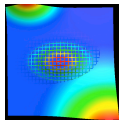
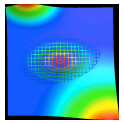


Nagy adathalmazok labor

2018-2019 őszi félév

2018.21-22

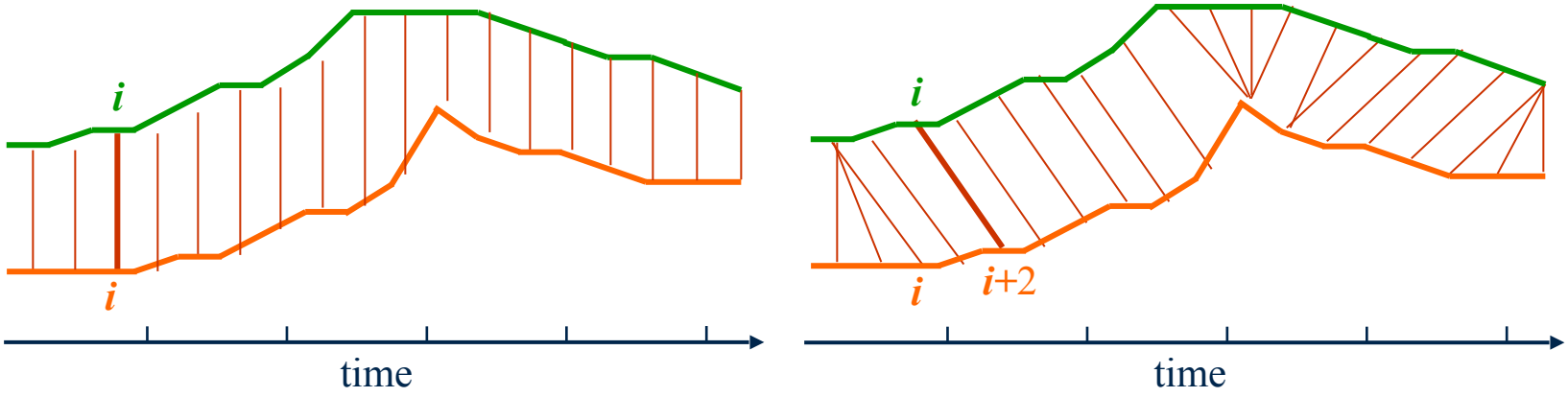
1. Time-series
2. Link analízis: HITS, PR
3. Hálózatok: ER, BA, WS, Broder
4. RF, AdaBoost,GBT

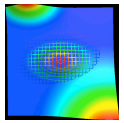


Dynamic Time-Warping

Dynamic Time-Warping [Berndt and Clifford, 1994, Ding et al., 2008]

Euklideszi távolsággal összehasonlítva





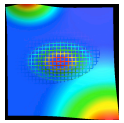
Dynamic Time-Warping

Dinamikus programozás:

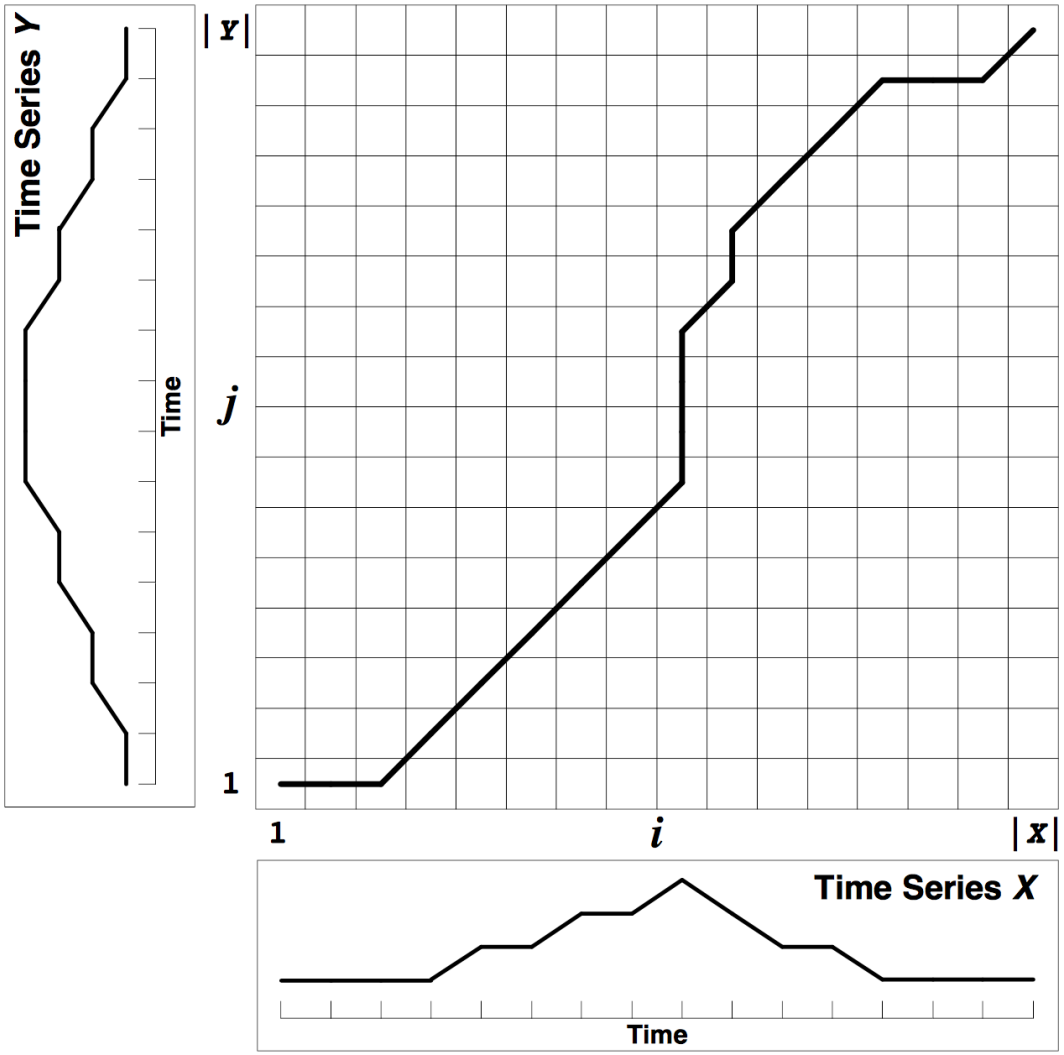
Legyen DTW((x1), (y1)) = |x1 –y1|

$$\begin{aligned} &\mathrm{DTW}^2((x_1,\ldots,x_n),(y_1,\ldots,y_m)) = \\ \min\ &(\mathrm{DTW}^2((x_1,\ldots,x_{n-1}),(y_1,\ldots,y_{m-1})) + (x_n - y_m)^2, \\ &\mathrm{DTW}^2((x_1,\ldots,x_{n-1}),(y_1,\ldots,y_m)), \\ &\mathrm{DTW}^2((x_1,\ldots,x_n),(y_1,\ldots,y_{m-1}))) \end{aligned}$$

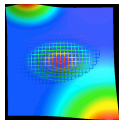
Amennyiben távolság alapú a modellunk a DTW használható [Ding et al., 2008].



Dynamic Time-Warping



src.: Salvador&CHan

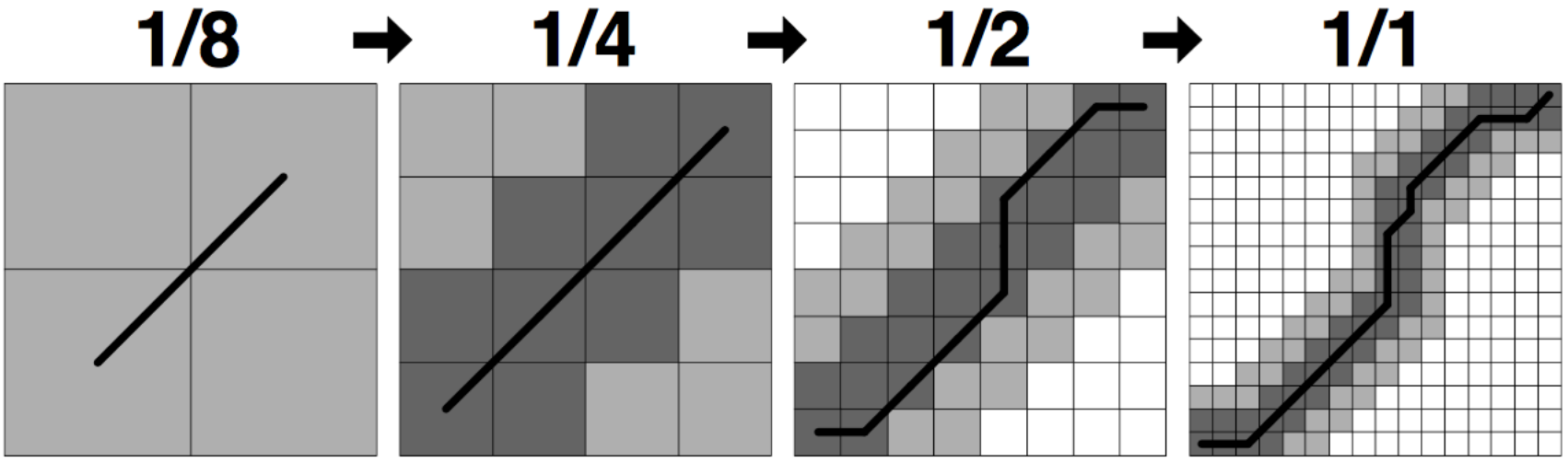


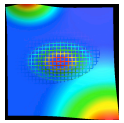
Dynamic Time-Warping

FastDTW [Salvador & Chan, 2007]

Csak hosszú sorozatokra

Python ☺️

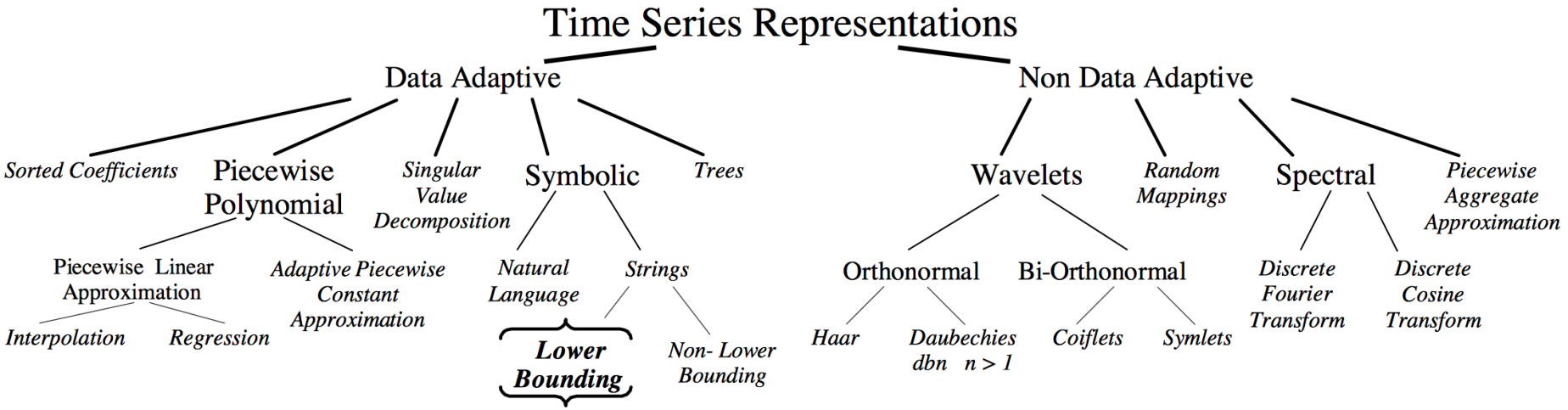


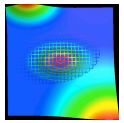


SAX

Lin et al. 2003:
A Symbolic Representation of Time Series, with Implications for Streaming Algorithms

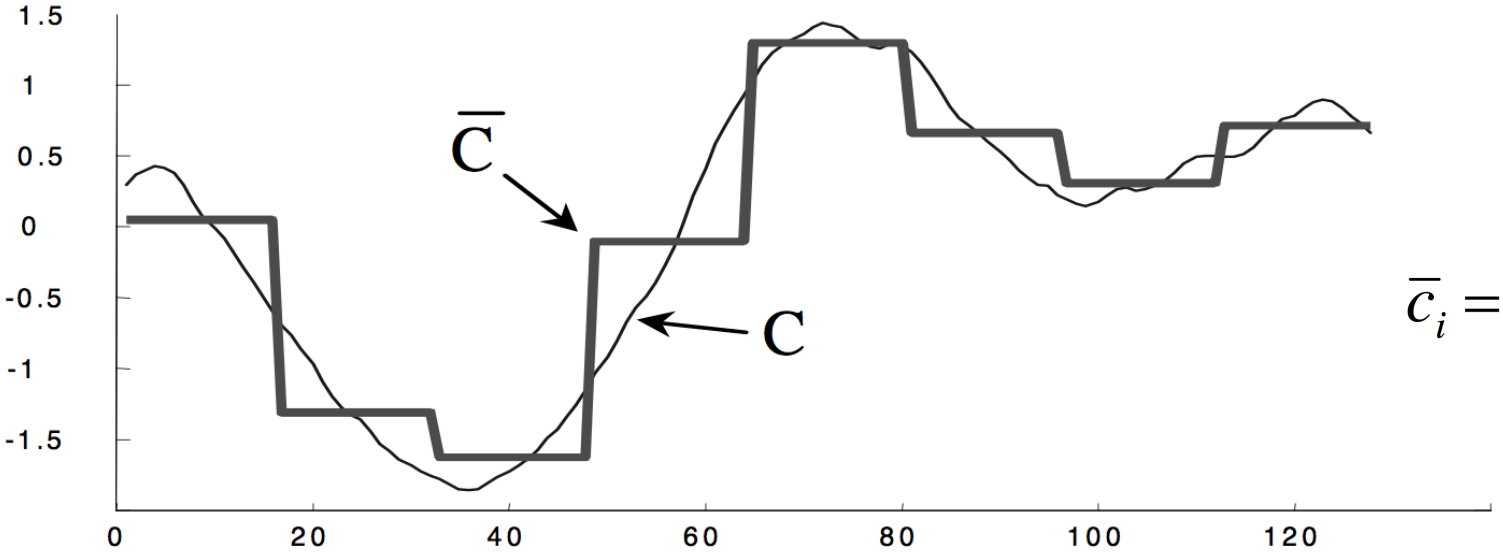
Ötlet: alsó közelítése a távolságoknak



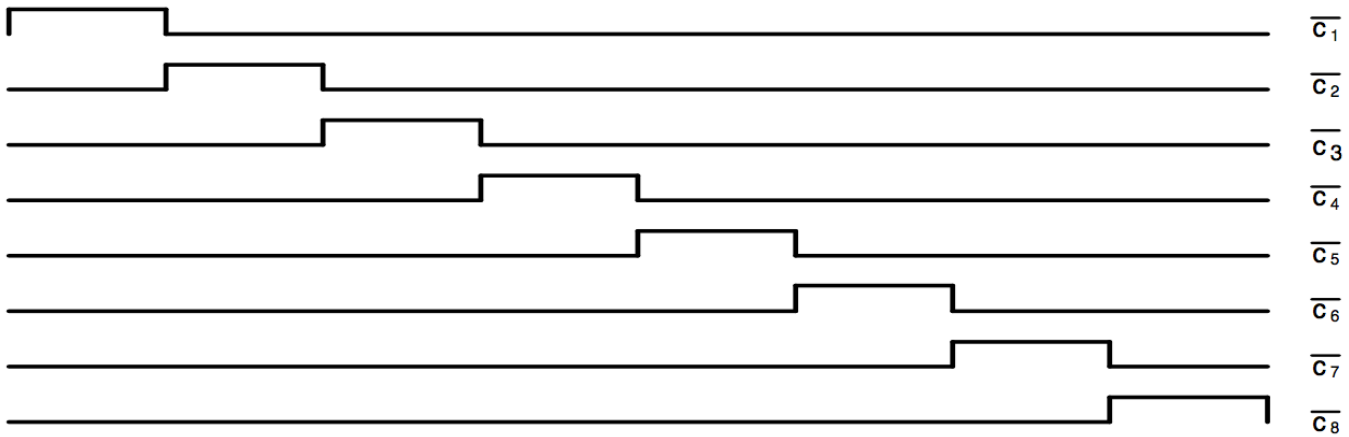


SAX

Piecewise Aggregate Approximation (PAA)

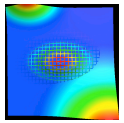


$$\overline{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j$$



De: folytonos

Lin et al. 2003:

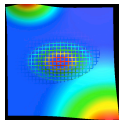


SAX

Diszkretizáció: normalizált TS igazából normál eloszlás -> Határpontok N(0,1)

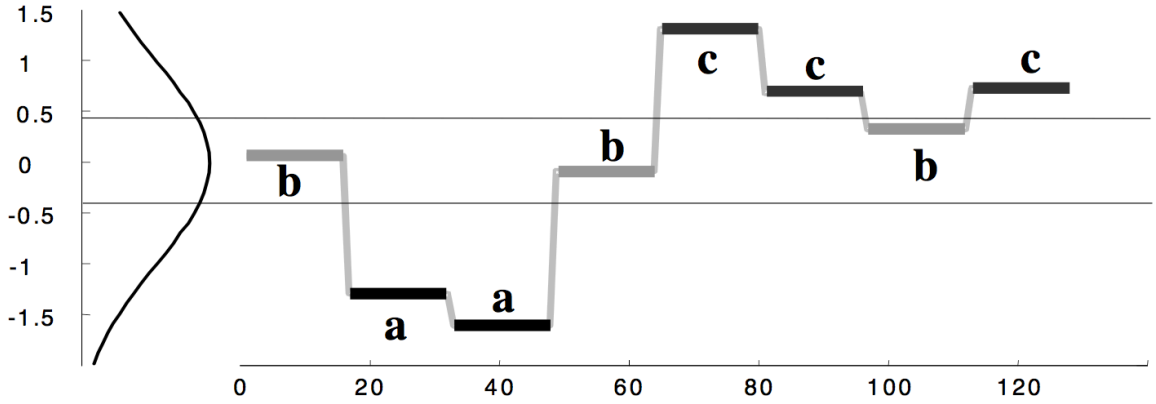
β_i \ a	3	4	5	6	7	8	9	10
β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
β_3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
β_4			0.84	0.43	0.18	0	-0.14	-0.25
β_5				0.97	0.57	0.32	0.14	0
β_6					1.07	0.67	0.43	0.25
β_7						1.15	0.76	0.52
β_8							1.22	0.84
β_9								1.28

Lin et al. 2003:



SAX

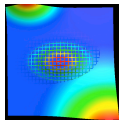
Folytonos TS -> “words”



