek. (nem klaszterező algoritmus, klaszterezés előkészítő)

Az algoritmusnak az adatpontokon kívül a egy előre definiált eps generáló távolságra illetve egy MinPts korlátra van szüksége



Sűrűség alapú módszerek

OPTICS

$N(p)$ a p eps sugarú környezete: $N(p) = \{q \in P \mid d(p,q) < \text{eps}\}$

$p \in P$ magpont, ha $N(p) \geq \text{MinPts}$

$p \in P$ határpont, ha $N(p) < \text{MinPts}$

$p \in P$ közvetlenül elérhető $q \in P$ pontból, ha

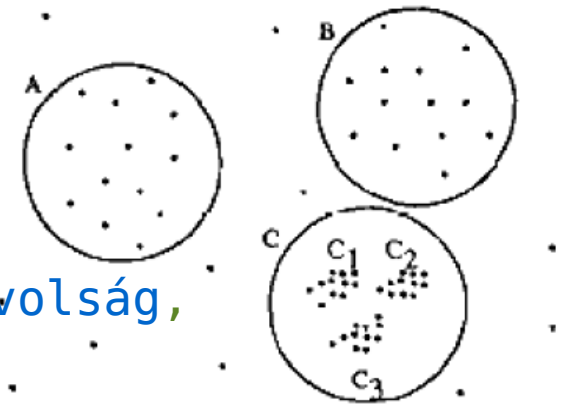
1. $p \in N(q)$ és
2. q magpont

Legyen egy $p \in P$ pont k -távolsága az a $d(p,q)$ távolság, melyre igaz:

1. legalább k olyan $r \in P \setminus \{p\}$ pont van, hogy $d(p,r) \leq d(p,q)$
2. legfeljebb $k-1$ olyan $r \in P \setminus \{p\}$ pont van, melyre $d(p,r) < d(p,q)$

$p \in P$ magpont magtávolsága megegyezik MinPts ($k=\text{MinPts}$) távolságával p -nek

$p \in P$ elérhető távolsága $o \in P$ magponttól $\max(\text{magtávolság}(o), d(p,o))$

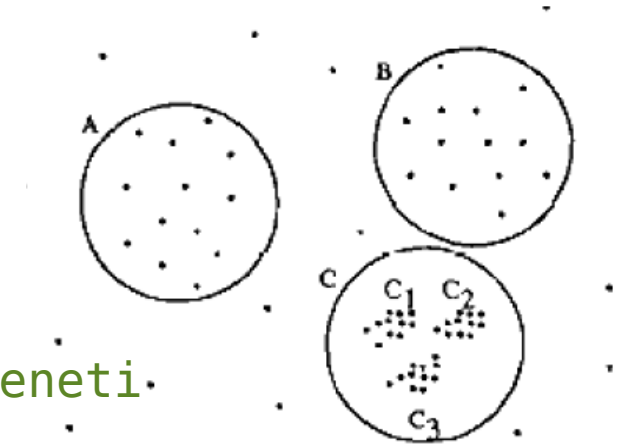


Sűrűség alapú módszerek

OPTICS

OPTICS Algoritmus:

1. vegyünk egy eddig nem vizsgált elemet
2. ha az elem nem magpont, akkor berakjuk a kimeneti halmazba
3. ha az elem magpont, akkor a szomszédai bekerülnek a bővítési halmazba. Maga a pont a kimeneti halmazba kerül
4. meghatározzuk minden bővítési halmaz elemének a kimeneti halmaztól vett legkisebb távolságát, majd rendezzük a bővítési halmazt e szerint
5. a bővítési halmaz első elemét átrakja a kimeneti halmazba, majd a szomszédáival kiegészítjük a bővítési halmazt
6. ha a bővítési halmaz kiürült, a 4- pontba lépünk, egyébként az elsőbe

