

Klaszterezés

Lukács András

alukacs@sztaki.hu

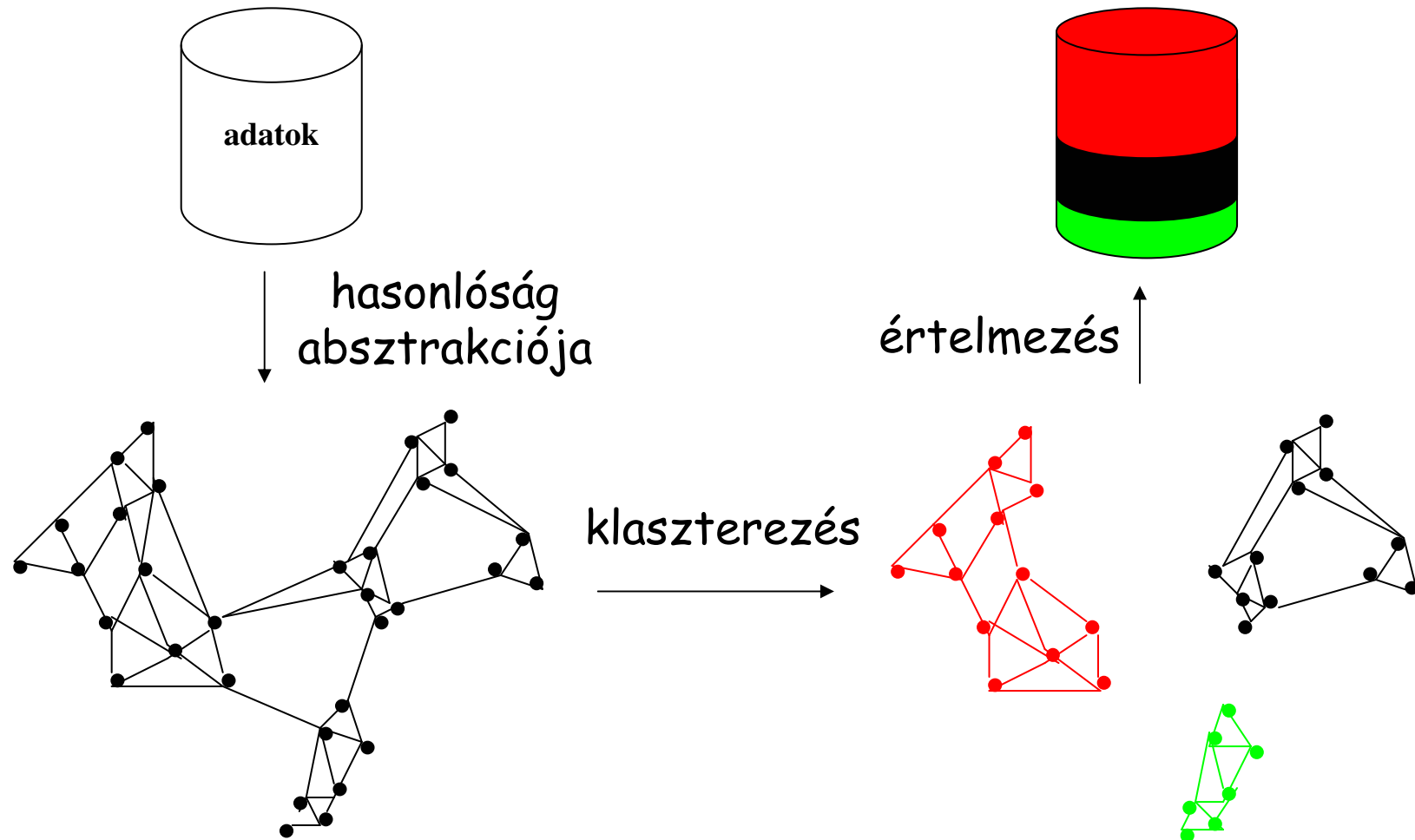
<http://www.sztaki.hu/~alukacs>

<http://www.ilab.sztaki.hu/websearch>

Témák

- k-közép, k-medoids
- DBSCAN
- OPTICS
- ROCK
- Chameleon

Klaszterezés



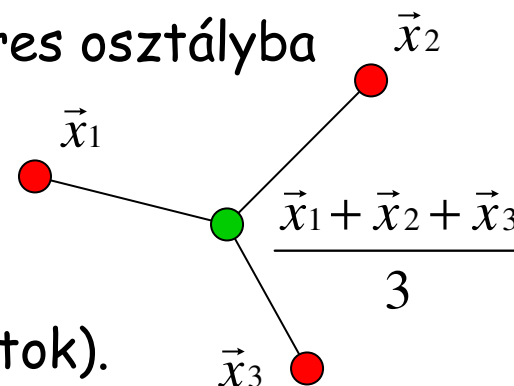
A k-közép (k-means) klaszterező algoritmus

Adott véges sok pont egy valós lineáris térben.
A klaszterek k száma előre adott.

A k-közép algoritmus lépései:

0) Inicializálás: a pontokat k darab nemüres osztályba particionáljuk.