-> L2/DTW alsó közelítés ismert

De továbbra sem hatékony

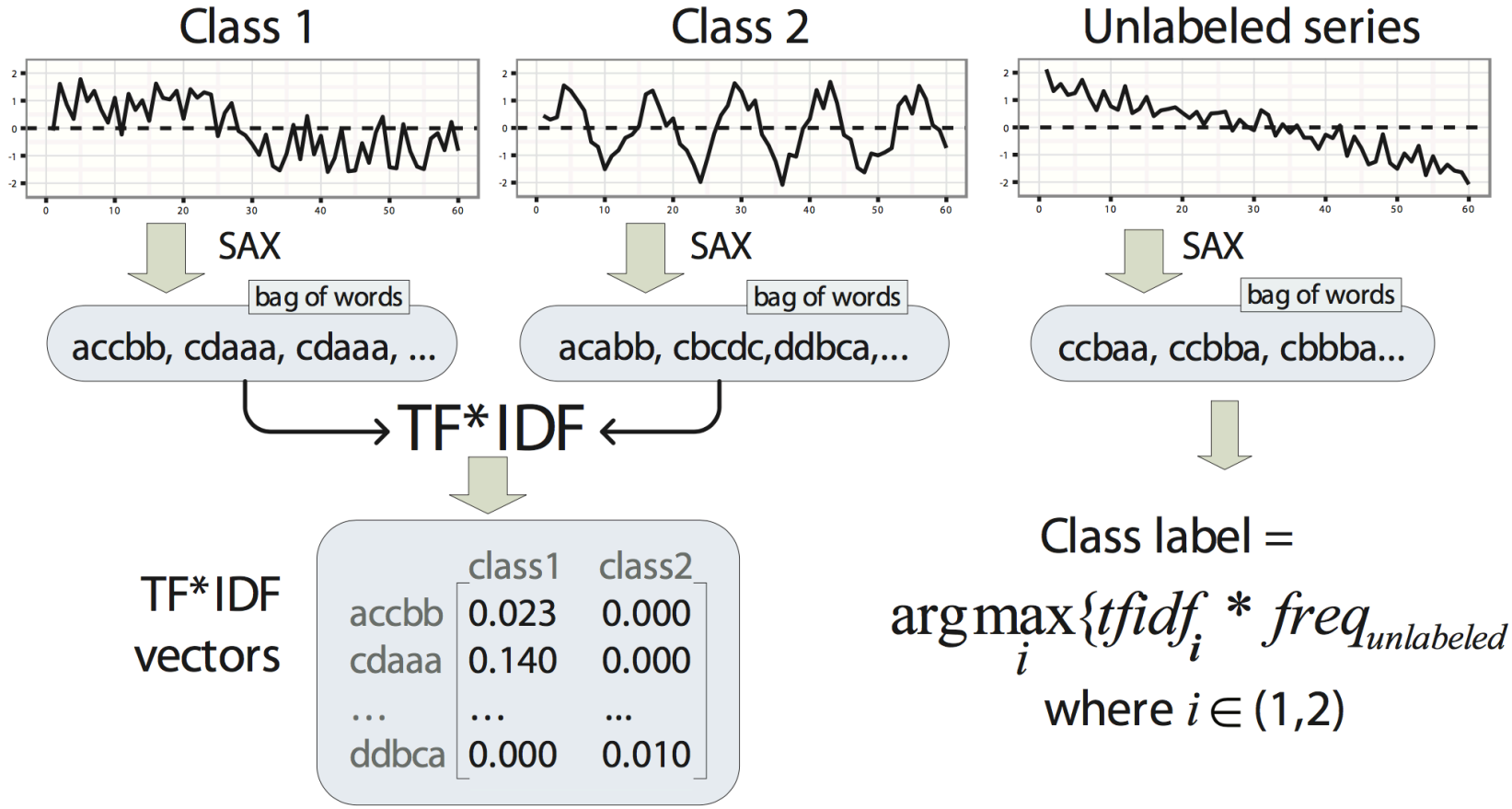
Lin et al. 2003:



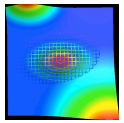
SAX-VSM

Senin et al. 2013:

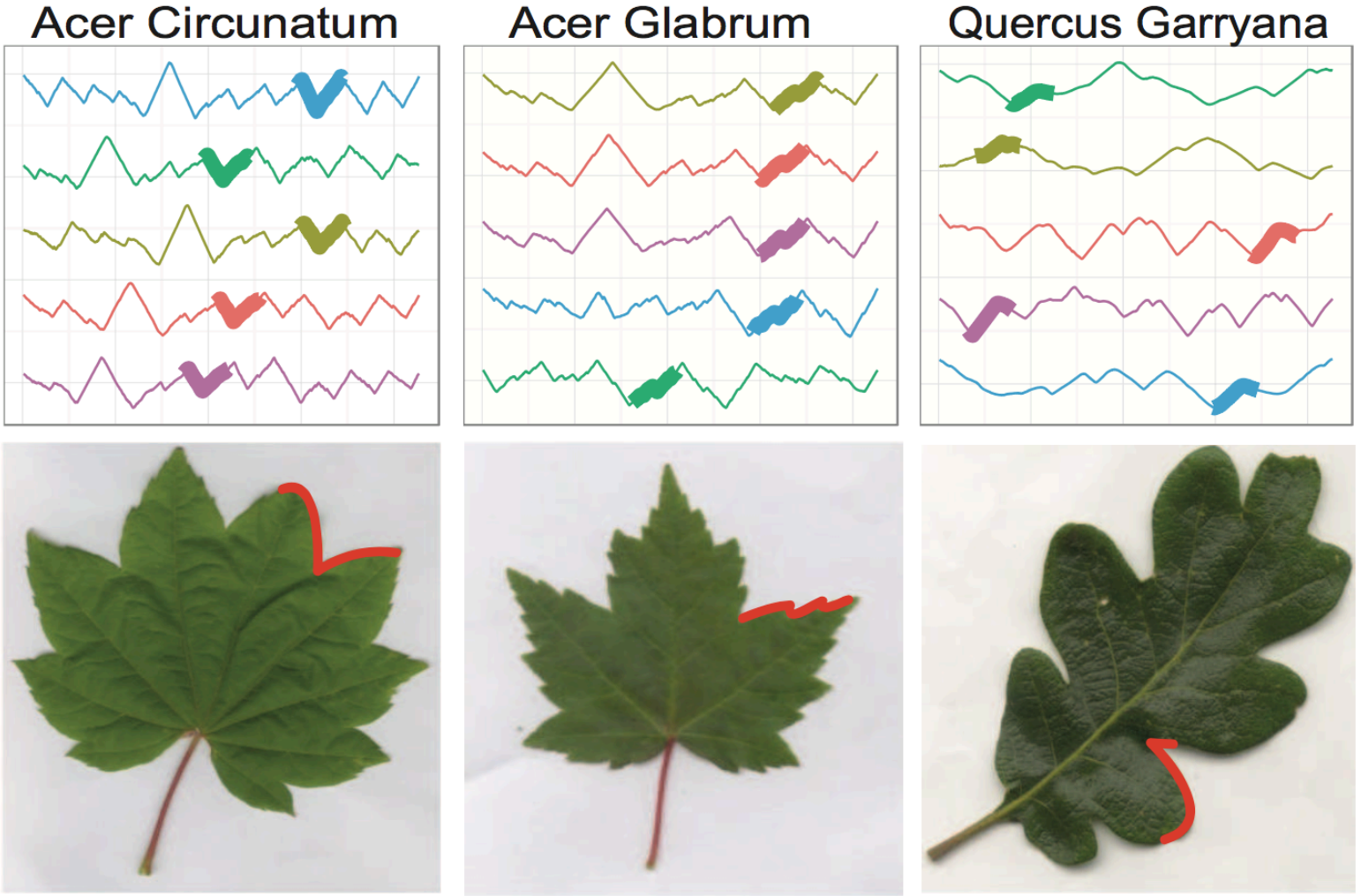
SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model



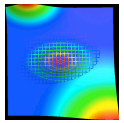
Senin et al. 2013:



SAX-VSM

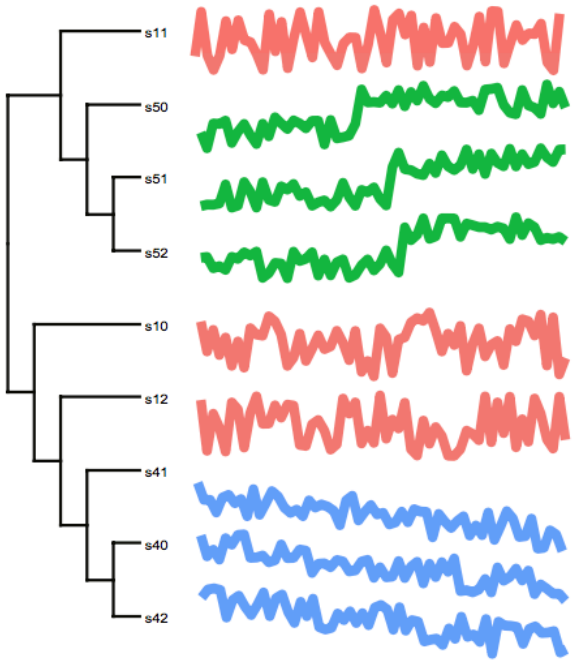


Senin et al. 2013:

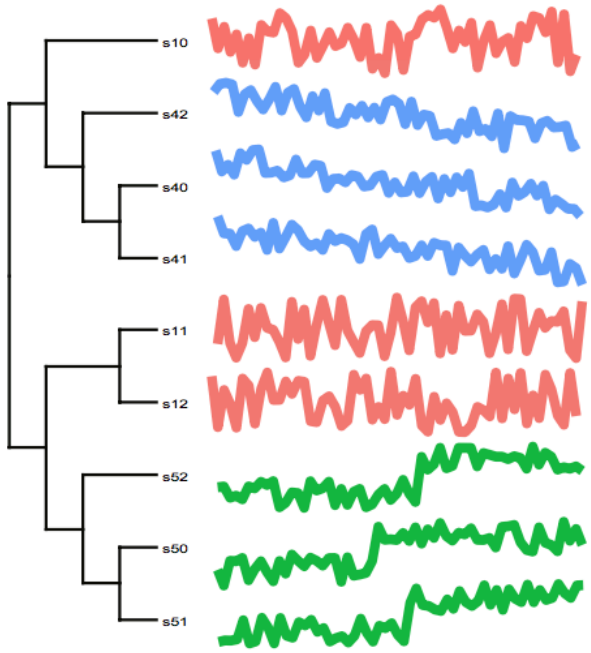


SAX-VSM

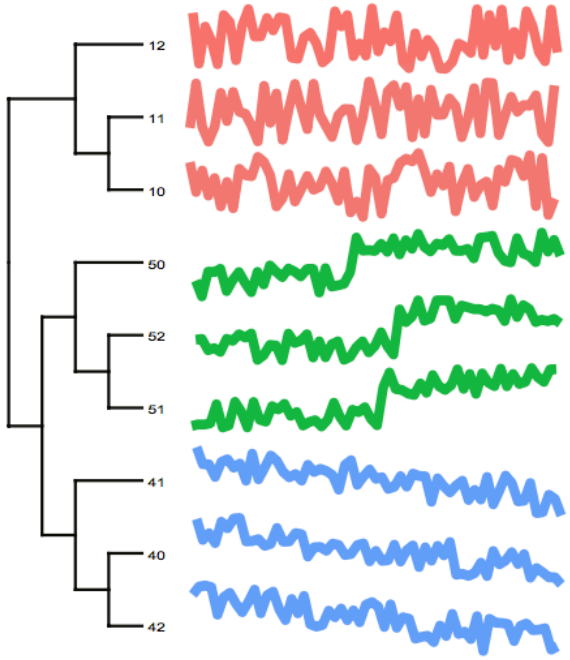
Euclidean



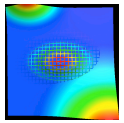
DTW



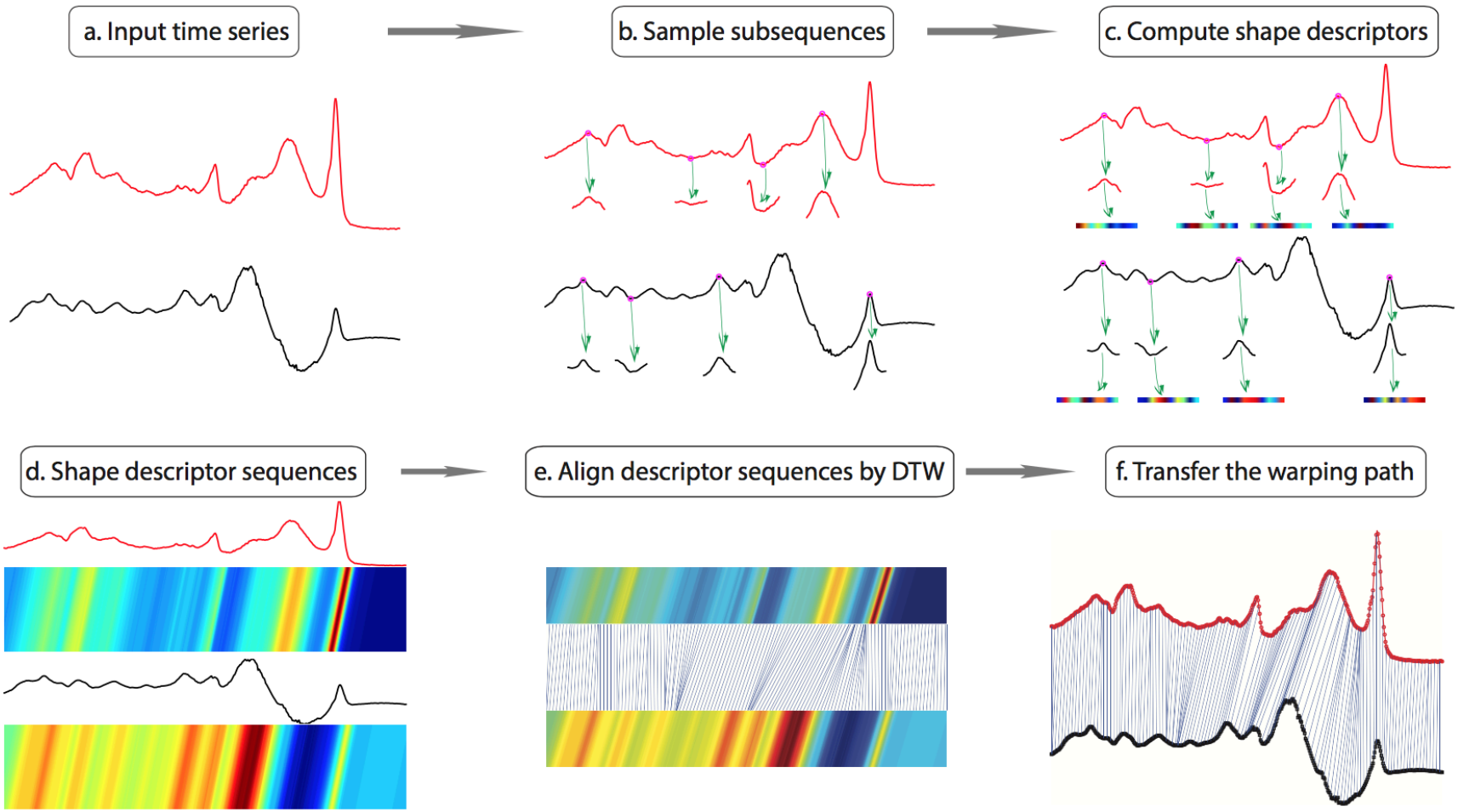
SAX-VSM



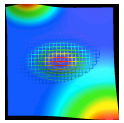
Senin et al. 2013:



Zhao 2016: shapeDTW: shape Dynamic Time Warping



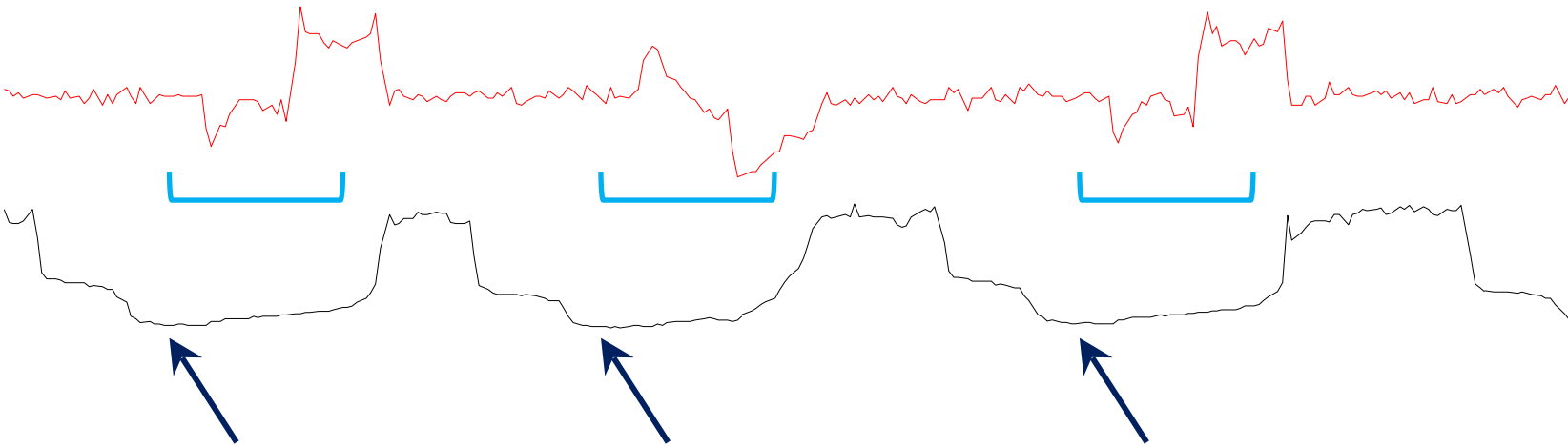
https://github.com/jiapingz/shapeDTW



Matrix Profile [Yeh et al. 2016, Zhu et al. 2017, Linardi et al. 2018 ...]