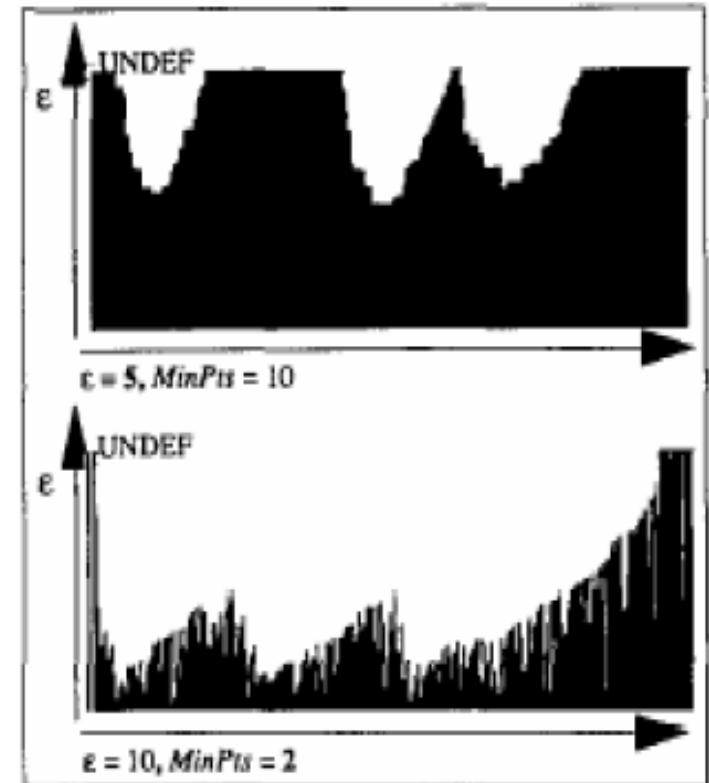
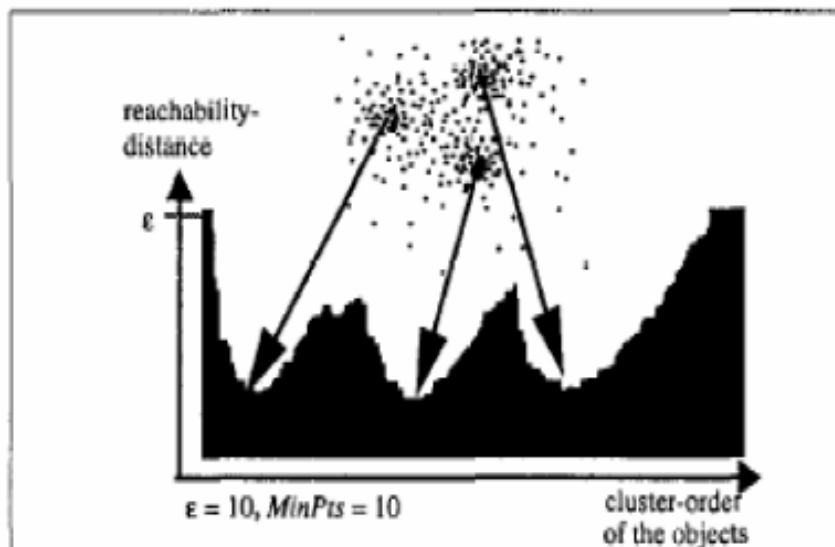


Sűrűség alapú módszerek

OPTICS

OPTICS kimenete

A gödrök jelentenék a klasztereket
Az epsilon-tól erősen függ a kimenet, de a
MinPts-től nem



Klaszterezés weka-val

Próbáljuk ki a következő adatokon:

testdata1.arff : 2700 pont, 2D -> 2 valós klaszter (Hi)

testdata2.arff : 900, 2D -> 9 valós klaszter

Miert nem tudunk 100-as csoportokat kialakítani?

Indítás: véletlenül választott pontokból vagy véletlen pontokból?

Sokszor kell lefuttatni, s a legjobbat választani?

Mi a legjobb? Legkisebb átlagos távolság , kiegyensúlyozott pontthalmazok?

Mi legyen a K? Euklideszi távolság? (Minkowski : L1, L2 vagy esetleg más)