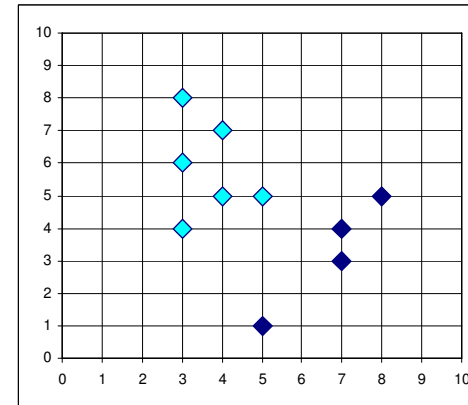
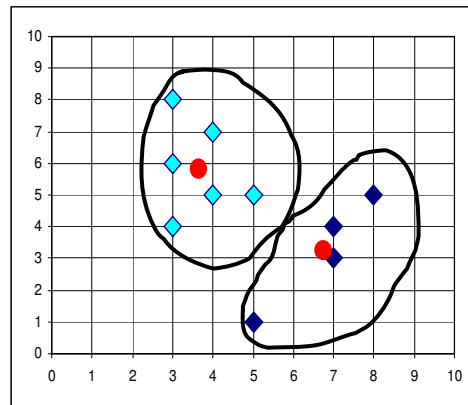
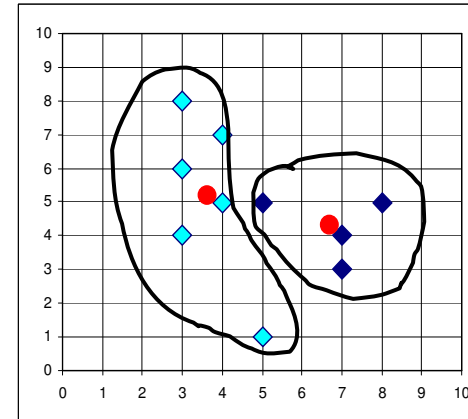
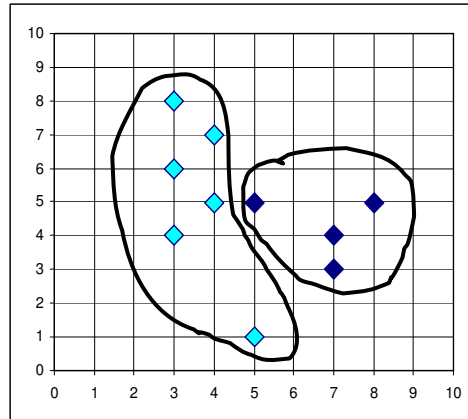
1) Kiszámoljuk a pillanatnyi partíció osztályainak **középpontjait** (baricentrikus középpontok).



2) Hozzárendeljük mindegyik pontot a hozzá legközelebbi középponthez. Így kapunk egy partíciót k osztállyal.

3) Visszamegyünk az 1) lépéshez, ha változott a partíció, ha nem változott, megállunk.

Példa a k-közép algoritmusra R^2 -ben



A k-közép algoritmus előnyei és hátrányai

Előnyök:

- $O(tkn)$ futási idő, ahol n a pontok száma, k az osztályok száma és t az iterációs lépések száma. Tipikusan $k, t \ll n$.
- Egyszerű implementáció.

Hátrányok:

- Csak akkor alkalmazható, ha tudunk középpontokat számolni. Nem működik kategorikus adatokra.
- Gyakran lokális optimumban áll meg az algoritmus. Globális optimum megtalálásához további módszerek kellenek.
- Az osztályok k számát előre meg kell adni.
- Nem kezeli jól a zajos adatokat és a magányos pontokat.
- Mindig konvex osztályokat igyekszik megtalálni (még akkor is ha nem konvexek az optimálisak).

Sűrűség, szomszédsági gráf

Szomszédsági reláció:

$$N_{\Theta} = \{(p, q) \in P \times P \mid \text{sim}(p, q) \geq \Theta\}, \text{ vagy}$$

$$N_{\Theta} = \{(p, q) \in P \times P \mid \text{dist}(p, q) \leq \Theta\}$$

attól függően, hogy hasonlóság-, vagy távolságfüggvényt használunk. Θ tetszőleges, általunk választott hasonlósági, vagy távolsági korlát.

A $p \in P$ pont szomszédainak halmaza:

$$N_{\Theta}(p) = \{q \in P \mid (p, q) \in N_{\Theta}\}.$$

A $p \in P$ pont sűrűsége:

$$|N_{\Theta}(p)|.$$

Szomszédsági gráf, $NG = (P, E)$:

A $p, q \in P$ csúcsok között pontosan akkor vezet él, ha $(p, q) \in N_{\Theta}$ fennáll.

Hasonlóság, távolság

A kosarak közötti hasonlóságra **Jaccard együttható**:

$$\text{sim}(k_1, k_2) = \frac{|k_1 \cap k_2|}{|k_1 \cup k_2|}.$$

Segítségével könnyen definiálhatunk a kosár típusra értelmezett **távolságfüggvényt**:

$$\text{dist}(k_1, k_2) = 1 - \text{sim}(k_1, k_2) = 1 - \frac{|k_1 \cap k_2|}{|k_1 \cup k_2|} = \frac{|k_1 \cup k_2|}{|k_1 \cup k_2|} - \frac{|k_1 \cap k_2|}{|k_1 \cup k_2|} = \frac{|k_1 \cup k_2| - |k_1 \cap k_2|}{|k_1 \cup k_2|}$$

$$\text{dist}(k_1, k_2) = \frac{|k_1 \Delta k_2|}{|k_1 \cup k_2|}.$$

Tétel:

Legyenek A, B, C a T elemhalmaz tetszőleges kosarai. Ekkor fennáll, hogy $\text{dist}(A, B) \leq \text{dist}(A, C) + \text{dist}(B, C)$.

DBSCAN

- 1.) $p \in P$ **magpont**, ha $|N_{\Theta}(p)| \geq MinPts$.
- 2.) a C klaszter azon pontjai, melyek nem magpontok, tehát $|N_{\Theta}(p)| < MinPts$ a klaszter **határ**-, vagy **keretpontjai**.

MinPts alkalmasan választott sűrűségi korlát.

A $p \in P$ pont **közvetlenül elérhető** (*directly density-reachable*) a $q \in P$ pontból, ha

- 1.) p a q szomszédja ($p \in N_{\Theta}(q)$),
- 2.) q magpont ($|N_{\Theta}(q)| \geq MinPts$).