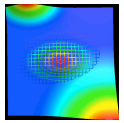
time series, T

matrix profile, P



Lokális minimumok -> hipotézis ezek a "motif"

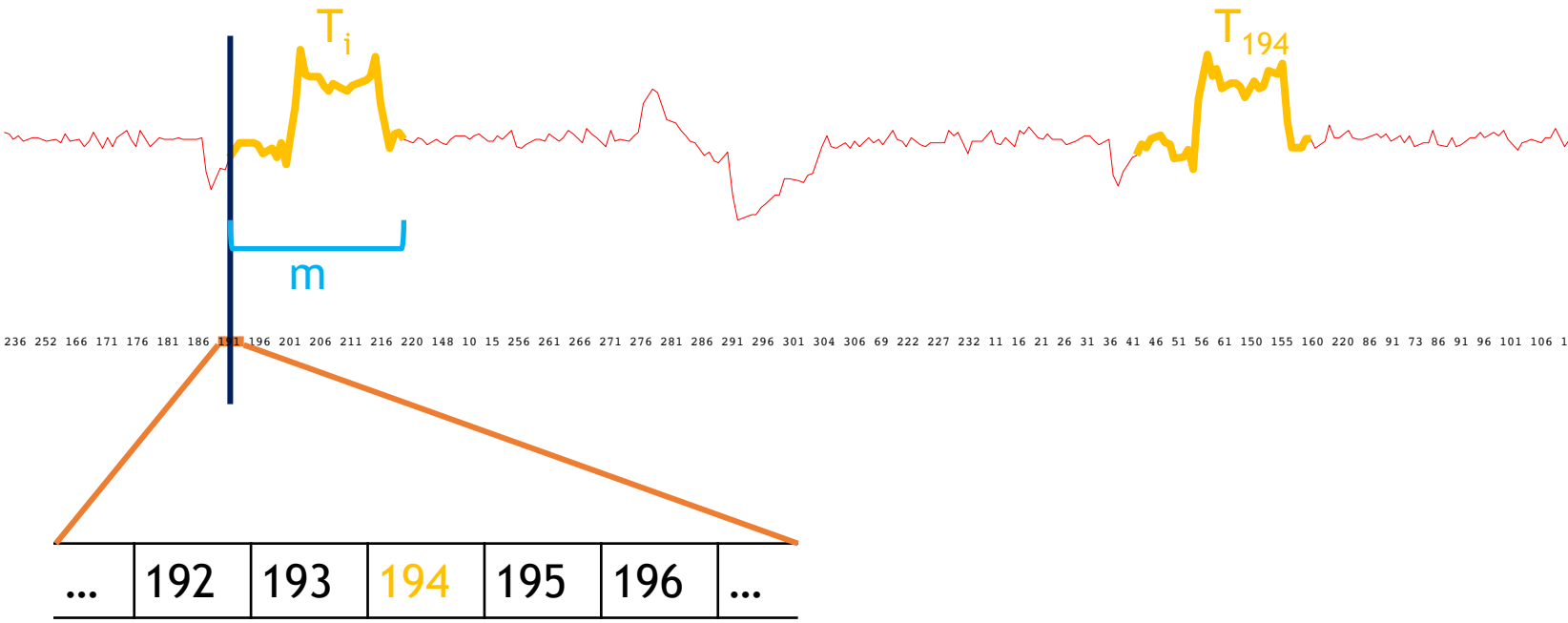
http://www.cs.ucr.edu/%7Eeamonn/MatrixProfile.html



Matrix Profile [Yeh et al. 2016, Zhu et al. 2017, Linardi et al. 2018 ...]

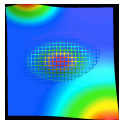
time series, T

matrix profile index, I



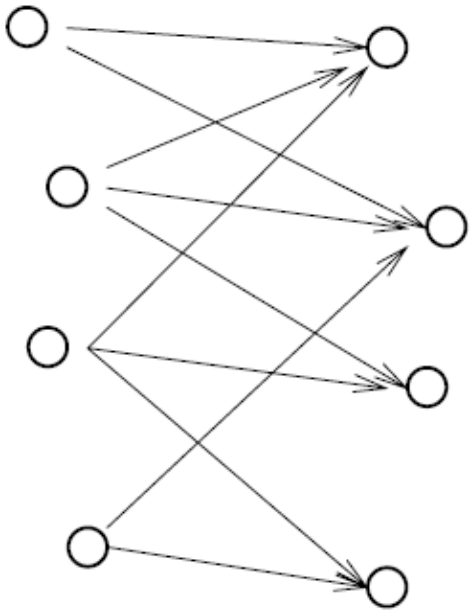
T_i legközelebbi szomszédja T_{194}

http://www.cs.ucr.edu/%7Eeamonn/MatrixProfile.html

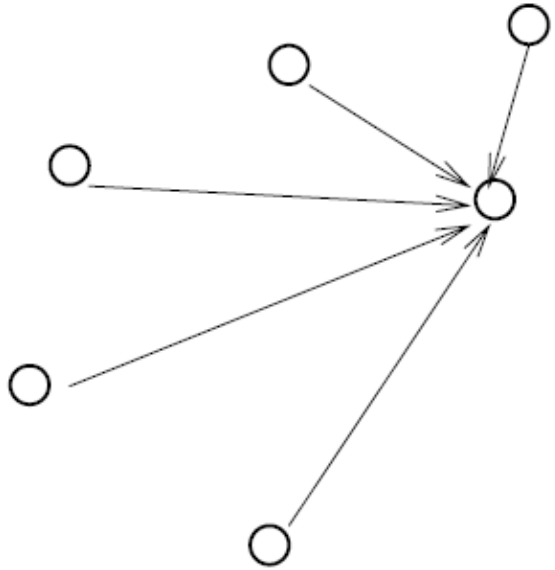


Web gráf: HITS bevezetés

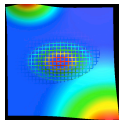
Hyperlink-Induced Topic Search (HITS)



hubs



authorities



Web gráf: HITS

Hyperlink-Induced Topic Search (HITS) Kleinberg ’98

1. Hubs: gyűjtőoldalak, azon oldalak melyek jó authority oldalakra mutatnak
2. Authorities: maguk a releváns oldalak
   Azok az oldalak akikre jó hub-ok mutatnak

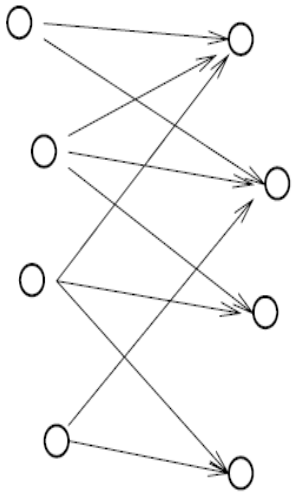
A célunk meghatározni a két csoportot.

Minden oldalhoz hozzárendelünk egy nem negatív authority (x) és hub (y) értéket:

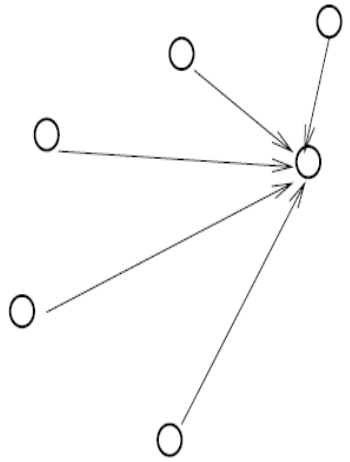
$$I(G, x, y): x^p \leftarrow \sum_{q:(q,p)\in E} y^q$$

$$O(G, x, y): y^p \leftarrow \sum_{q:(p,q)\in E} x^q$$

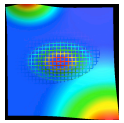
$$\sum_p (x^p)^2 = 1 \qquad \sum_p (y^p)^2 = 1$$



hubs



authorities



Web gráf: HITS

HITS(G,k,q)

G: oldalak és élek halmaza, melyek a q keresés mag vagy bővítési halmazába tartoznak
k: konstans
z legyen egy n dimenziós valós vektor (1; 1; 1; : : :; 1)
Legyen x_0 := z:
Legyen y_0 := z:

for i = 1,2 .. k

O(G,x_{i-1}; y_{i-1}) → megkapjuk az új x' súlyokat

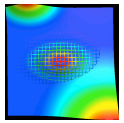
I(G,x_{i-1}; y_{i-1}) → megkapjuk az új y' súlyokat

Normalizáljuk x'-t, megkapjuk az új x-et

Normalizáljuk x'-t, megkapjuk az új x-et

Adjuk vissza (x_k; y_k)-t.

Állítás: (x_1,x_2,x_3,...) és (y_1,y_2,y_3,....) sorozatok konvergálnak.



HITS

Bizonyítás: Legyen A az adjacencia mátrix a kiválasztott részgráfon (“focus graph”, x:auth, y:hub)

$$x^{(k+1)} = y^{(k)} A$$
$$y^{(k+1)} = x^{(k+1)} A^T$$

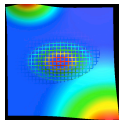
bontsuk ki:

$$x^{(k+1)} = x^{(1)} (A^T A)^k = x^{(1)} U W U^T$$

$$y^{(k+1)} = y^{(1)} (A A^T)^k = y^{(1)} V W V^T$$

ahol W diagonális.

Állítás: a normalizált $x^{(k)}$ és $y^{(k)}$ sorozat konvergál (kezdővektor x=y=(1,1,…,1))



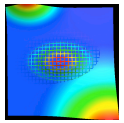
HITS

Állítás átfogalmazva :

$(AA^T)^j y^{(1)}/||(AA^T)^j y^{(1)}||$ sorozat konvergál

Segédtétel: ha egy nxn-es M mátrix pozitív szemidefinit szimmetrikus mátrix, melynek a sajátértékei $\lambda_1>\lambda_2\geq\lambda_3...\geq\lambda_k\geq 0$ ($k<n$), akkor minden d dimenziós valós v vektorra igaz, hogy kifejezhető $v = \Sigma_{i=1..k}\ \alpha_i\omega^{(i)}$ ahol minden i-re $||\omega^{(i)}||=1$ és $\omega^{(i)\ T}\omega^{(j)} = 0$, ha $i\neq j$.

Miután $\omega^{(i)}$ az i-dik sajátvektor: $M\ \omega^{(i)} = \lambda_i\omega^{(i)}$

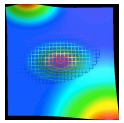


HITS

Miután AA^T = M pozitív szemidefinit szimmetrikus, így:

$$\frac{M^j v}{||M^j v||} = \frac{\sum_{i=1}^{k} \alpha_i M^j w^{(i)}}{||\sum_{i=1}^{k} \alpha_i M^j w^{(i)}||} = \frac{\sum_{i=1}^{k} \alpha_i \lambda_i^j w^{(i)}}{\sqrt{\sum_{i=1}^{k} (\alpha_i \lambda_i^j)^2}}$$

$$\frac{\alpha_1 \lambda_1^j w^{(1)} + \sum_{i=2}^{k} \alpha_i \lambda_i^j w^{(i)}}{\sqrt{(\alpha_1 \lambda_1^j)^2 + \sum_{i=1}^{k} (\alpha_i \lambda_i^j)^2}} \cdot \frac{\frac{1}{\lambda_1^j}}{\frac{1}{\lambda_1^j}} = \frac{\alpha_1 w^{(1)} + \sum_{i=2}^{k} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^j w^{(i)}}{\sqrt{\alpha_1^2 + \sum_{i=1}^{k} (\alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^j)^2}} \rightarrow w^{(1)}$$



Random surfer
“Stochasztikus szörfölő”

Feltételezés: véletlen séta az éleken.

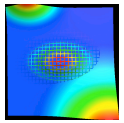
Minden pillanatban uniform módon egy hyperlinken tovább lépünk.

$$Pr(i \mid j) = 1/d(j)$$

Vegyük az adjacencia mátrixát a gráfunknak.

Cseréljük ki az értekeket az átmenet valószínűségekre: M

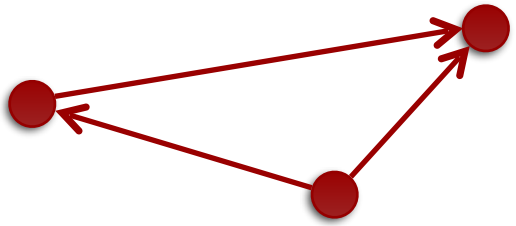
Sorösszeg?



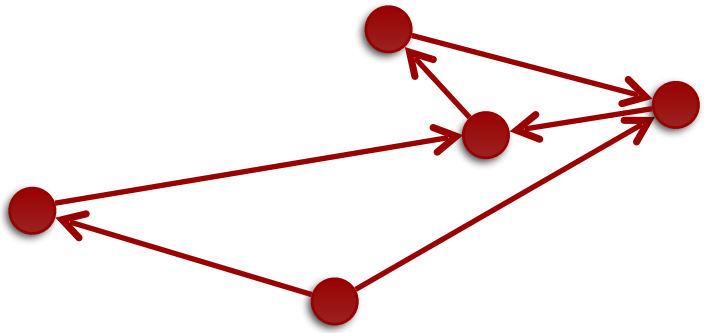
Random surfer
“Stochasztikus szörfölő”

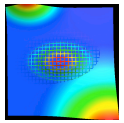
Milyen esetekben nem 1 a sorösszeg?

Zsákutca, forrás:



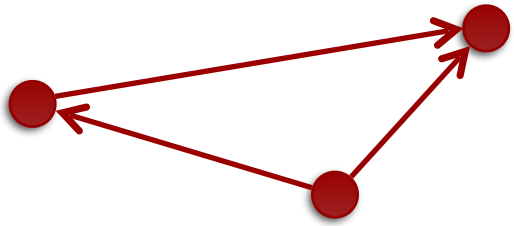
Pókháló:



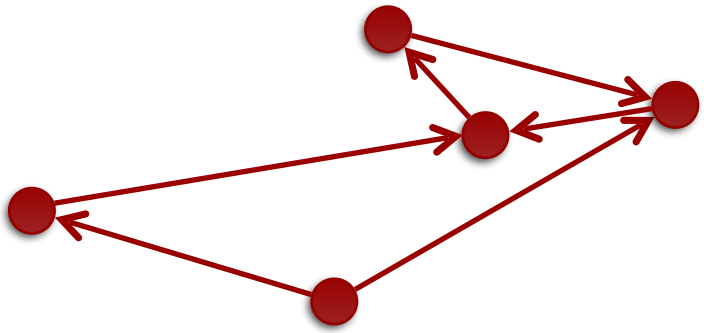


Random surfer
“Stochasztikus szörfölő”

Zsákutca, forrás:



Pókháló:



Ergódikus:

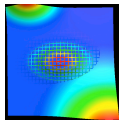
- erősen összefüggő
- aperódikus

Ha egy MC ergódikus (Perron-Frobenius):

Létezik stacionárius eloszlás:

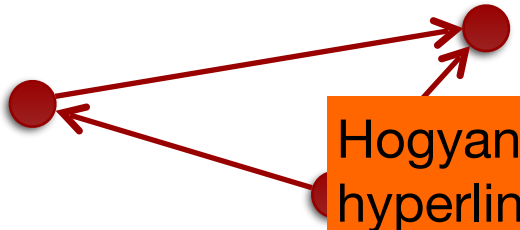
$$\pi^T M = \pi^T$$

Sőt: a legnagyobb sajátérték egyszeres!

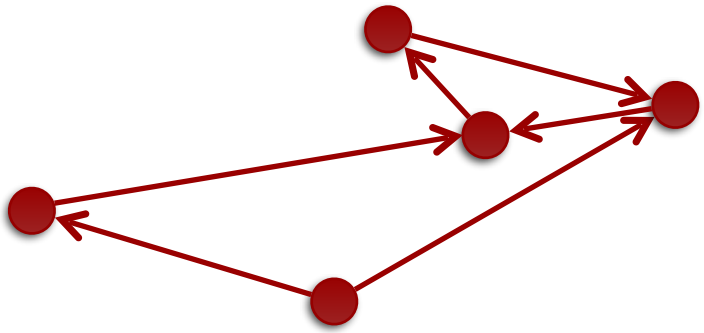


Random surfer
“Stochasztikus szörfölő”

Zsákutca, forrás:



Pókháló:



Ergódikus:

- erősen összefüggő
- aperódikus

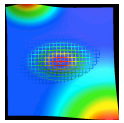
Hogyan tehetjük könnyen ergódikussá a meglévő hyperlink gráfunkat?

Tényleg uniform a teleportáció?

Létezik stacionárius eloszlás:

$$\pi^T M = \pi^T$$

Sőt: a legnagyobb sajátérték egyszeres!



PageRank

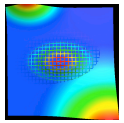
Larry Page , Sergey Brin , Rajeev Motwani és Terry Winograd 1998-as cikke.

Adott egy hyperlinkekkel összekötött dokumentumokból álló hálózat, mint pl. a WWW.

Általános szöveg alapú keresés során a releváns dokumentumok (a keresett szavakat tartalmazó dokumentumok) sorrendjét a legjobb illeszkedés alapján határozzák meg. (amelyik dokumentum leginkább illeszkedik a kérdésre az kerül előre)

Az algoritmus sok hibás vagy igazából nem releváns találatot hátra sorol, felhasználva a hivatkozások hálózatát.

Az elv egyszerű: amelyik oldalra többen és/vagy fontosabbak hivatkoznak, fontosabb mint amire kevesebben és/vagy kevésbe fontosak.

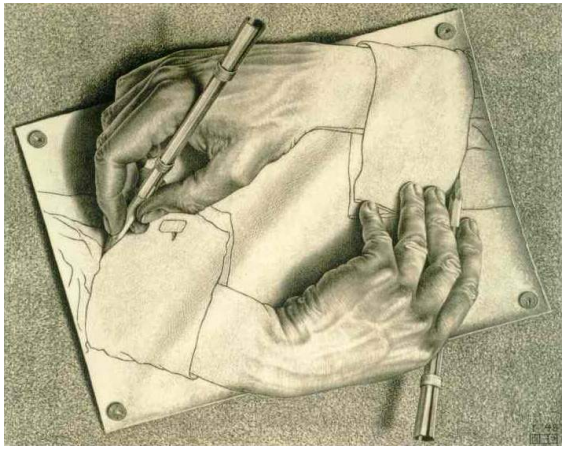


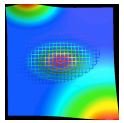
PageRank

Egy adott A dokumentum (oldal) PageRank értéke PR(A):

$$PR(A) = \sum_{B \in I_A} \frac{PR(B)}{L(B)}$$

I jelöli az egy elemre hivatkozó oldalak halmazát, L(B) a B oldal kimeneti linkjeinek száma, PR(B) pedig B PageRank értéke.



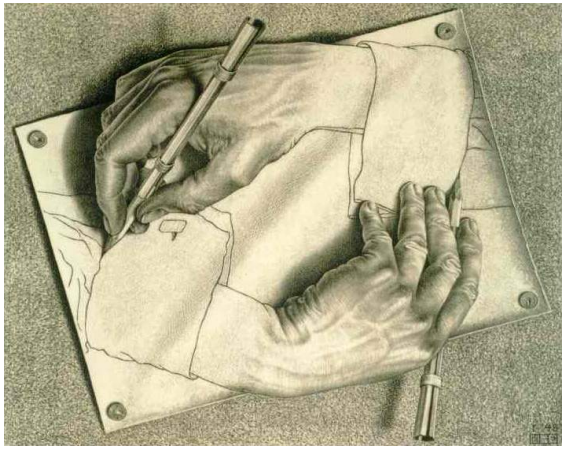


PageRank

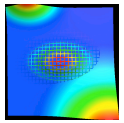
Egy adott A dokumentum (oldal) PageRank értéke PR(A):

$$PR(A) = \sum_{B \in I_A} \frac{PR(B)}{L(B)}$$

I jelöli az egy elemre hivatkozó oldalak halmazát, L(B) a B oldal kimeneti linkjeinek száma, PR(B) pedig B PageRank értéke.