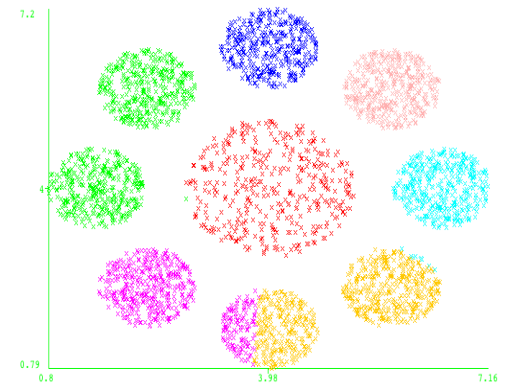
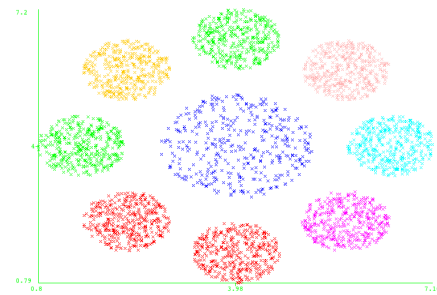
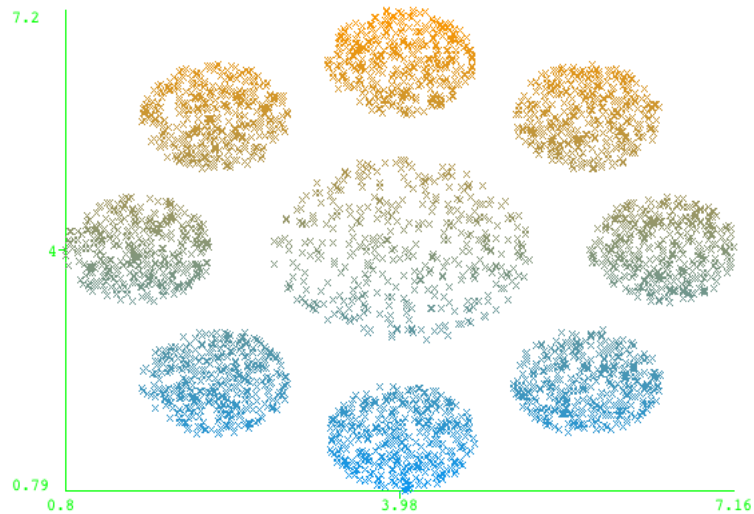
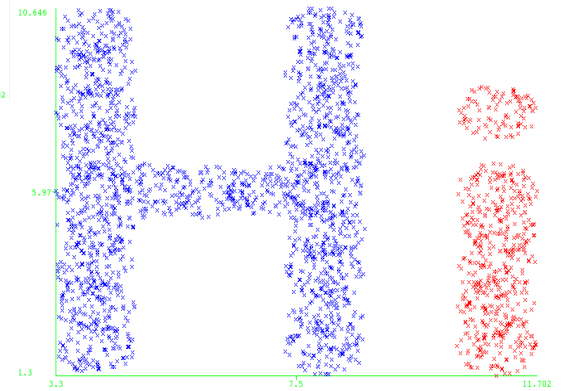
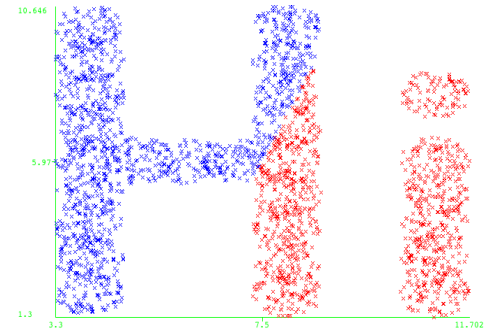
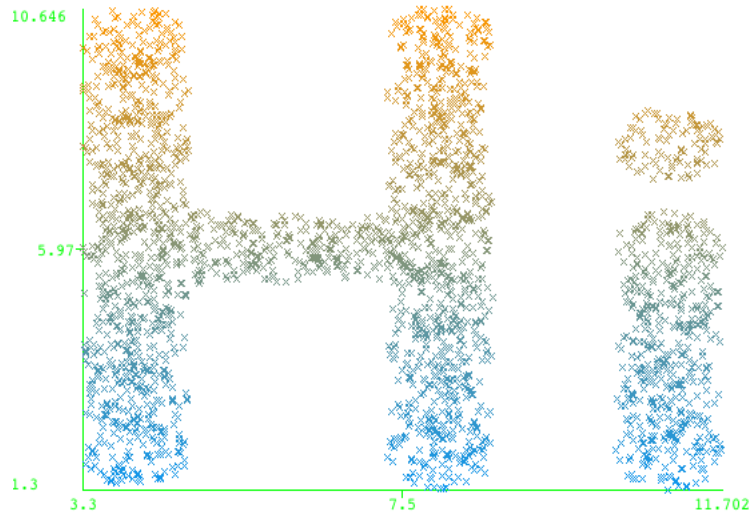
Mi legyen a kiürült halmazokkal?