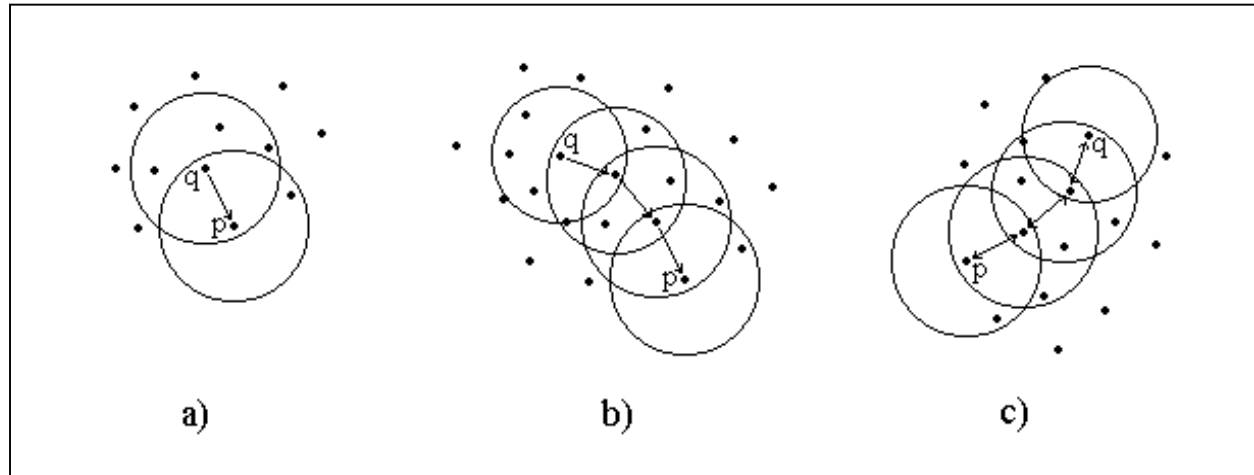
A $p \in P$ pont **elérhető** (*density-reachable*) a $q \in P$ pontból, ha

$\exists q = p_1, p_2, \dots, p_n, p_{n+1} = p$ pontsorozat, hogy $\forall i = 1, 2, \dots, n$ -re p_{i+1} közvetlenül elérhető p_i -ből.

A $p, q \in P$ pontok **összekapcsoltak** (*density-connected*), ha

$\exists r \in P$ pont, hogy p és q is elérhető r -ből.

DBSCAN



Példa: 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)

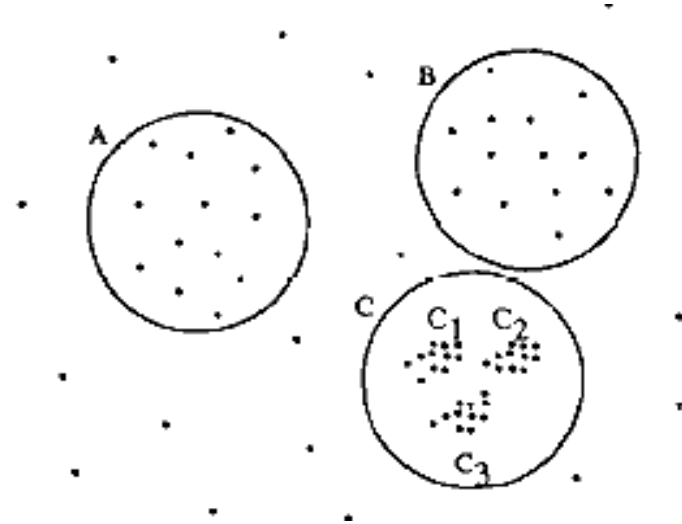
A $C \subseteq P$, $C \neq \emptyset$ halmaz **klaszter**, ha kielégíti a következő feltételeket:

- (1.) (összefüggőség) $\forall p, q \in C$ pont összekapcsolt egymással.
- (2.) (maximálisság) $\forall p \in P, q \in C$ pontra, ha p elérhető q -ből, akkor $p \in C$.

OPTICS

M. Ankerst, M. M. Breunig, HP. Kriegel,
J. Sander, *Ordering Points To Identify
the Clustering Structure*, SIGMOD-99

Probléma: a valós adathalmazoknál általában nem teljesül az a feltétel, hogy egy globális sűrűség-paraméterrel leírható lenne.



Az OPTICS algoritmus csak előkészíti a klaszterezést, rendezi az adathalmaz pontjait, a kapott rendezést megjelenítve a klaszter-struktúra azonosítható.

Input: ϵ generáló távolság, **MinPts** sűrűségi korlát.

Output: a pontok lineáris rendezése (releváns információ a klaszter-szerkezetről).

Magpontok

Egy pont ε -sugarú környezete: $N_\varepsilon(p) = \{q \in D \mid d(p,q) \leq \varepsilon\}$

Egy $p \in P$ pont **magpont**, ha $|N_\varepsilon(q)| \geq \text{MinPts}$

Egy $p \in P$ pont **határpont**, ha $|N_\varepsilon(q)| < \text{MinPts}$

A $p \in P$ pont **közvetlenül elérhető** a $q \in P$ pontból, ha

1) $p \in N_\varepsilon(q)$ és

2) $|N_\varepsilon(q)| \geq \text{MinPts}$ (magpont feltétel).

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

(i) legalább k olyan $r \in P \setminus \{p\}$ pont van, melyre $d(p,r) \leq d(p,q)$

(ii) legfeljebb $k-1$ olyan $r \in P \setminus \{p\}$ pont van, melyre $d(p,r) < d(p,q)$

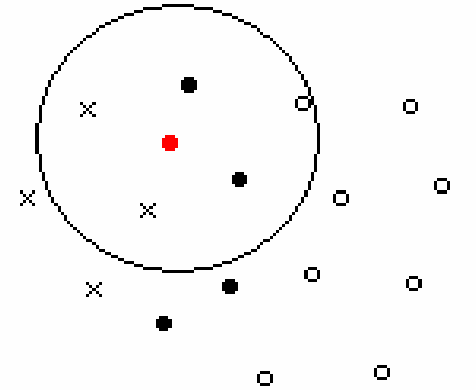
A $p \in P$ magpont **magtávolsága** a MinPts távolsága p -nek.

A $p \in P$ pont **elérhető-távolsága** $o \in P$ magponttól:

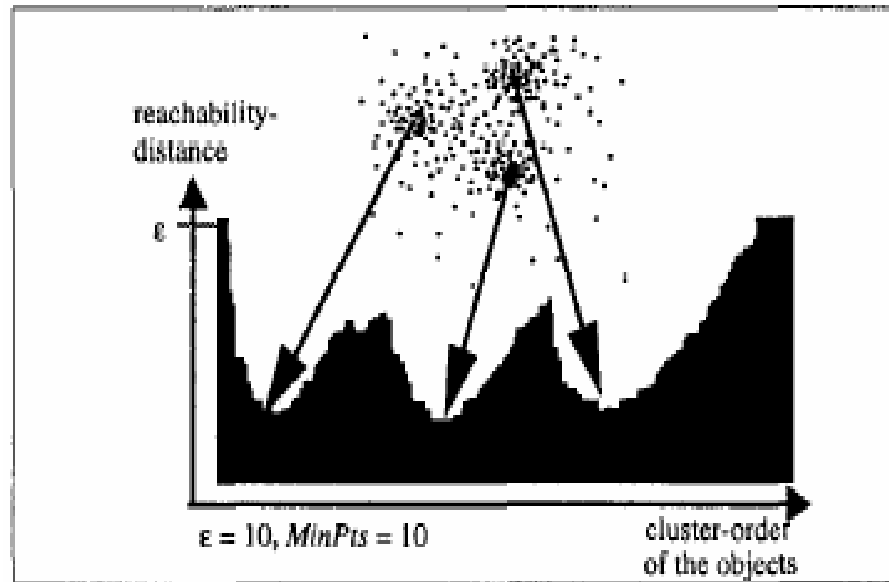
$$\max(\text{magtávolság}(o), d(p,o))$$

Az OPTICS algoritmus

1. Vesz egy elemet a még **nem vizsgált pontok** halmazából.
2. Ha az elem **nem magpont**, akkor berakja az **output halmazba**.
3. Ha az elem **magpont**, akkor egy **bővítési halmazba** rakja a pont szomszédait. Magát a pontot pedig berakja az **output halmazba**.
4. A bővítési halmaz minden elemére meghatározza (vagy felülírja) az output halmaztól mért legkisebb távolságot. E távolság alapján rendezi a bővítési halmazt.
5. Kiválasztja a bővítési halmaz legelső elemét, berakja az **output halmazba**, majd a szomszédaival bővíti a bővítési halmazt.
6. Vissza a 4-re, ha a bővítési halmaz kiürült, ha nem vissza 1-re.



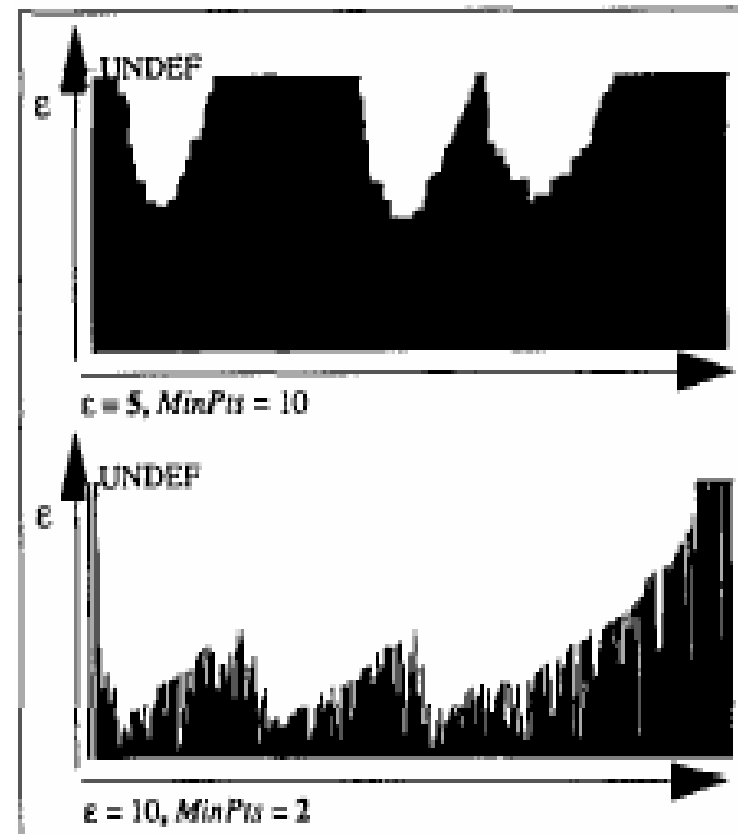
Az OPTICS outputja és értelmezése



- a "gödrök" jelzik a klasztereket
- automatizált klaszterkinyerés a deriváltak alapján lehetséges

A paraméterek hatása az outputra:

- nem érzékeny az eredmény a **MinPts**-ra
- az ϵ növelésével felderíthetők a klaszterek finomszerkezete, de lassabban fut le az algoritmus

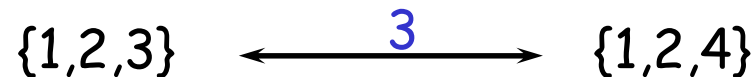


ROCK

Alapötletek:

- szomszédok számosságán alapuló hasonlóságfüggvény (Jaccard együttható)
- **link** fogalma: két pont közös szomszédainak száma

Példa: $\{1,2,3\}, \{1,2,4\}, \{1,2,5\}, \{1,3,4\}, \{1,3,5\}, \{1,4,5\}, \{2,3,4\}, \{2,3,5\}, \{2,4,5\}, \{3,4,5\}$