Mi a kapcsolat a sztochasztikus szörfölő modellel?



PageRank

“Random surfer” modell: folyamatosan csökkenő aktivitás → teleportation

$$PR(A) = 1 - d + d \sum_{B \in I_A} \frac{PR(B)}{L(B)}$$

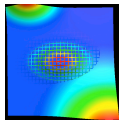
vagy

$$PR(A) = \frac{1-d}{N} + d \sum_{B \in I_A} \frac{PR(B)}{L(B)}$$

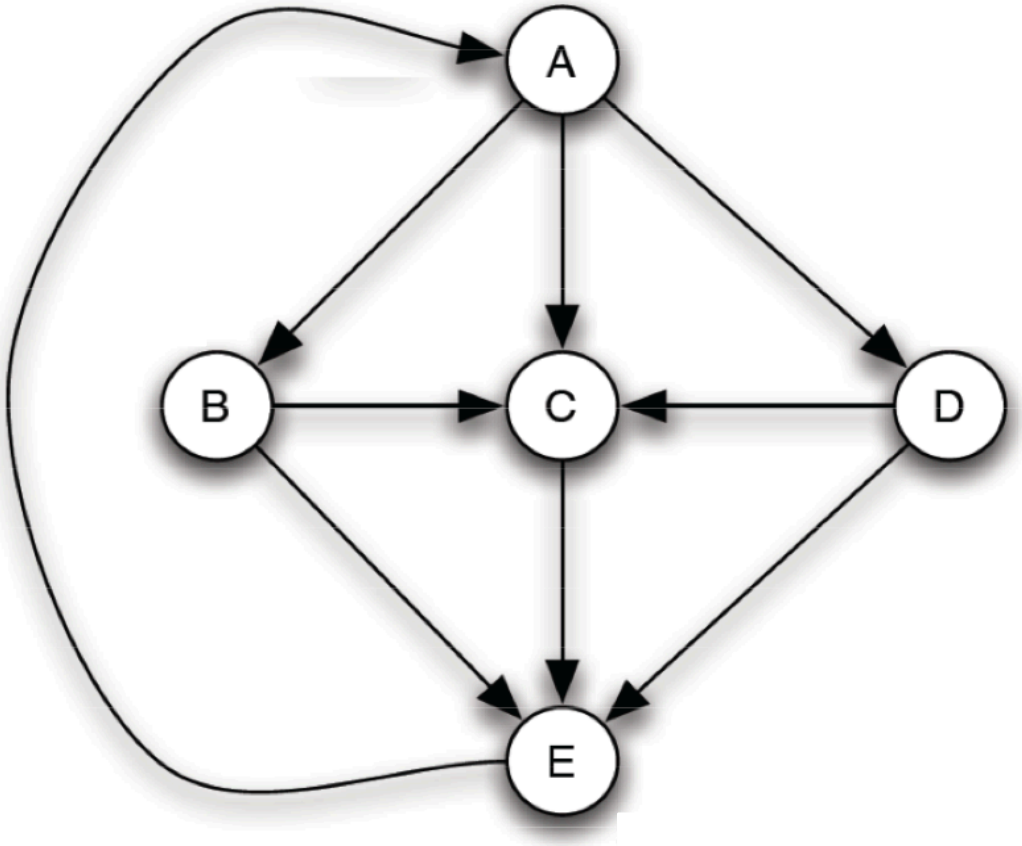
ahol N az összes oldal száma.

Az algoritmus lehetőséget ad linkfarmok vagy mesterséges (fizetett) hivatkozások alapján manipulálásra.

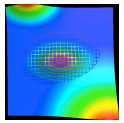
Épp ezért egy jó kereső emellett nem veszi figyelembe a már detektáltan ranking módosító honlapokat illetve linkeket -> web SPAM!



Page Rank



Órai feladat 1:
Mennyi lesz a PR értéke az A,B,C,D,E pontoknak?

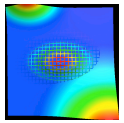


HITS vs. PageRank

	HITS	PageRank
Gráf	Keresésenként más	fix
Mértékek	Hub és auth. értékek	PR értékek

Ezeken kívül fontos különbség, hogy a PageRank a teljes gráf struktúráját próbálja feltérképezni, a HITS eredeti célja egy topik alapján kiválasztott részgráf pontjait rendezi két csoportba.

Mindkét módszer más más esetekben hatékony, de a HITS-et már maga a számításigénye miatt is ritkán használják a gyakorlatban.



Generatív hálózat modellek

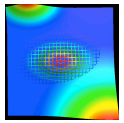
Jelenségek terjedésére nagy hálózatokban általában mint véletlen folyamatokra gondolunk

pl. járványok vagy szociális hálózatok

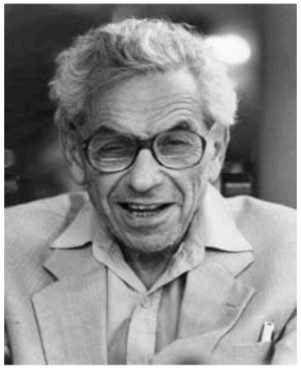
Közös bennük, hogy a hálózat struktúrája komplex

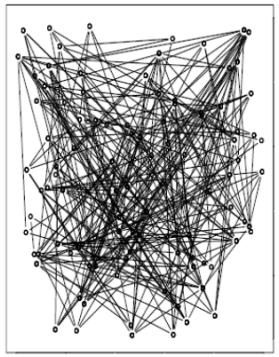
Alapvető cél, hogy generatív modellekkel szimuláljunk valós hálózatokat

Mit lehet tudni ezekről a hálózatokrol? Mit lehet mérni?

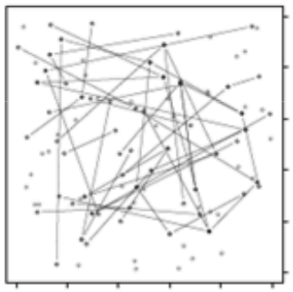


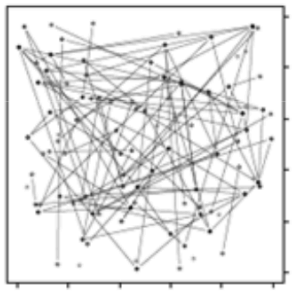
Véletlen gráf avagy Erdős-Rényi
Forrás: Benczúr András

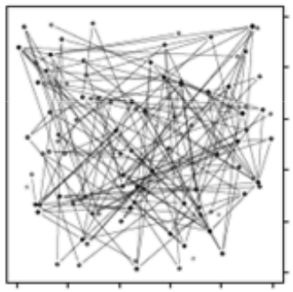


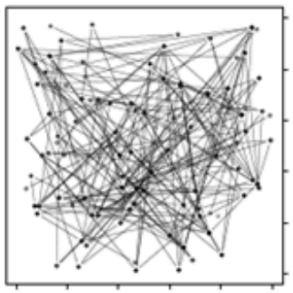


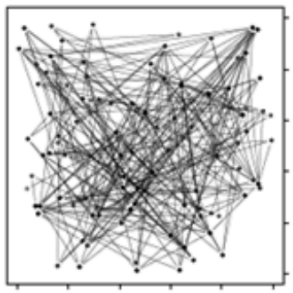
Erdös Rényi Erdös–Rényi

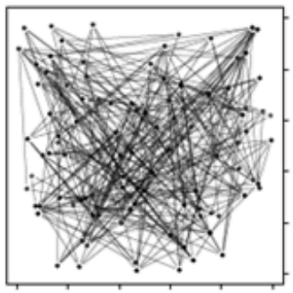


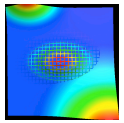












Véletlen gráf avagy Erdős-Rényi

G(n,p): n csúcs mellett p valószínűséggel (függetlenül!) behúzunk egy élt

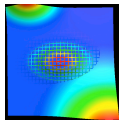
Sokat tudunk róla (ER 1959, Bollobás et al 2002):

Élek várható száma

$$|E| = \binom{n}{2} p = pn(n-1)/2$$

Átlagos fokszám

$$z = \frac{2|E|}{n} = \frac{2\binom{n}{2}p}{n} = (n-1)p$$



Véletlen gráf avagy Erdős-Rényi

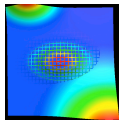
Fokszám eloszás Binomiális:

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

Ami ha n nagy Poisson (z legyen az átlagos fokszám)

$$P(k) = \frac{z^k e^{-z}}{k!}$$

Első kérdés: milyen az ismert hálózatainkban a fokszámeloszlás?



Véletlen gráf avagy Erdős-Rényi

Összefüggőség: ha … , akkor majdnem biztosan

np < 1: legnagyobb összefüggő komponens O(log(n)) nagyságrendű

np = 1: legnagyobb összefüggő komponens O(n^(2/3)) nagyságrendű

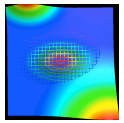
np > 1 konstans: O(n) a legnagyobb összefüggő komponens és a második legnagyobb legfeljebb O(log(n)) nagyságrendű

np < (1-e) log(n): van izolált csúcs

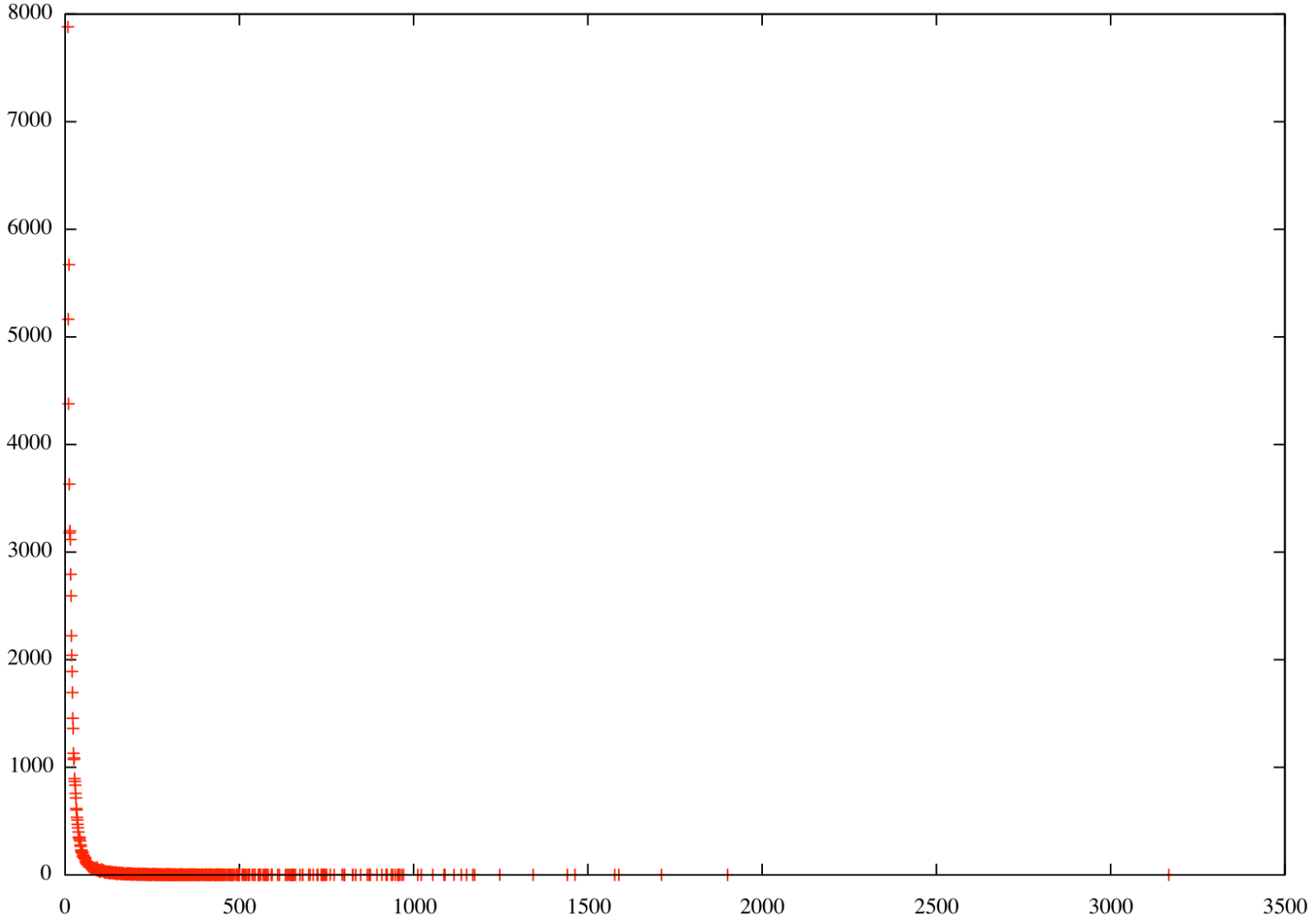
np > (1+e) log(n): összefüggő

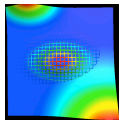
Átmérő:

log(n)/log(p(n-1))

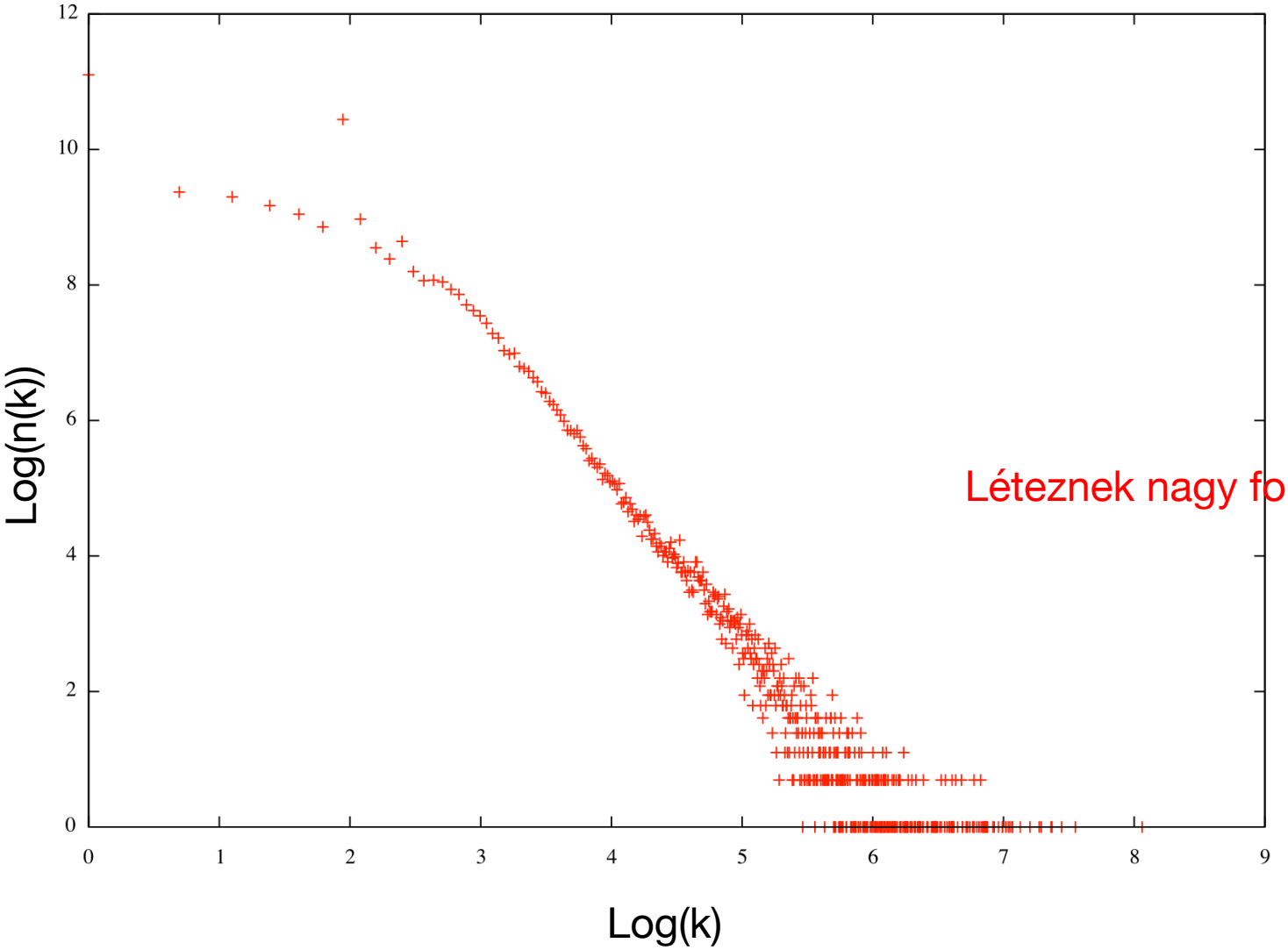


Wiki graph (z=11.1667)