Normalizálni kell e a tanulóhalmazt?

$$\left(\sum |x_i - y_i|^p\right)^{1/p}$$

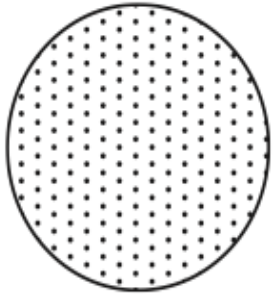
Órai feladat 2: Próbáljuk ki a DBSCAN, OPTICS-ot és a k-meanst WEKA-ban! (adatb4.zip)

Klaszterezés weka-val

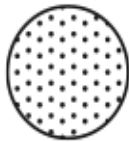


Órai feladat 3:

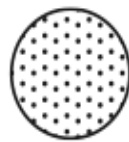
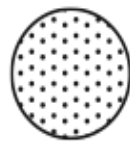
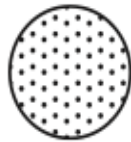
Mit tesz egy sűrűség, egy center és egy kapcsolat alapú klaszterező algoritmus?
(feltételezzük, hogy a pontok sűrűsége a mintában állandó)



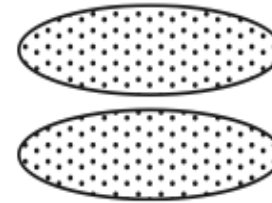
a)



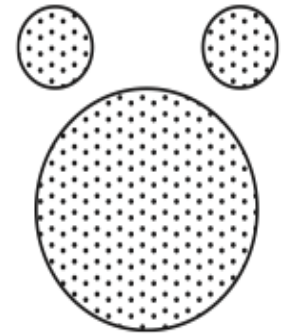
b)



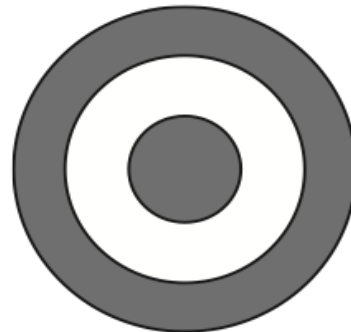
c)



d)



e)



f)

Klaszterezési eljárások

Klaszterezési eljárások jóságára függően az ismert adatoktól több lehetőség is adódik:

- a) amennyiben nem ismerjük az adat eredeti elrendezését, az adott metódushoz igazodva tudunk hibafüggvényt meghatározni:
 - K-means : négyzetes hiba a klaszterközepontoktól
 - DBSCAN: outlierok valószínűsége vagy az összekapcsolás szórása
- b) ismert valamely adathalmaz klaszterezése
 - osztályozásból ismert mértékek: f-measure, accuracy stb.
 - tisztaság: minden klaszterhez hozzárendeljük a hozzá csoportosított pontok közül a leggyakrabban előforduló referencia osztályt. Majd megszámoljuk mennyire jellemzi az adott klaszter az adott osztály pontokra átlagosan.

pl. 1-es klaszter:

1-es osztály: 1, 2-es osztály: 4, 3-as osztály: 2
tehát a legjellemzőbb osztály a 2-es,

2-es klaszter: 1-es osztály: 4, 2-es osztály: 3, 3-as osztály: 2

tehát a legjellemzőbb osztály az 1-es

$$\text{Purity} = (4+4) / (1+4+2+4+3+2) = 0.5$$

Sajnos a purity növekszik nagyobb klaszterszám mellett....

Klaszterezési eljárások

Kölcsönös információ (mutual information):

$$MI(K, C) = \sum_k \sum_j p_{kj} \log \frac{p_{kj}}{p_k p_j}$$

Ahol p_{kj} annak a valószínűsége, hogy egy pont a k -dik klaszterhez és j -dik osztályba tartozik, p_j a j -dik osztály valószínűsége, p_k pedig a k -dik klaszterbe tartozás valószínűsége.

A tisztasághoz hasonlóan a maximumát akkor is felveszi, ha minden pontot egy különálló csoportba sorolunk.

Ennek elkerülésére normalizáljuk az entrópia segítségével:

$$H(K) = \sum -p_k \log p_k$$

$$H(C) = \sum -p_c \log p_c$$

$$NMI(K, C) = \frac{\sum_k \sum_j p_{kj} \log \frac{p_{kj}}{p_k p_j}}{\frac{H(C) + H(K)}{2}}$$

Órai feladat 4:

Vegyük a következő klaszterezést és eredeti osztálybesorolást: Számítsuk ki a tisztaság, kölcsönös információ és a normalizált kölcsönös információ értékét! Van olyan tanult klaszterezési eljárás ami ennél jobban teljesítene?