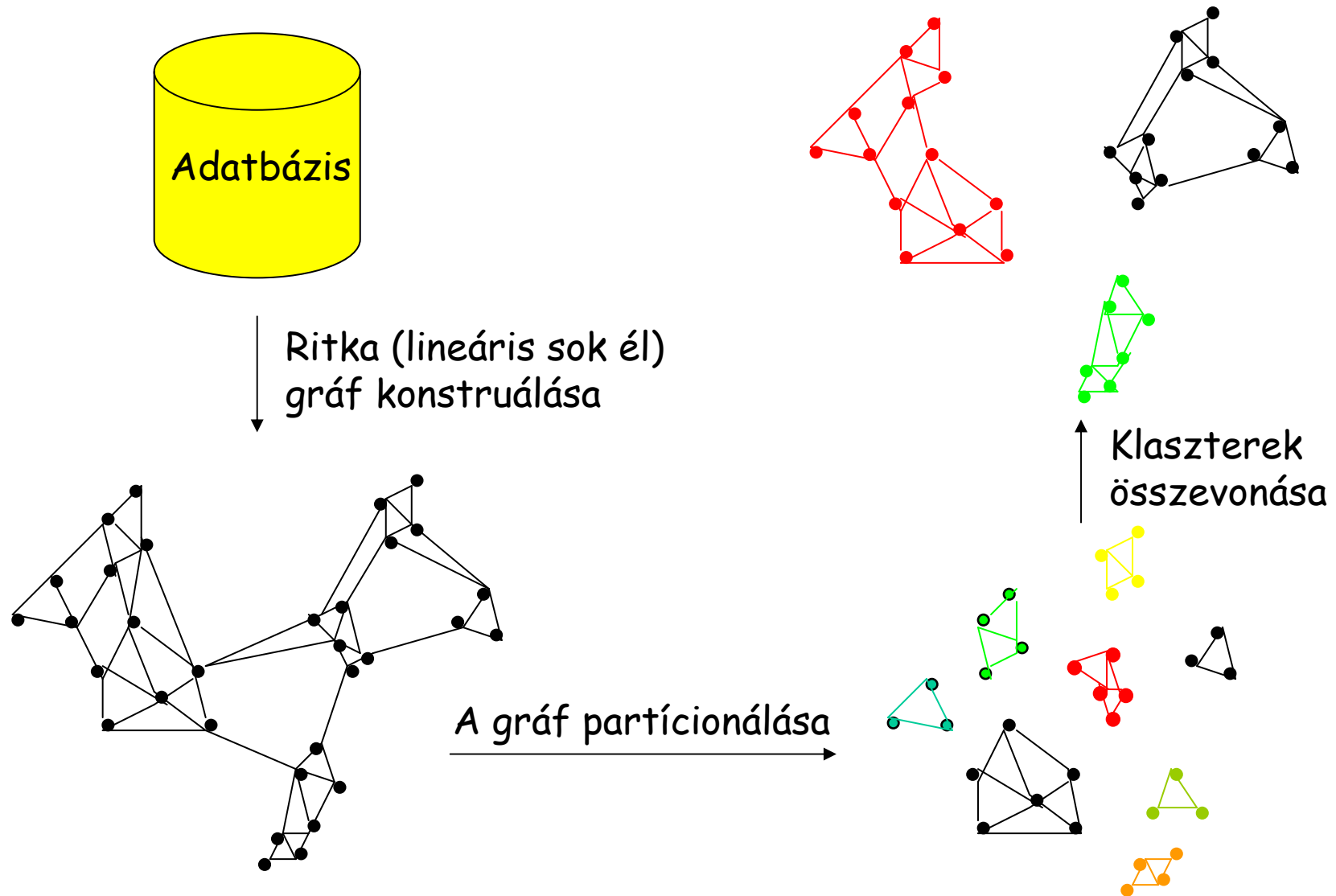
- az algoritmus a linkek számosságán alapulva méri a klaszterek hasonlóságát.
- hierarchikus, alulról felfele építkező klaszterezés.
- futási idő: $O(n^2 \log n)$

CHAMELEON

- **Hasonlósági gráfot** klaszterez (pl. k legközelebbi szomszéd gráfja).
- A hasonlóság mérőszámai adatbázis függőek (dinamikus modell):
 - két klaszter akkor lesz összevonva, ha a két klaszter közötti kapcsolat mérőszámai relatív nagyobbak mint klasztereken belüli kapcsolatokat mutató mérőszámoknál.
- Kétfázisú algoritmus
 - 1) a gráfot elég sok relatív kicsi (homogén) al-klaszterre bontjuk (ez bonyolult és nehéz),
 - 2) az al-klaszter struktúrát alulról fölfelé haladva hierarchikusan klaszterezzünk (a dinamikus modellnek megfelelően).

G. Karypis, E.H. Han and V. Kumar, CHAMELEON:
hierarchical clustering using dynamic modeling, 1999

CHAMELEON vázlat



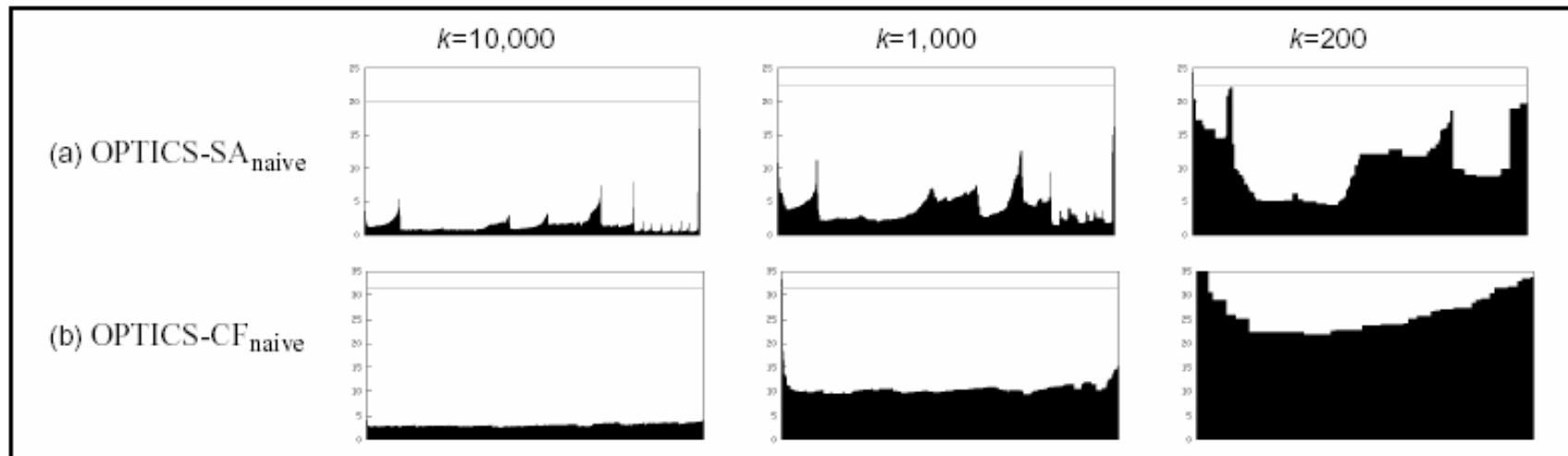
Adattömörítés

Probléma: $O(n \log n)$ is lassú nagyon sok adatnál

Adattömörítés kell !