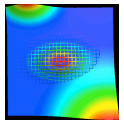




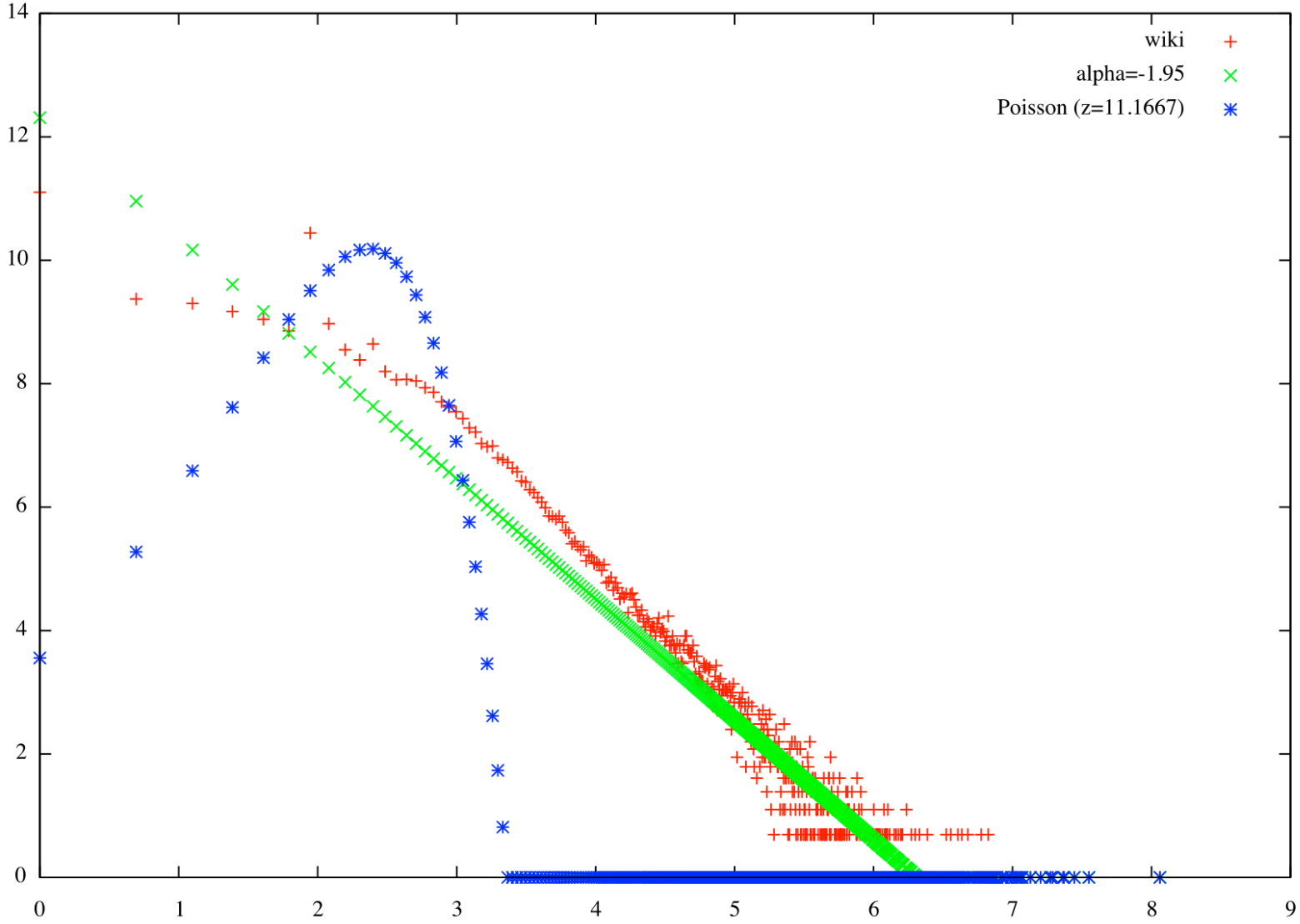
Sokat nem mondott… log-log skála

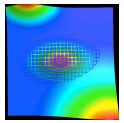


Léteznek nagy fokszámú elemek!
(heavy tail)

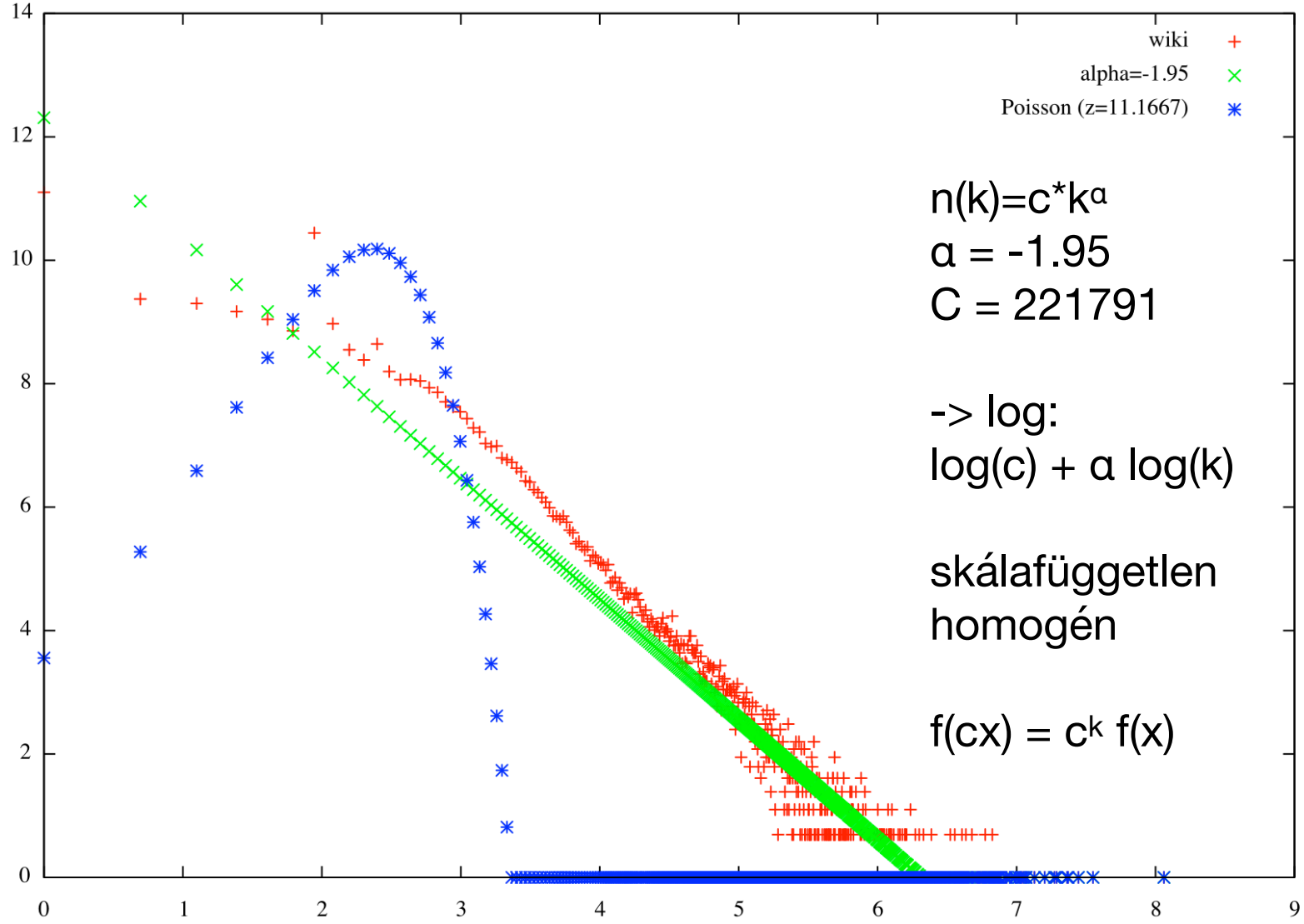


Wiki graph is hatványeloszlás (vagy?)!





Wiki graph is hatványeloszlás (vagy)!



$n(k)=c*k^{\alpha}$

$\alpha = -1.95$

$C = 221791$

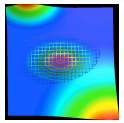
-> log:

$\log(c) + \alpha \log(k)$

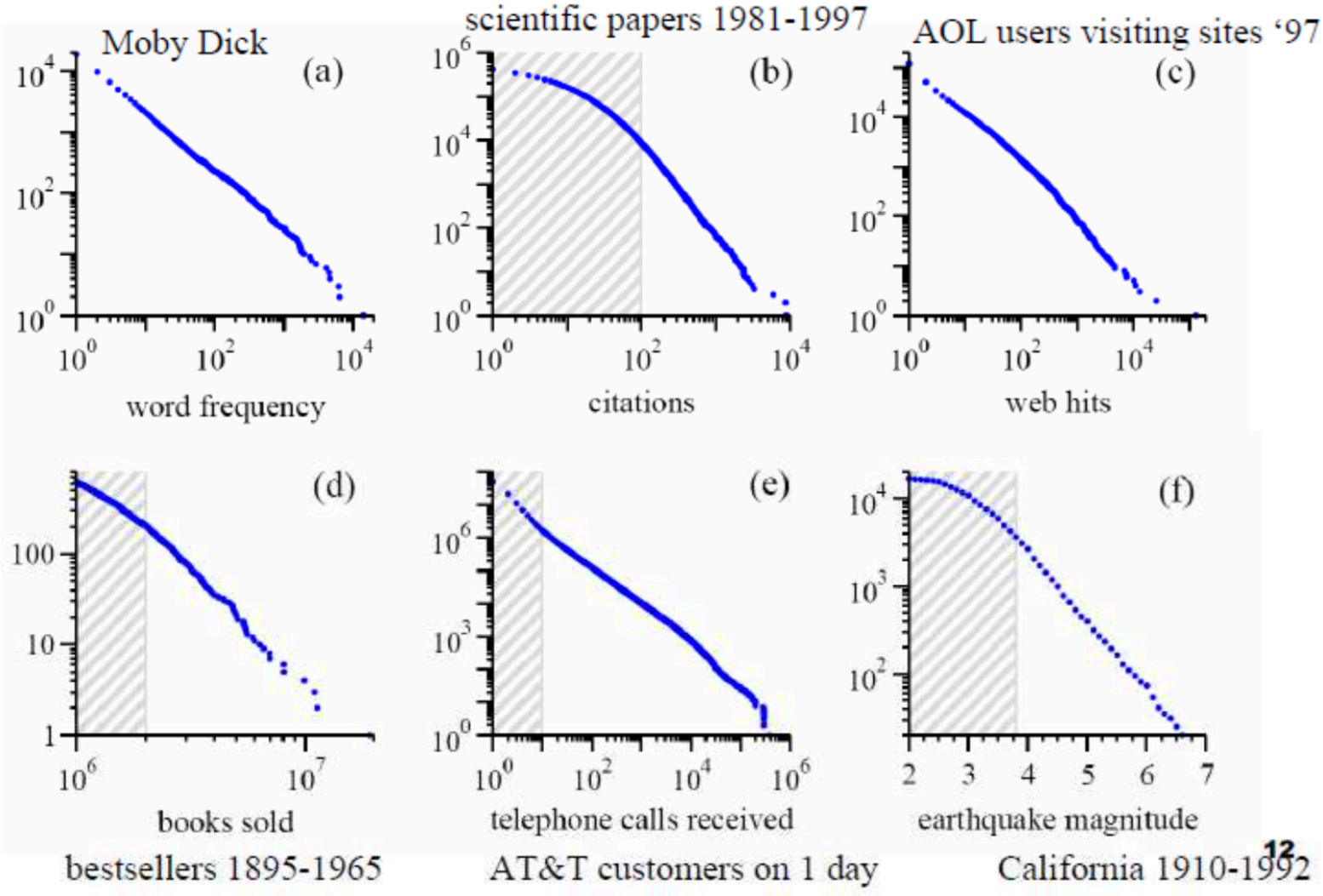
skálafüggetlen

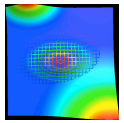
homogén

$f(cx) = c^k f(x)$



Példa hatványeloszlásokra (ábrák: Daniel Bilar)





Sokszor felfedezték

Pareto (1897): 80-20 szabály

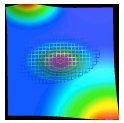
Yule (1925): evolúció

Zipf (1949): szóeloszlás

Simon (1955): Zipf alapján

Price (1976): hivatkozási gráf!

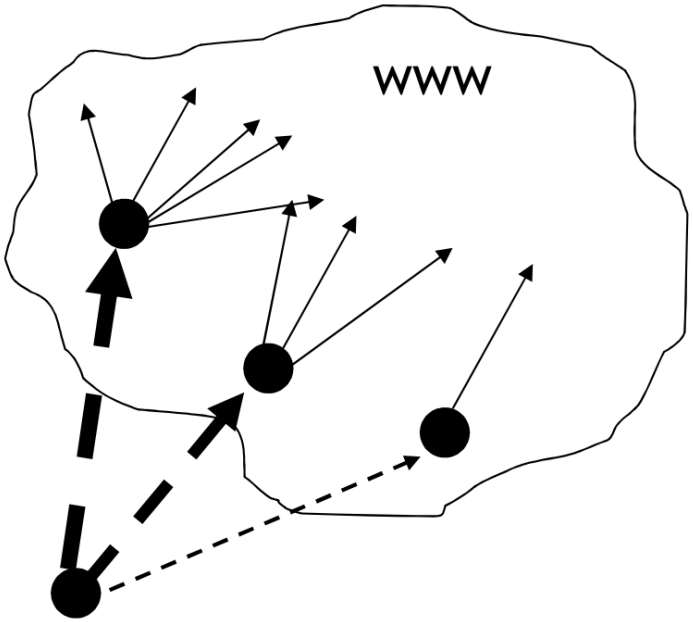
és Barabási-Albert modell (1999): WWW gráf fokszámeloszlása



Örülünk Vincent, de kéne valamivel modelleznünk

Preferential attachment

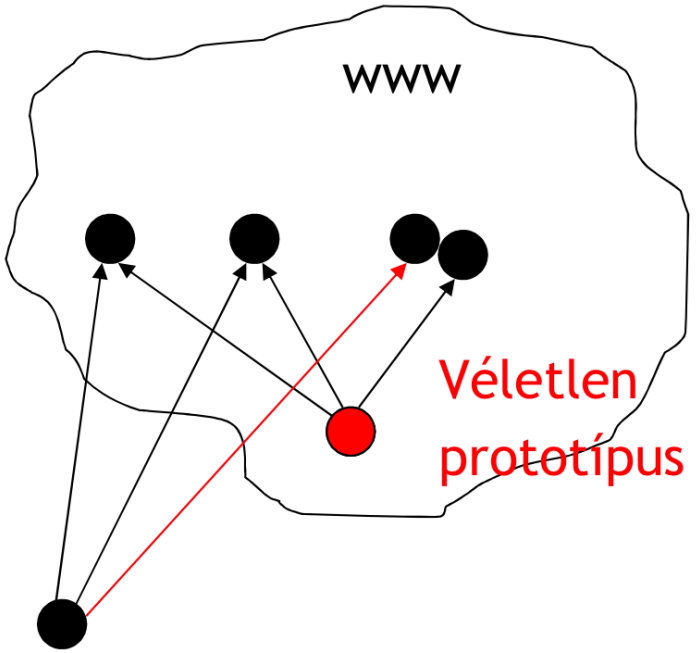
(Albert, Barabási 1999)



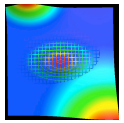
Paraméter: m éllel kapcsolódik egy új pont

Evolving copy

(Broder et al., 2000)



Paraméter: másolás minősége



Barabási-Albert modell

1. A modell minden lépésben egy új csúccsal bővíti a meglévő gráfot

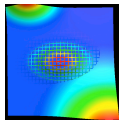
2. Majd a meglévő fokszámok alapján kiválaszt m csúcsot és összeköti az új csúccsal

A kapott hatványeloszlás (i-dik időpillanatban keletkezett csúcs t időpontbeli fokszámára vontakoztatva):

$$P(k_i(t) = k) = m^2 t\left(\frac{1}{k^2} - \frac{1}{(k-1)^2}\right) \sim k^{-3}$$

1. Növekednie kell folyamatosan (ER nem!)
2. A növekedésnek BA szerint kell megtörténnie

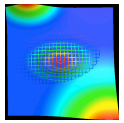
Ha bármelyik nem igaz, nem lesz PL!



Készen vagyunk?

Már van hatványeloszlásunk!

De egy valós hálózat más tulajdonságokkal is rendelkezik:



Készen vagyunk?

Már van hatványeloszlásunk!

De egy valós hálózat más tulajdonságokkal is rendelkezik:

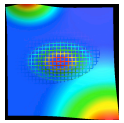
Kis világ modell (Watts és Strogatz)

1. Alacsony átlagos távolság

    Régóta ismert jelenség, hogy pár emberen keresztül mindenkit ismerünk

    Általában log(N)-el arányos

Nem szorosan:
Friendship paradoxon (Scott L. Feld 1991):
legtöbb embernek kevesebb barátja van, mint a barátainak átlagosan



Készen vagyunk?

Már van hatványeloszlásunk!

De egy valós hálózat más tulajdonságokkal is rendelkezik:

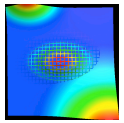
Kis világ modell (Watts és Strogatz)

1. Alacsony átlagos távolság

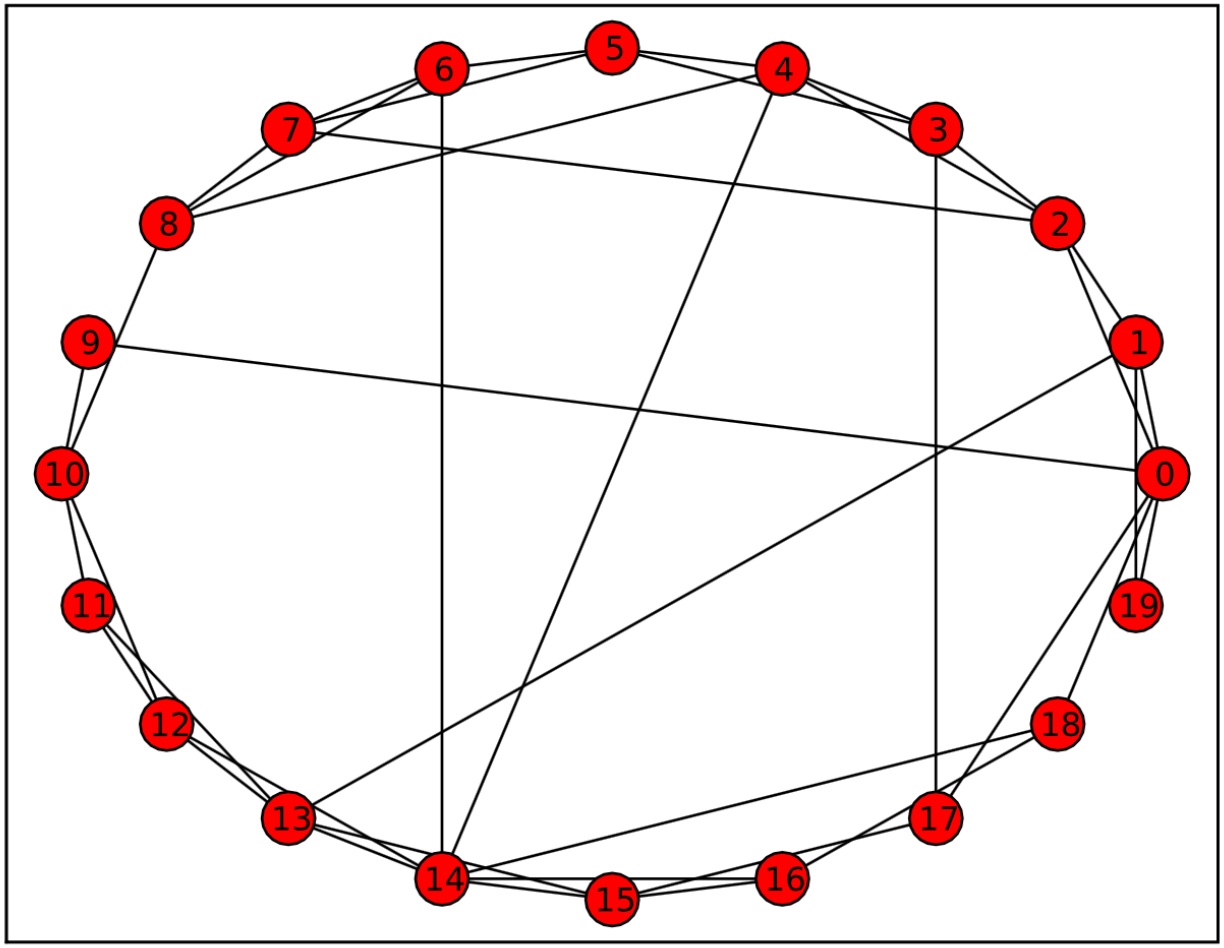
2. Erős klaszterezettség:
   A csúcsok szomszédai átlagosan “erősen” összefüggenek

   külön csúcsokra (k_i a kifok):

$$C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$$



Watts-Strogatz modell

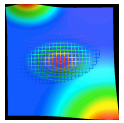


r reguláris gyűrű

Ebből véletlen veszünk el és kötjük be ER szerűen máshova

Eredmény: kis-világ

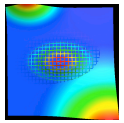
De nem hatványeloszlás!