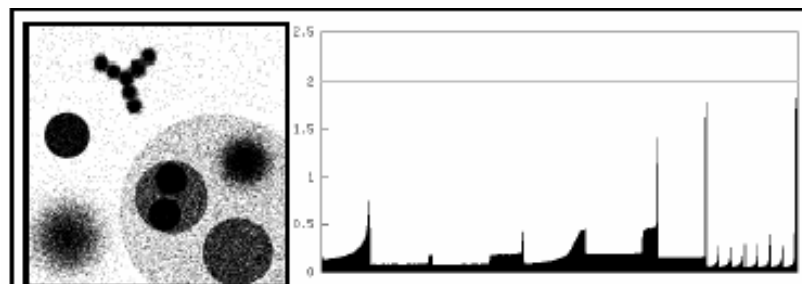
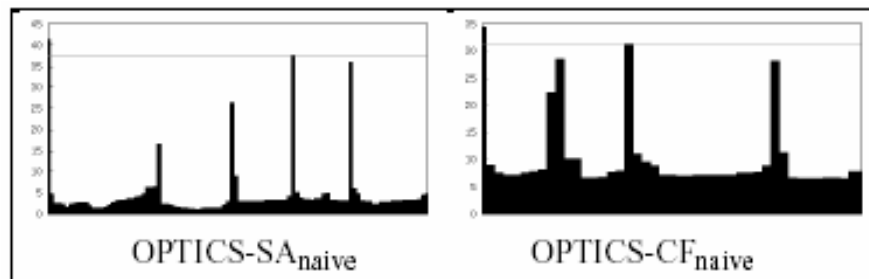
- Vektortérben elégséges statisztika
 - $(n, LS, ss) = \text{Clustering Feature}$
 - n = tömörítésben az adatpontok száma
 - LS = lineáris összeg
 - ss = négyzetes összeg
- Adatok csoportosítása
 - BIRCH algoritmus: CF-fa
 - véletlen minta: legközelebbi mintához tartozik a többi

Problémák a naive tömörítésekkel

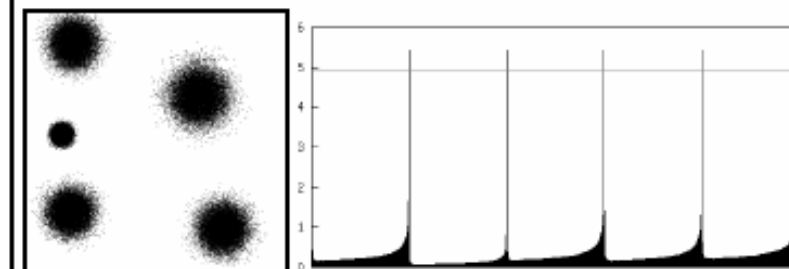


- strukturális torzítás
- méret torzítás
- elveszett objektumok

(súlyozás sem segít)



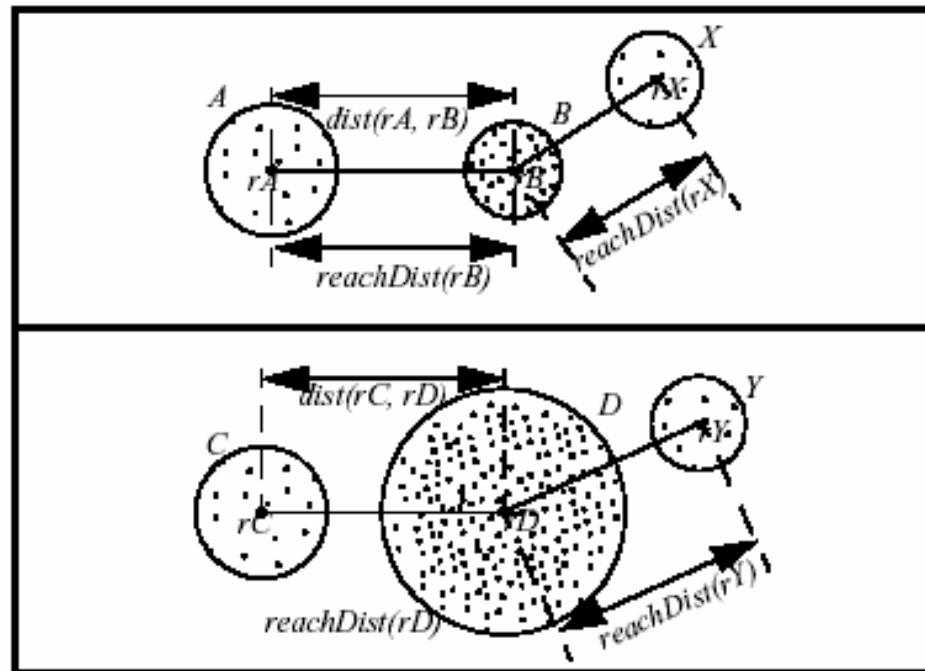
(a) data set DS1 and its reachability-plot



(b) data set DS2 and its reachability-plot

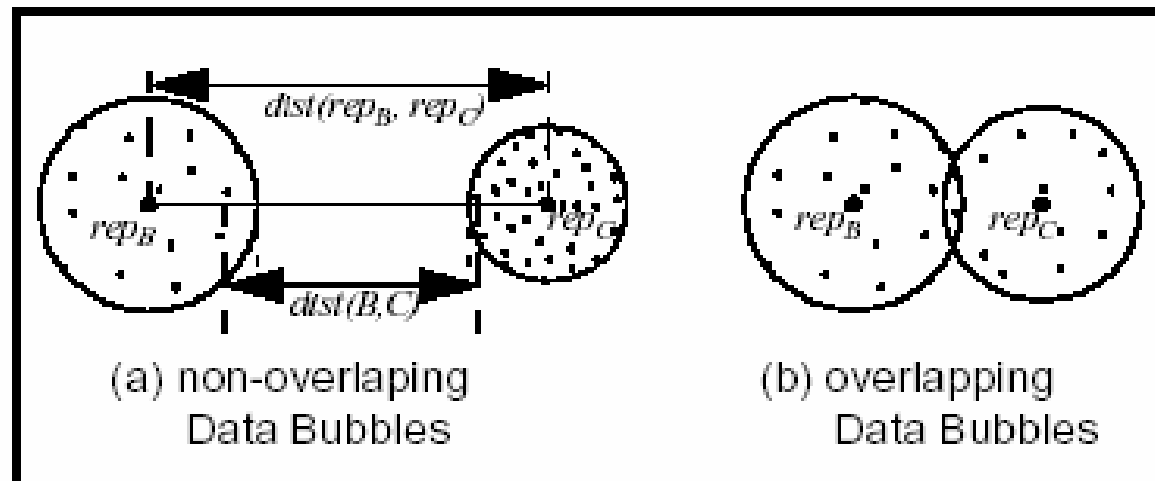
A probléma oka

- A minták közti távolság nem reprezentálja jól az eredeti pontok közti távolságokat.
- Klaszteren belül elérhető-távolságok jóval kisebbek, mint a minták közt vett elérhető-távolságok.