Összefoglalva

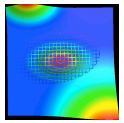
	Fokszámeloszlás	Klaszterezettség	Átlagos távolság
Valós hálózatok	Hatványeloszlás	erősen	kicsi
Erdős-Rényi	Poisson	gyengén	kicsi
Barabási-Albert	Hatványeloszlás	gyengén	kicsi
Watts-Strogatz	Poisson	erősen	kicsi
Broder et al.	Hatványeloszlás	erősen	kicsi

Forrás: Benczúr András



Random Forest (Breiman, 2001)

Döntési fa vagy erdő?



Random Forest (Breiman, 2001)

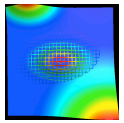
Döntési fá vagy erdő?

Bagging (Breiman , 1996):

- újramintavételezés -> modell aggregáció

Miért?

-> DT esetében?



Random Forest (Breiman, 2001)

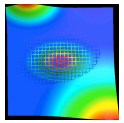
Zajos attribútumok?

-> minden vágás előtt mintavételezés: m attribútum kiválasztása (M-ből)

Random Forest:

1. Bagging: 100-500
2. Fa építés
    1. Minden levelet vágunk a mintavételezett attribútumhalmaz alapján
3. Aggregáció: pl. többbségi döntés, várható érték stb.

Mi a gyakorlati különbség a DT és az RF között?



AdaBoost

Freund and Schapire (1995):

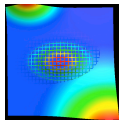
Adaptive boosting

"Weak" ("rosszul" teljesítő) modellek halmaza, lineáris kombinált

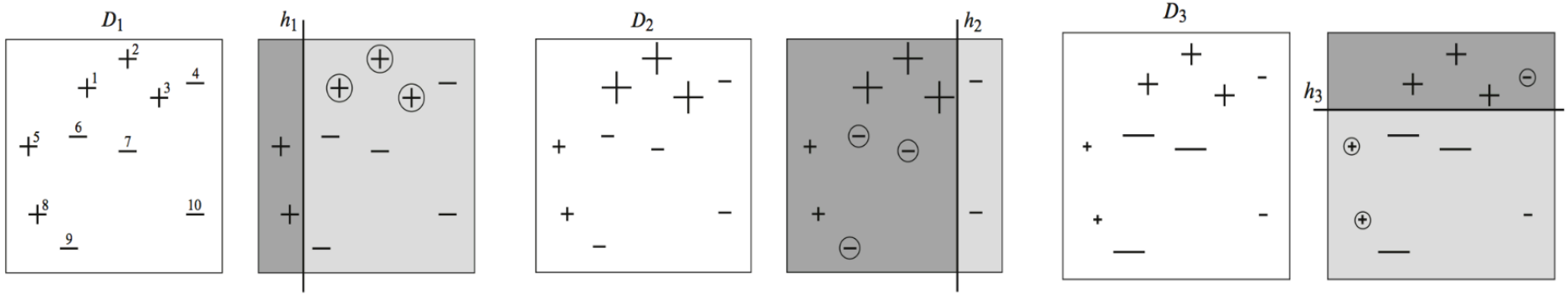
Meta osztályozó

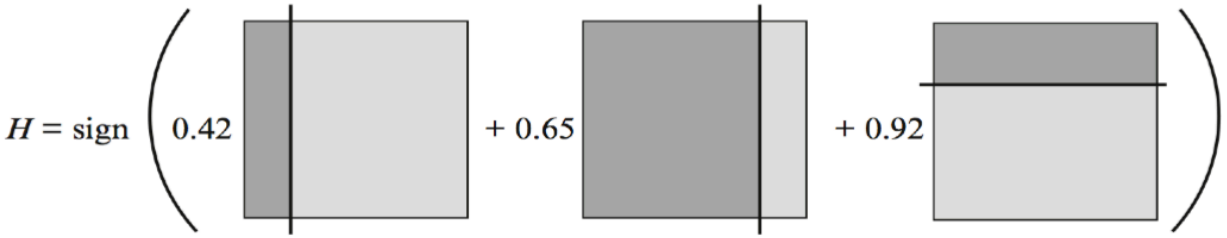
Továbbfejlesztései: Gradient Boosting, Gradient Boosted Tree or LogitBoost.

Kaggle: xGBT



AdaBoost





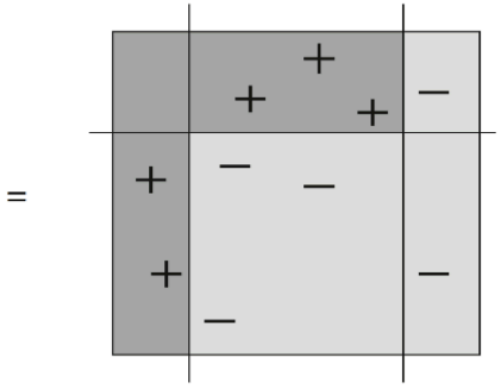
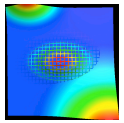


Fig.: Freund & Schapire



AdaBoost

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, \ldots, m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathcal{X} \to \{-1, +1\}$.
- Aim: select h_t with low weighted error:

$$\varepsilon_t = \Pr_{i \sim D_t} \left[h_t(x_i) \neq y_i \right].$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$.
- Update, for $i = 1, \ldots, m$:

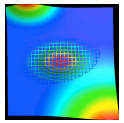
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

src: Schapire



AdaBoost: Viola Jones

Haar like attribútumok
162k potenciális terület

AdaBoost Decision stump ->
<100 features

Cascade osztályozó

Hogyan lehet gyorsan kiszámolni?

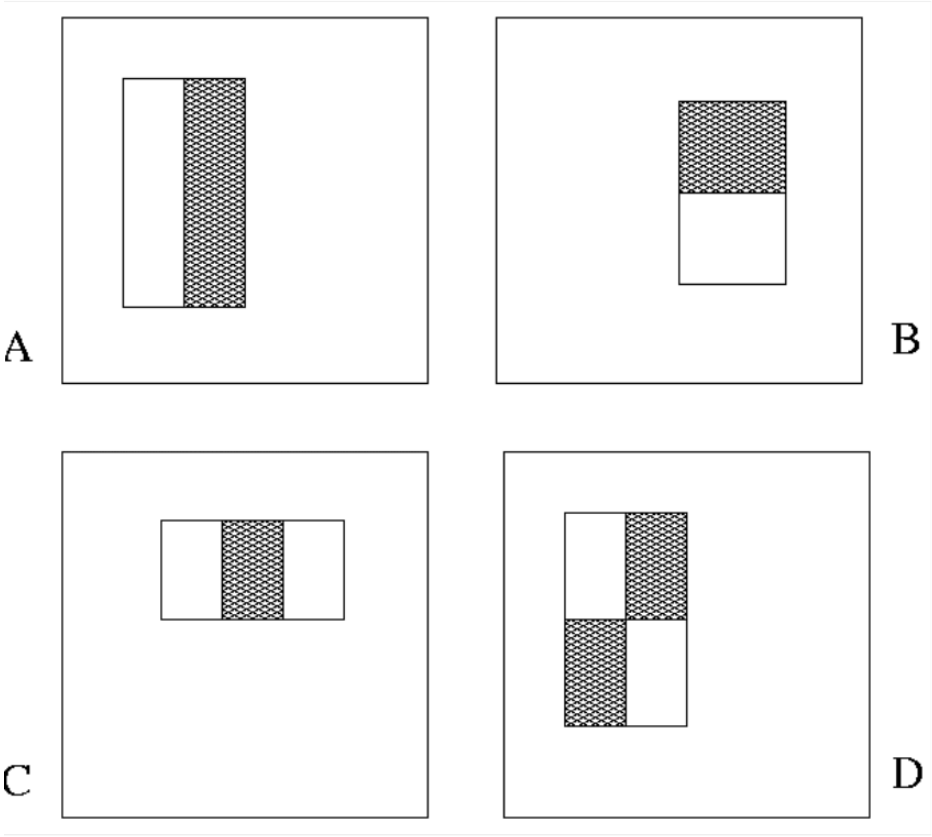
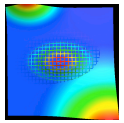
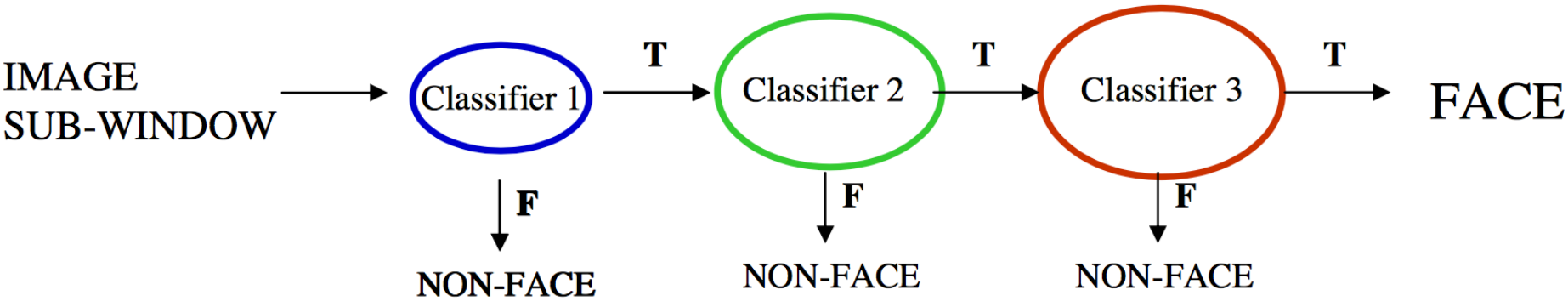


Fig.: Viola&Jones



Viola Jones detector



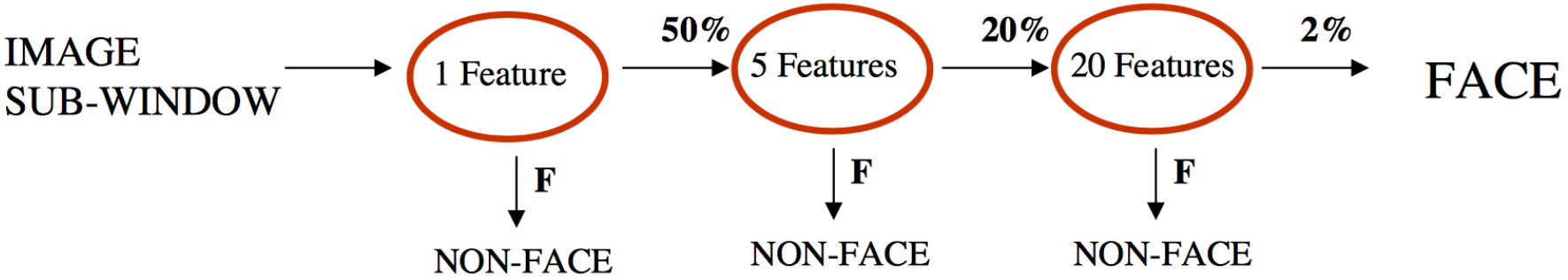
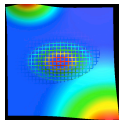


Fig.: Viola&Jones



Viola Jones detector

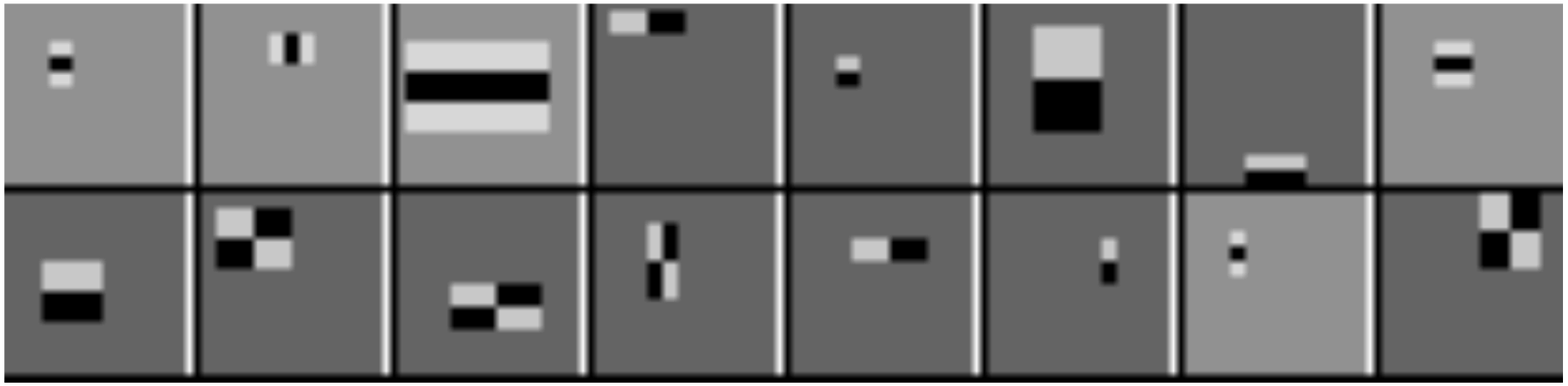
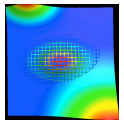
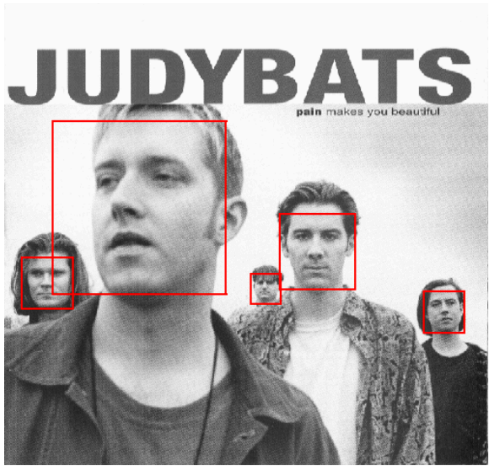
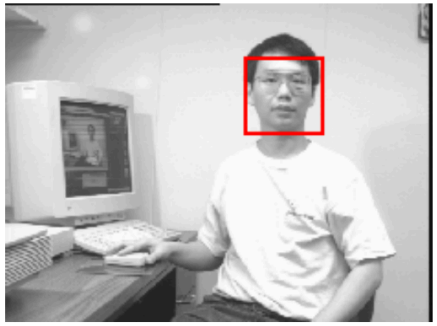


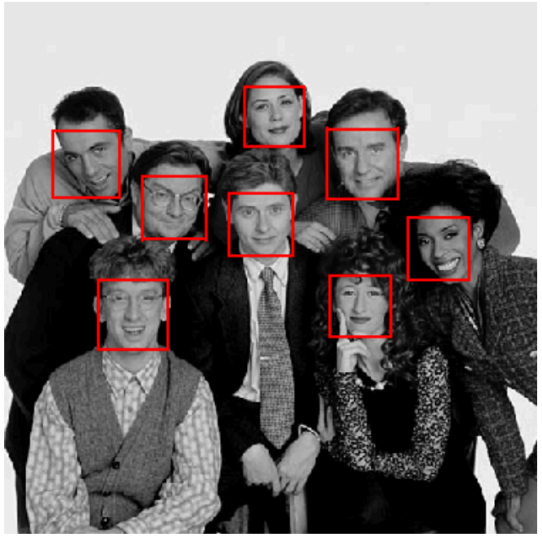
Fig.: Viola&Jones



Viola Jones detector







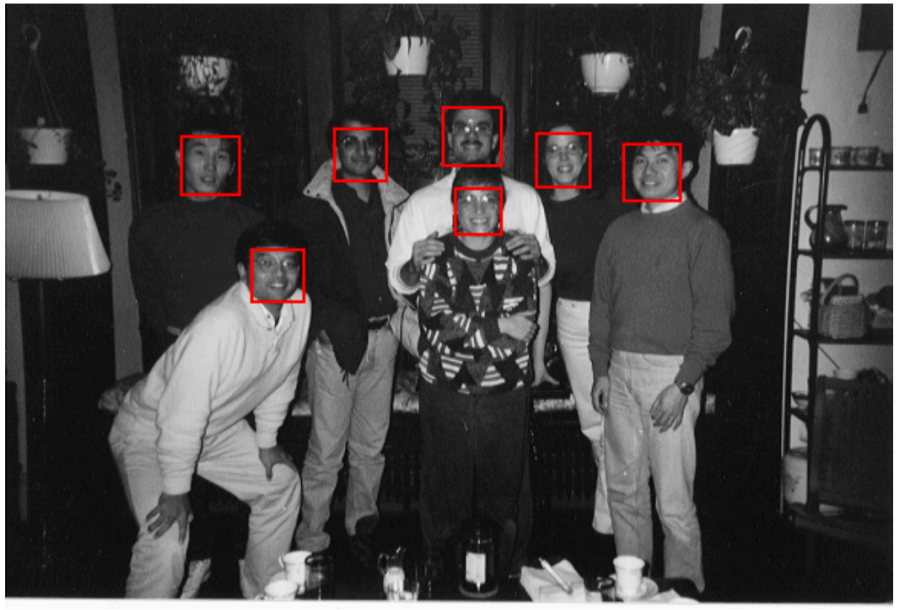
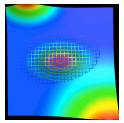


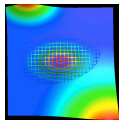
Fig.: Viola&Jones



Gradient Boosted Trees (Friedman, Hastie, Tibshirani)

AdaBoost: decision stump mint gyenge ("weak") osztályozó

Miért DS?



Gradient Boosted Trees

Regressziós fák! (DT + valós kimenet)

Predikció:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

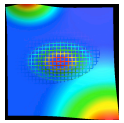
Vagy (n a tanulóhalmaz számossága):

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

"Error" (bal) és komplexitás (jobb)

VC tételek -> alacsony komplexitás

src.: Chen



Gradient Boosted Trees

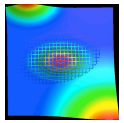
Hogyan lehet optimalizálni?