Adatbuborék fogalma

- Adatbuborék B_X , $X=\{X_i\}$ $1 \leq i \leq n$
 - rep_X : B_X reprezentánsa (nem kell, hogy eleme legyen X -nek)
 - e_X : extent; a „legtöbb” elem rep_X körül ekkora sugarú körben van
 - $nnDist(k, B_X)$: becsült átlagos k -távolság X -beli pontokra, $k = 1 \dots MinPts$
- Távolság B és C adatbuborék között: $dist(B,C)$
 - 0, ha $B=C$
 - $\max(nnDist(1,B), nnDist(1,C))$, ha $dist(rep_B, rep_C) < e_B + e_C$
 - $dist(rep_B, rep_C) - (e_B + e_C) + nnDist(1,B) + nnDist(1,C)$

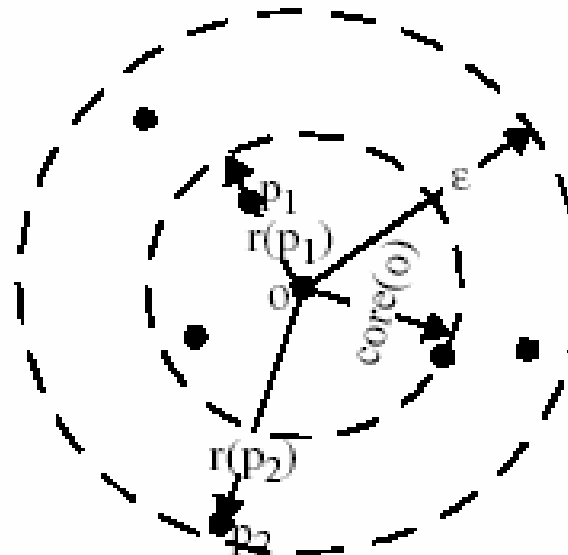


OPTICS távolság értékei

- $N = |\{X | \text{dist}(B, X)\} \leq \varepsilon$
- $\text{core-dist}_{\varepsilon, \text{MinPts}}(B) =$
 - ∞ , ha N -ben MinPts -nél kevesebb pont van
 - $\text{dist}(B, C) + \text{nnDist}(k, C)$, ahol
 - $C = \text{argmax}(\text{dist}(B, C))$: N -beli C -nél B -hez közelebbi buborékokban MinPts -nél kevesebb pont van)
 - $k =$ amennyivel kevesebb pont van
 - B -ben általában több pont van, mint MinPts , ekkor $\text{nnDist}(\text{MinPts}, B)$ lesz az érték

OPTICS távolság értékei

- $\text{reach-dist}_{\epsilon, \text{MinPts}}(B, C) = \max(\text{core-dist}_{\epsilon, \text{MinPts}}(C), \text{dist}(C, B))$
- $\text{virtual-reach-dist}(B)$
 - $\text{nnDist}(\text{MinPts}, B)$ ha $n_B \geq \text{MinPts}$
 - $\text{core-dist}(B)$, egyébként



Példa: d-dimenziós euklideszi vektortér esete

- $B_X(\text{rep}, n, \text{extent}, \text{nnDist}(k, B))$
 - $LS = \text{sum}(X_i)$, $ss = \text{sum}(X_i^2)$
 - $\text{rep} = LS/n$

$$\text{extent} = \sqrt{\frac{\sum_i \sum_j \|X_i - X_j\|^2}{n(n-1)}} = \sqrt{\frac{2n \cdot ss - 2LS^2}{n(n-1)}}$$

$$\text{nnDist}(k) = \left(\frac{k}{n}\right)^{1/d} \cdot \text{extent}$$

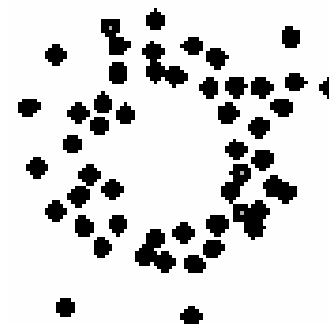
- $V(r) = f(d) r^d$: n pontú, r sugarú gömb térfogata
- $k V(r)/n = f(d) \text{nnDist}(k)^d$: k pontú, $\text{nnDist}(k)$ sugarú gömb

Adatbuborékok nem vektortérben

- Nincs elegendő statisztika:
 - rep nem határozható meg átlagként
 - buborékok közti távolságok nem számolhatók könnyen, pontosan
 - extent nem számolható könnyen, „irány” függő
 - nnDist sem számolható könnyen, pontosan

Buborék reprezentálás

- Csak egy ténylegesen létező pont lehet
- Medoid
 - átlagos távolsága a legkisebb a többi ponthoz
 - $O(n^2)$ számolni
 - nagy tömörítés \rightarrow sok adaton kell
- Lehetőségek
 - kezdeti véletlenül választott minta a rep
 - mintavételezni az adatbuborékot
 - jelölteket fenntartani
 - az első ötlet a leggyorsabb, a többi nem javít sokat ahhoz képest, amennyivel többet kell számolni
- Tehát a véletlenül választott minta lesz a rep
- Adatokat a legközelebbi mintához rendeljük hozzá