Iteratívan:

$$
\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}
$$

Tegyük vissza az objektív függvénybe ☺️

src.: Chen



Gradient Boosted Trees

Eredmény:

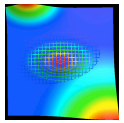
$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant$$

Már csak egy loss függvényre van szükségünk.

RMSE?

Hogy lehet optimalizálni RMSE-re?

src.: Chen



Gradient Boosted Trees

Megoldás: Taylor sorba fejtjük az objektív függvényt!

Eredmény:

$$Obj^{(t)} \simeq \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + constant$$

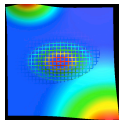
ahol

$$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$$

és

$$h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$$

src.: Chen



Gradient Boosted Trees

Komplexitás?

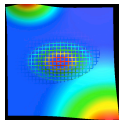
Attól függ…. De:

1. a pontok súlya
2. a fák mérete?

Próbáljuk ki a 'person.txt’-n (otthon):

scikit ensemble (AdaBoost és RandomForest)
xGBoost ("Kaggle’s favourite flawor")

src.: Chen

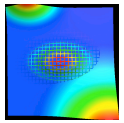


Klaszterezés

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 (n=3 -> 5)
- épp ezért a klaszterzés egyik fő paramétere a klaszterek száma

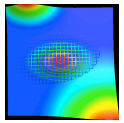
Amennyiben a klaszterezés végeredménye diszjunkt csoportok hard klaszterezésről beszélünk (pl. k-közép (k-means)). Szoft (vagy gyenge) klaszterezés esetében csak azt várjuk el, hogy minden (pont,klaszter) párra egy a klaszterba tartozástól függő mértéket rendeljünk.



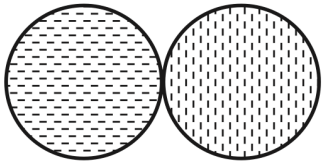
Klaszterezés

Klaszterezési algoritmus kiválasztása előtt érdemes az adatot is megfigyelni:

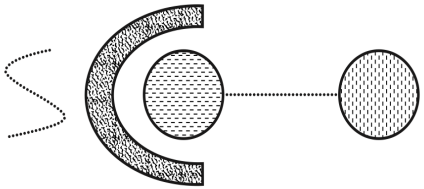
1. Jól szeparált csoportok (a legtöbb módszer jó)
   Minden egy csoportba tartozó elem közelebb van a csoport többi eleméhez mint bármilyen más csoportba tartozó elem
2. Középpont alapú csoportok (pl. kmeans)
   Minden csoportnak meghatározhatunk egy középpontot, melyhez minden adott csoportba tartozó elem közelebb van, mint más csoportok középpontjai
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)



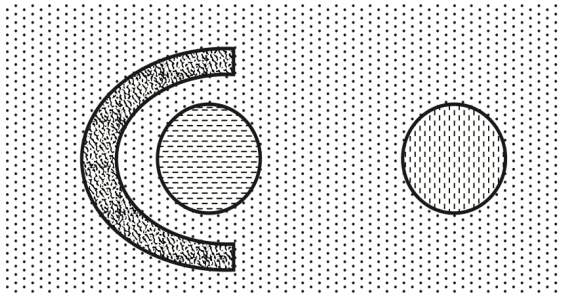
1. Jól szeparált csoportok (a legtöbb módszer jó)
   Minden egy csoportba tartozó elem közelebb van a csoport többi eleméhez mint bármilyen más csoportba tartozó elem
2. Középpont alapú csoportok (pl. k-means)
   Minden csoportnak meghatározhatunk egy középpontot, melyhez minden adott csoportba tartozó elem közelebb van, mint más csoportok középpontjai
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) ?

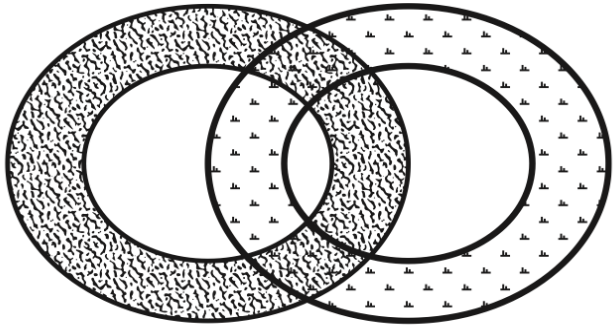


b) ?

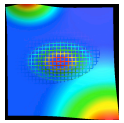


c) ?

d) ?



e) ?



k-Means

Feltesszük, hogy az adatpontjaink egy vektortérben helyezkednek el.

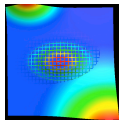
A klasztereket a középpontjuk (súlypontjuk) határozza meg.

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 közepe
5 for minden u Є D
6 do u legyen az argmin_i d(u; r(Ci)) indexű klaszterben
7 C az új klaszterezés
8 return C

A kezdőpontok meghatározása lehet:
a) véletlen pontokból
b) véletlen választott tanulópontokból

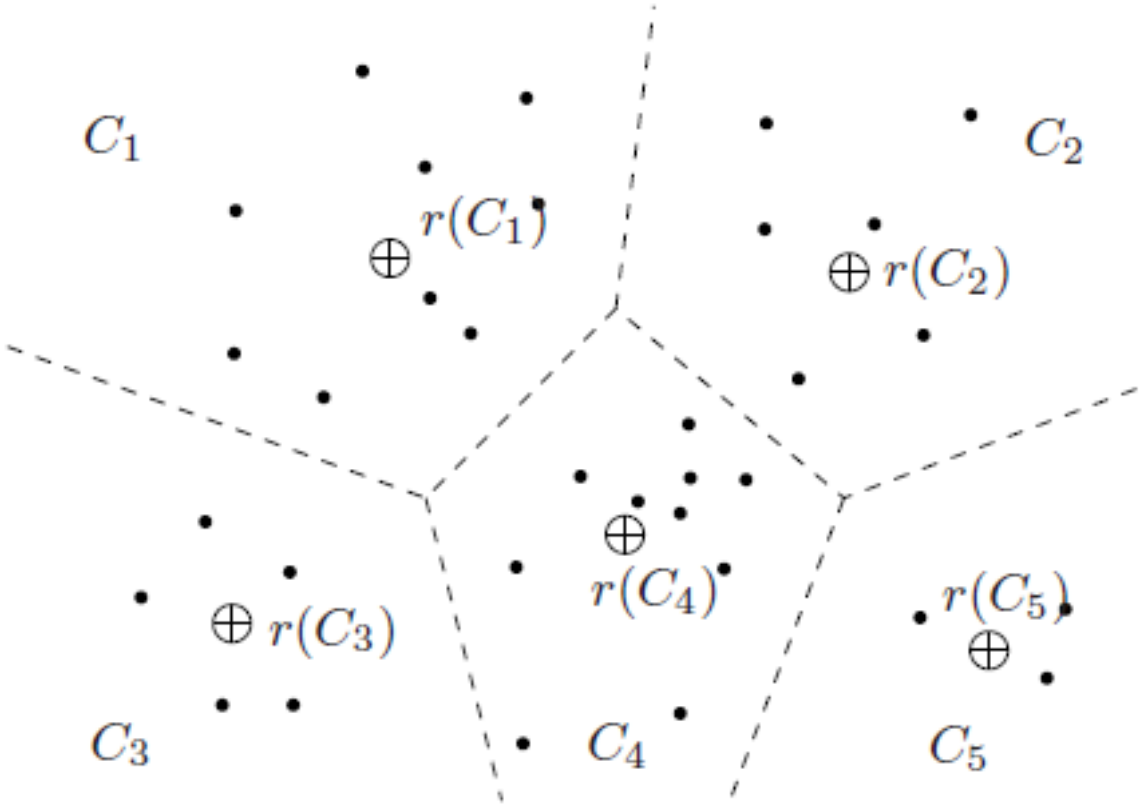
$$Err_{squared}(D,k)=\sum_{i=1}^{D}(x_i-c_i)^2$$

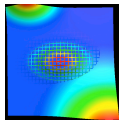
Az algoritmust megállítjuk, ha:
a) a klaszterezés nem változik
b) a küszöbhiba meghaladja a négyzetes hiba mértékét
c) elértük a maximális iterációk számát



k-Means

Sajnos csak egy lokális minimum-ot találtunk!





k-Means

Előnye, hogy a futásidő N*K*it ahol N a pontok, K a klaszterek, it pedig az iterációk száma

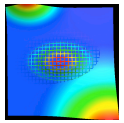
Zajos adatokon nem hatékony.

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 modellt 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.

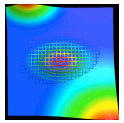


Bisecting k-Means

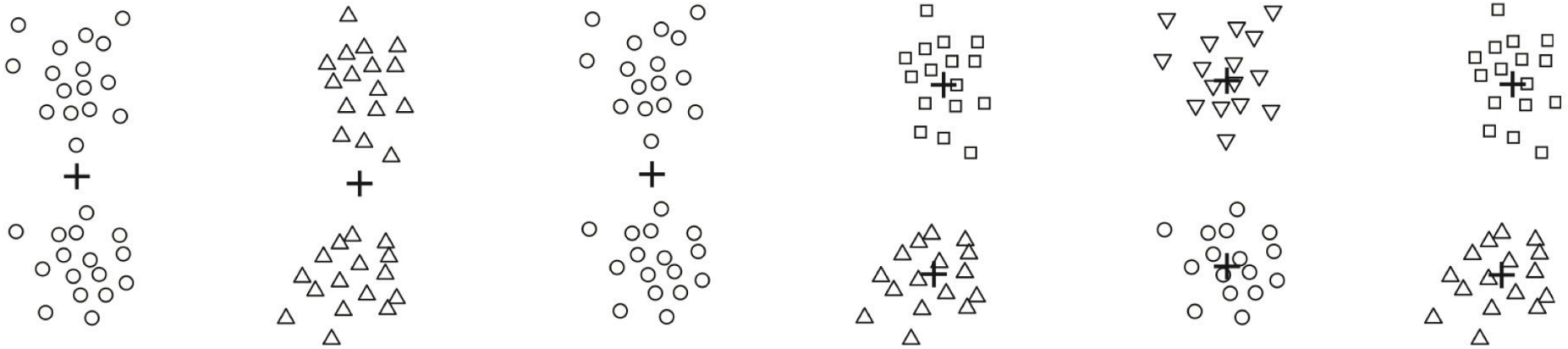
```
Bisecting K-means (D; k)
1     Legyen r(C1), r(C2),... , r(Ck') reprezentánsok tetszőleges k'<k elemű kezdeti
halmaza
2      while el nem értük a kívánt klaszterszámot
3            do válasszunk ki egy meglévő klasztert
4                  do n próbán keresztül
5                        k-means a klaszterbe tartozó pontokon
6                  válasszuk ki a legjobb klaszterezést a próbák közül
7            Cseréljük le a kiválasztott klasztert az új klaszterekre
8     return C
```

Hogyan válasszuk ki a legmegfelelőbb klasztert?

Legnagyobb SSE?
Méret?
Mindkettő?



Bisecting k-Means

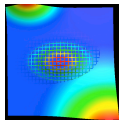


1. iteráció

2. iteráció

3. iteráció

Utólagos finomítás: k-Means



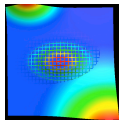
k-Medoid

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 össztávolságot (ez a költség)
3. Minden klaszter medoid 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-Means-nél?



Sűrűség alapú módszerek

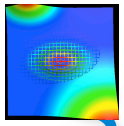
A k-Means egyik legnagyobb hibája, hogy akkor is K klasztert fog létrehozni ha nincs rá szükség.

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

Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu

DBSCAN: Density-Based Spatial Clustering of Applications with Noise

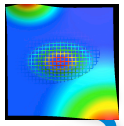
A pontok halmaza P
Szomszédossági reláció:

$N_\Theta = (\{(p,q) \in PxP \} \mid sim(p,q) > \Theta)$ vagy $N_\Theta(p) = (\{(p,q) \in PxP \} \mid dist(p,q) < \Theta)$

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

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

p sűrűsége: $| N_\Theta(p) |$



Sűrűség alapú módszerek DBSCAN

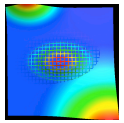
DBSCAN:

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

A p pont közvetlenül elérthető a q pontból, ha
1. p q szomszédja és
2. q magpont

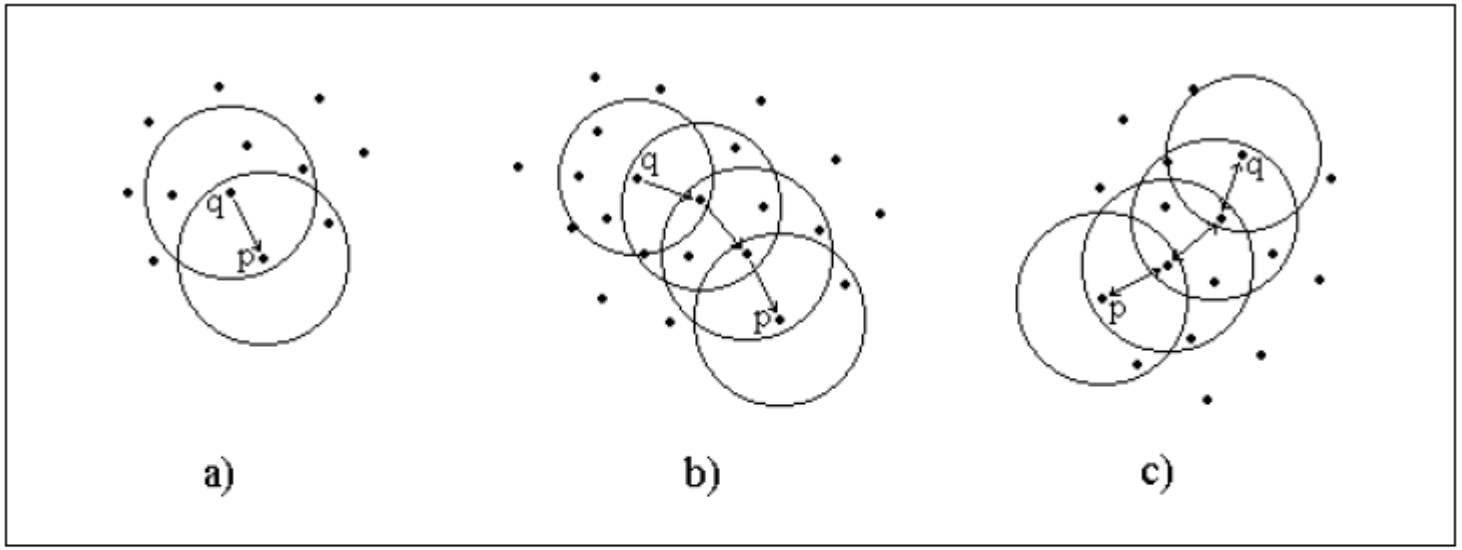
A p pont elérhető q pontból, ha
Létezik olyan $q=p_1,p_2, \ldots p_n,p_{n+1}=p$ sorozat, ahol minden i-re p_{i+1} közvetlenül elérhető p_i-ből

A p,q összekapcsoltak, ha létezik r Є P , melyre igaz, hogy 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 Є C összekapcsolt
2. (maximálisság) minden p Є P és q Є C , ha p elérhető q-ból akkor p Є C

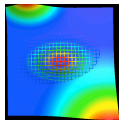


Példa: MinPts = 4

a) *p* közvetlenül elérhető *q*-ból (*q* magpont, *p* határpont)

b) *p* elérhető *q*-ból (*q* magpont, *p* határpont)

c) *p* és *q* összekapcsolt (mindkettő határpont)



Kapcsolati háló alapú klaszterezés

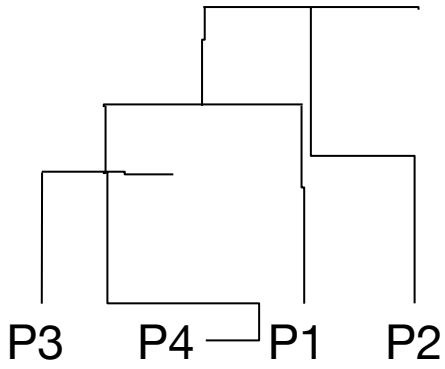
Két klaszter hasonlósága

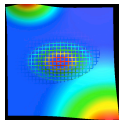
a) leghasonlóbb elemeiknek a hasonlósága (single-link)
b) legkevésbé hasonló elempárjaik hasonlósága (complete-link)
c) elemeik átlagos hasonlósága (average-link)

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: dendogram (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



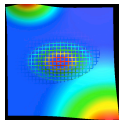


Példa

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

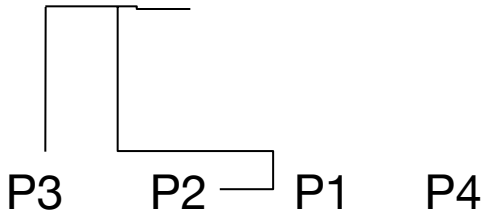


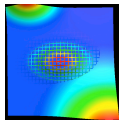
Single Link

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



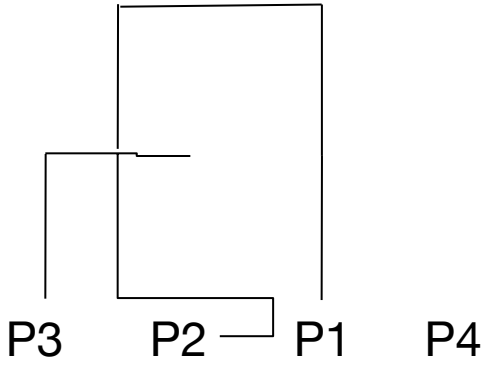


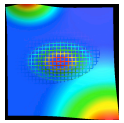
Single Link

Rajzoljunk dendogram-ot a következő példára!

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

Hasonlóság	P1	{P2,P3}	P4
P1	1	0.8	0.7
{P2,P3}	0.8	1	0.2
P4	0.7	0.2	1



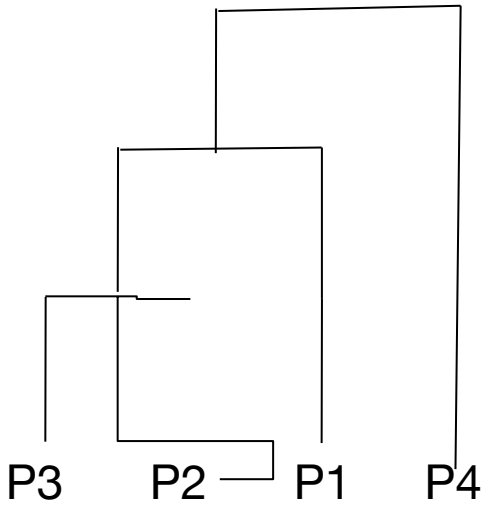


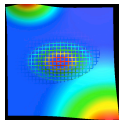
Single Link

Rajzoljunk dendogram-ot a következő példára!