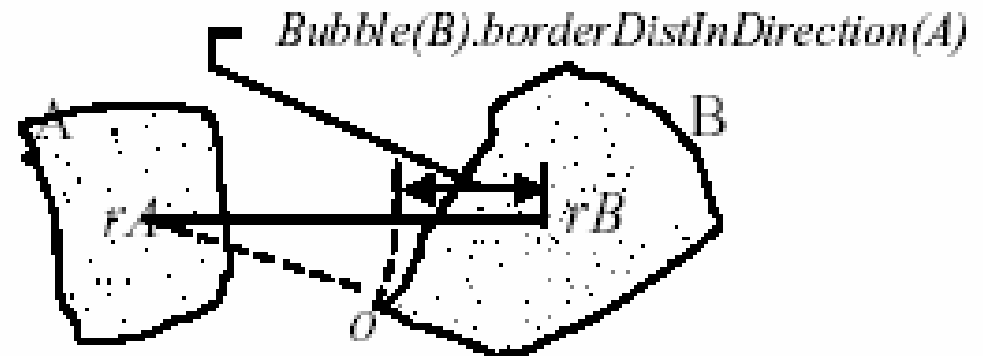
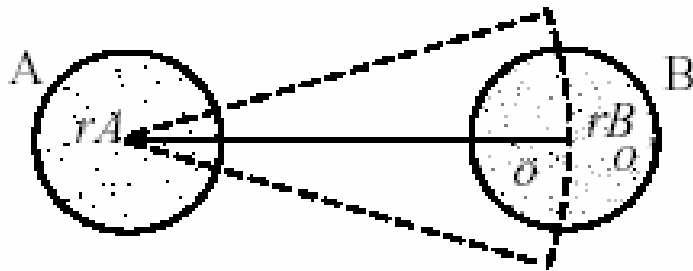


nnDist, core-dist, virtual reach-dist

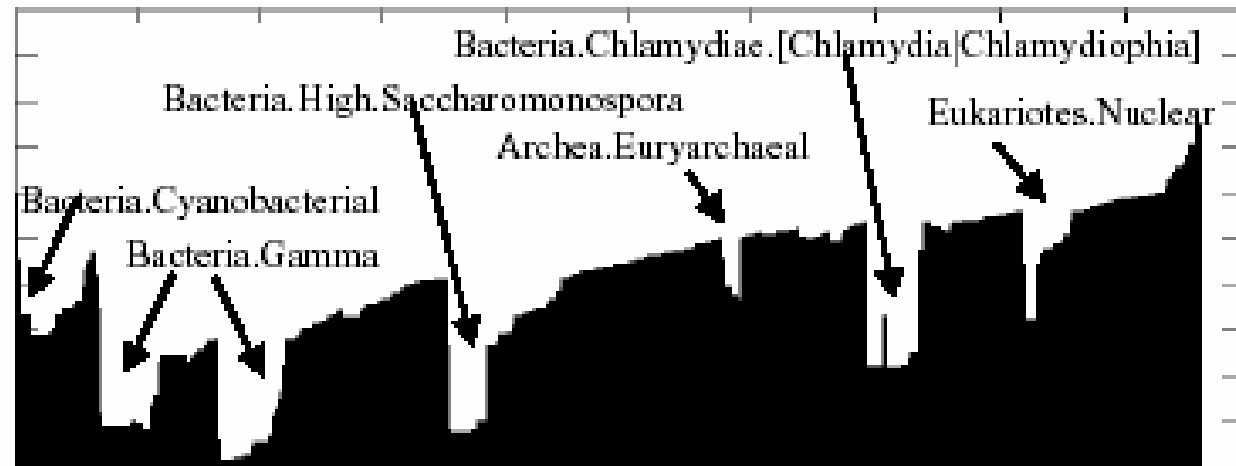
- $\text{nnDist}(\text{MinPts}, B) = \text{core-dist}(B) = \text{virtual reach-dist}(B)$
 - egyezik a vektorteres esettel, mert nagy a tömörítés, így sok pont van egy buborékban
- $\text{nnDist}(\text{MinPts}, B)$
 - rep-hez MinPts db legközelebbi pont közül a legtávolabbi
 - pontosabb számolás nem javít annyit, mint amennyivel többet kell számolni
 - MinPts növelésével egyre pontosabb becslést kapunk

Buborékok közti távolságok

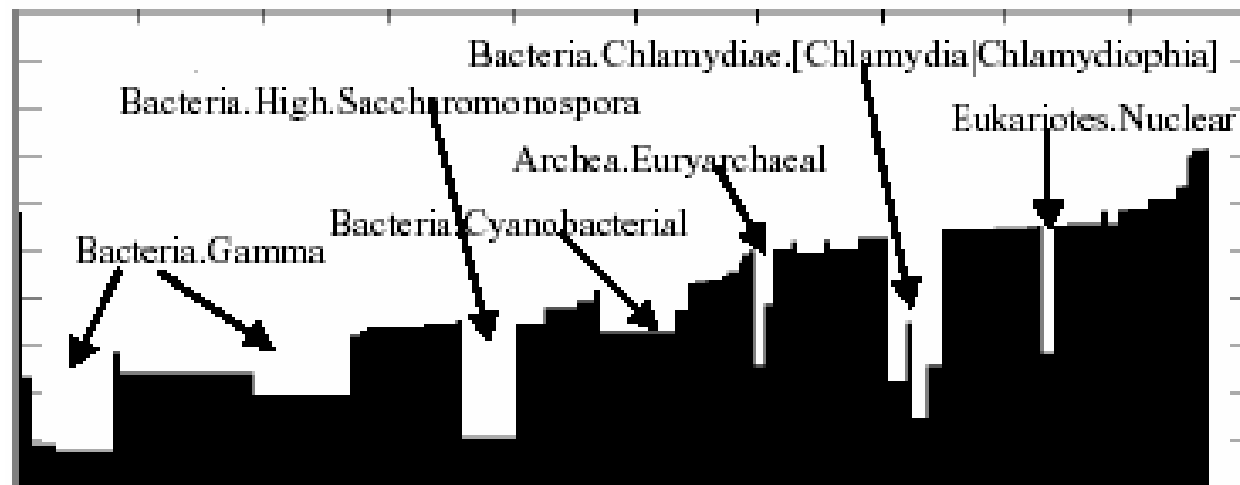
- A, B két buborék: r_A, r_B a reprezentánsok
 - $B.\text{InDirection}(A) = B_A = \{o \in B \mid \text{dist}(o, r_A) \leq \text{dist}(r_A, r_B)\}$
 - $B.\text{InRevDirection}(A) = B_{\text{rev}A} =$ többi pont B -ben
 - $B.\text{borderDistInDir}(A) = \text{dist}(r_A, r_B) - \min_{o \in B} \text{dist}(o, r_A)$



Valódi adat: RNS szekvenciák



(a) Result of OPTICS DS-Real, runtime = 6578 sec



(b) Result of OPTICS-Bubbles, runtime = 638 sec