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

Hasonlóság	{P1,P2,P3}	P4
{P1,P2,P3}	1	0.7
P4	0.7	1



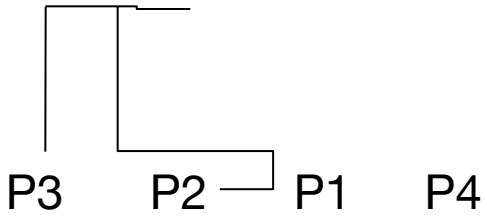


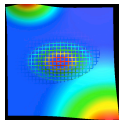
Complete Link

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



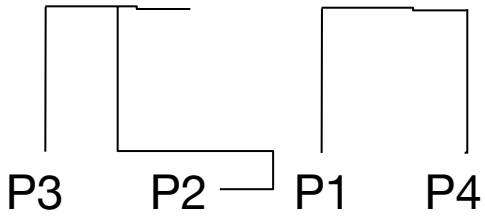


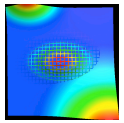
Complete Link

Rajzoljunk dendogram-ot a következő példára!

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

Hasonlóság	P1	{P2,P3}	P4
P1	1	0.2	0.7
{P2,P3}	0.2	1	0.1
P4	0.7	0.1	1



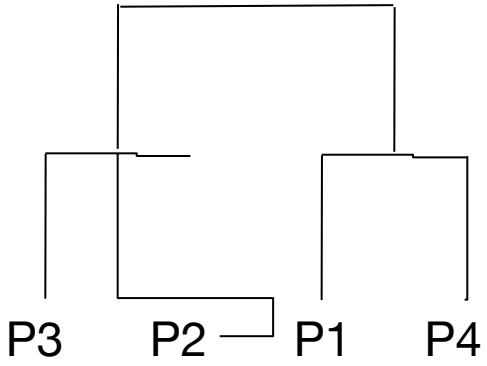


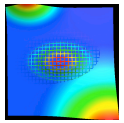
Complete Link

Rajzoljunk dendogram-ot a következő példára!

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

Hasonlóság	{P2,P3}	{P1,P4}
{P2,P3}	1	0.1
{P1,P4}	0.1	1



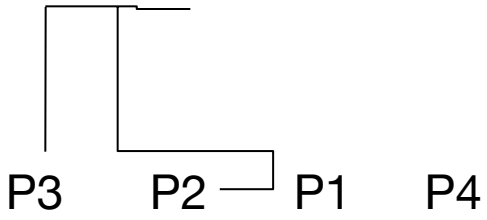


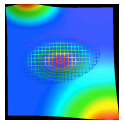
Average Link

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



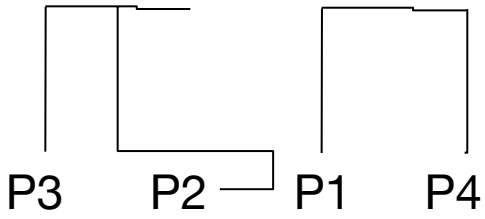


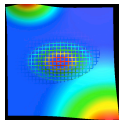
Average Link

Rajzoljunk dendogram-ot a következő példára!

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

Hasonlóság	P1	{P2,P3}	P4
P1	1	0.5	0.7
{P2,P3}	0.5	1	0.15
P4	0.7	0.15	1



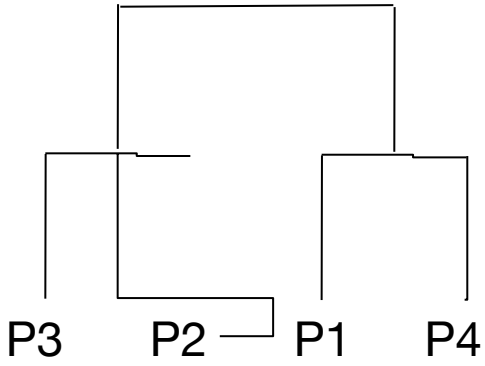


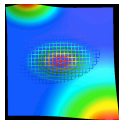
Average Link

Rajzoljunk dendogram-ot a következő példára!

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

Hasonlóság	{P2,P3}	{P1,P4}
{P2,P3}	1	0.325
{P1,P4}	0.325	1





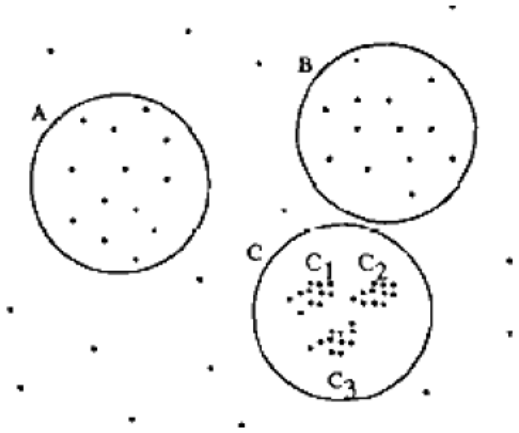
Sűrűség alapú módszerek OPTICS

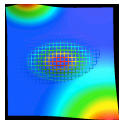
A DBSCAN feltételezi, hogy a sűrűség az egész adatra hasonló. (globális sűrűség)

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 vállnak 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

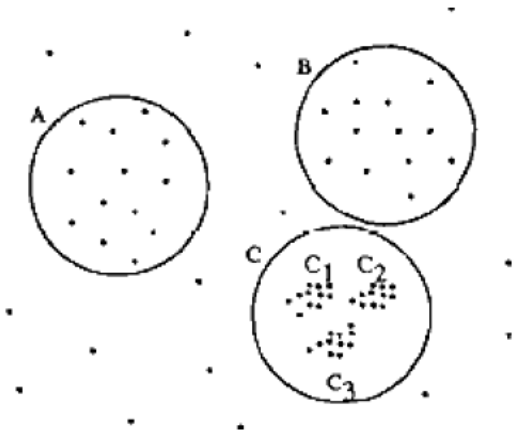




Sűrűség alapú módszerek OPTICS

N(p) a p eps sugarú környezete: N(p) = {q Є P | d(p,q) < eps }
p Є P magpont , ha N(p) >= MinPts
p Є P határpont, ha N(p) < MinPts
p Є P közvetlenül elérhető q Є P pontból, ha

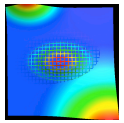
1. p Є N(q) és
2. q magpont



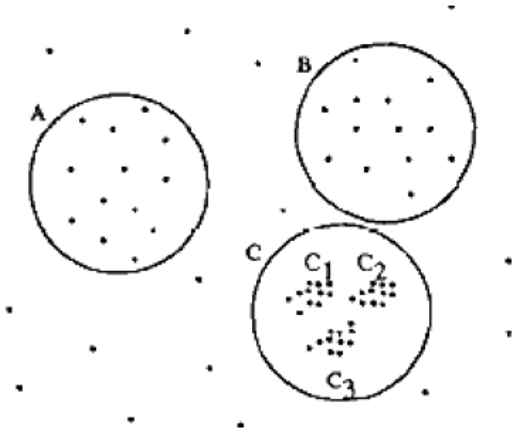
Legyen egy p Є P pont k-távolsága az a d(p,q) távolság, melyre igaz:

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

p Є P magpont magtávolsága megegyezik MinPts (k=MinPts) távolságával p-nek
p Є P elérhető távolsága o Є P magponttól max(magtávolság(o),d(p,o))

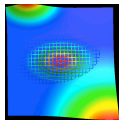


Sűrűség alapú módszerek OPTICS



OPTICS Algoritmus:

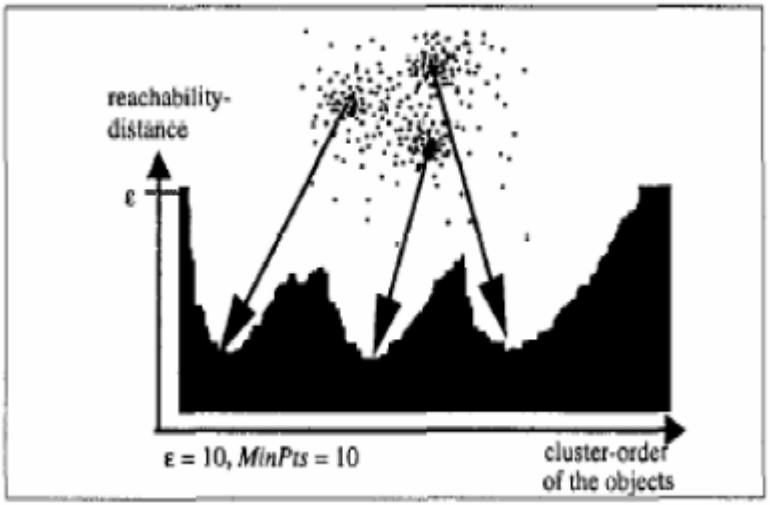
1. vegyünk egy eddig nem vizsgált elemet
2. ha az elem nem magpont, akkor berakjuk a kimeneti halmazba
3. ha az elem magpont , akkor a szomszédai bekerülnek a bővítési halmazba. Maga a pont a kimeneti halmazba kerül
4. meghatározzuk minden bővítései halmaz elemének a kimeneti halmaztól vett legkisebb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

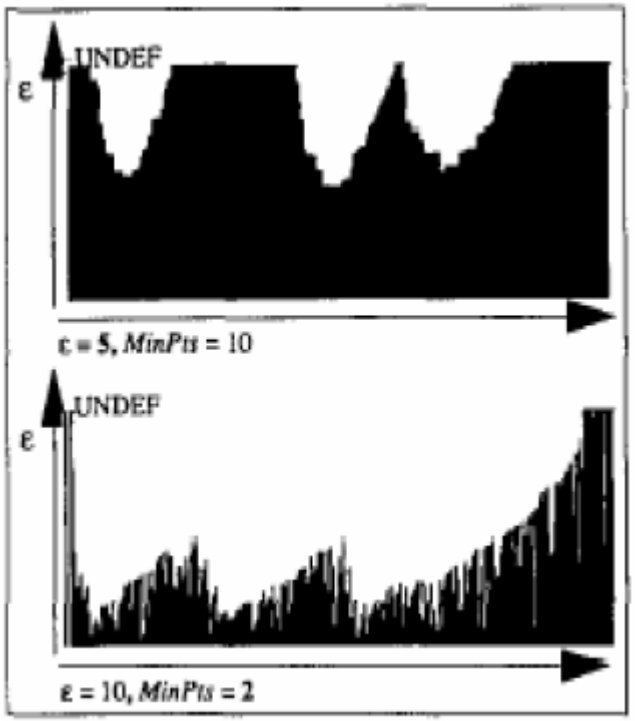


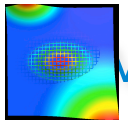
Sűrűség alapú módszerek OPTICS

OPTICS kimenete

A gödrök a klaszterek
Az epsilontól erősen függ a kimenet, de a MinPts-től nem

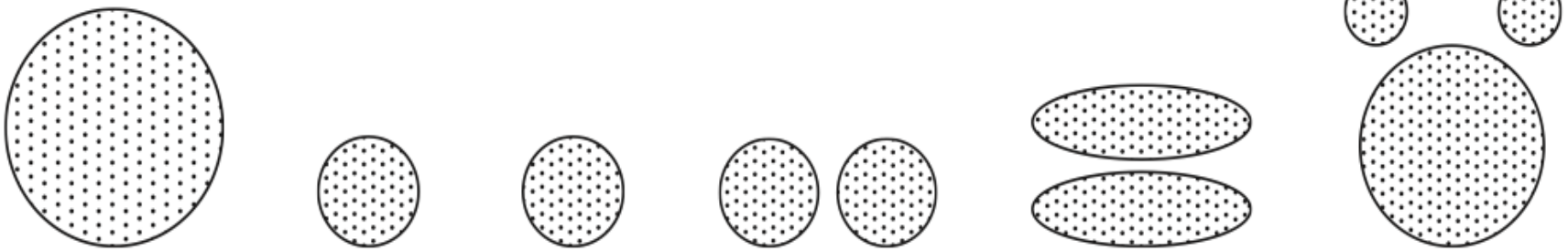




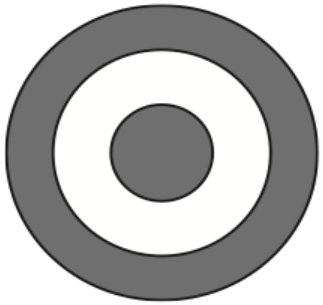


Mit tesz egy sűrűség,egy center és egy kapcsolat alapú klaszterező algoritmus?

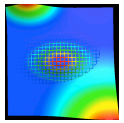
(feltételezzük, hogy a pontok sűrűsége a mintában állandó)



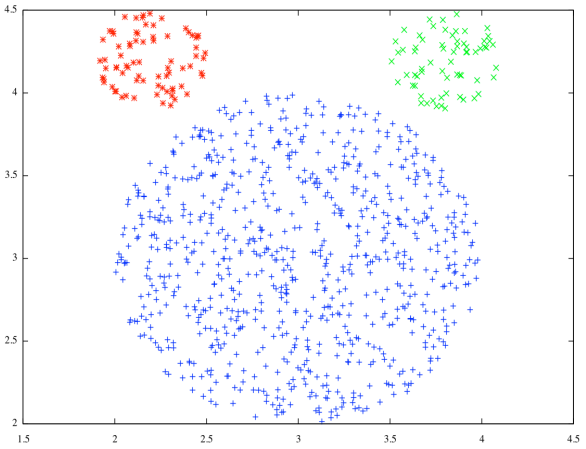
a) b) c) d) e)

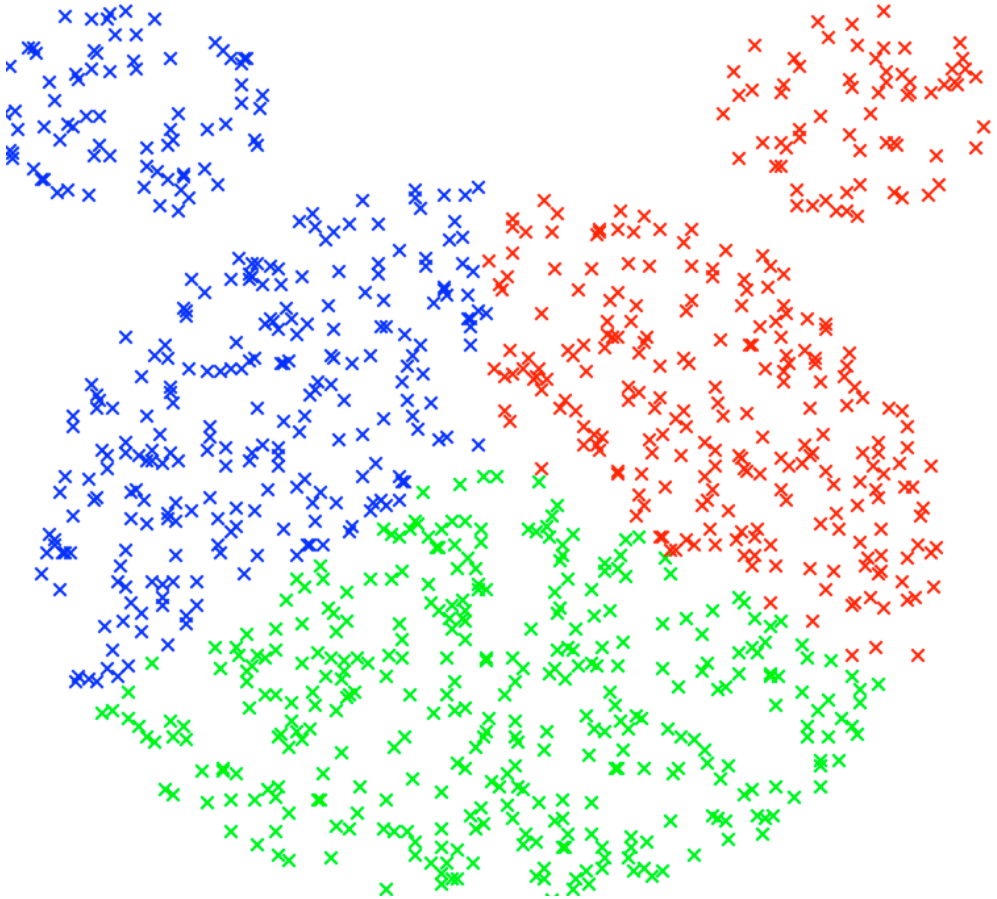


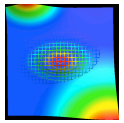
f)



Mickey mouse







Gaussian Mixture Model

Az e) példa megoldása nem sűrűség alapú módszerrel.

Ellentétben a K-means algoritmussal ahol konvex, diszjunkt klaszterek megtalálása a cél a GMM esetében keresett klaszterek a probléma terében értelmezett normális eloszlások.

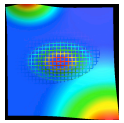
Az adott normális eloszlásokat várható értékükkel (vektor!) és kovariancia mátrixukkal jellemezzük. A K-means-hez hasonlóan előre meghatározott véges számú klaszteren értelmezzük. (legyen N a normális eloszlások száma)

Legyen a kervert sűrűségfüggvény d dimenziós vektortérben, N normális eloszlás mellett:

$$p(x \mid \Theta) = \sum_{i=1}^{N} \omega_i g_i(x) \qquad g_i(x) = \frac{1}{\sqrt{(2\Pi)^d \mid \Sigma_i \mid}} \exp^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}$$

ahol

$$\Theta = \{\omega_1, .., \omega_N, \mu_i, .., \mu_N, \Sigma_1, .., \Sigma_N\}$$



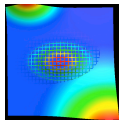
Gaussian Mixture Model

Keressük azt a Θ modelt, mely a leginkább illeszkedik a tanuló ponthalmazunkra. Ebben az esetben a mixture modelünk minden x tanulóhalmazbeli pont felett lokálisan maximális. A logaritmus függvény szigorúan monoton tulajdonsága miatt a következő függvény maximumát keressük:

$$\mathcal{L}(X) = \log p(X \mid \Theta) = \log \Pi_{t=1}^{T} p(x_t \mid \Theta) = \sum_{t=1}^{T} \log p(x_t \mid \Theta)$$

Ennek egy iteratív megoldása lehet pl. az EM (Expectation Maximization) algoritmus. Vezessünk be egy látens, segítő arányt (“membership probability”):

$$\gamma_i^{(k)}(x_t) = \frac{\omega_i^{(k-1)} g_i^{(k-1)}(x_t)}{\sum_{j=1}^{N} \omega_j^{(k-1)} g_j^{(k-1)}(x_t)}$$



Gaussian Mixture Model

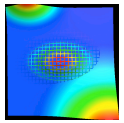
Paraméter-tér dimenziója: K*(1+D+D*D)

Az EM algoritmus menete:

E-step: Az előző iterációban megállapított valószínűség:

$$\gamma_i^{(k)}(x_t) = \frac{\omega_i^{(k-1)} g_i^{(k-1)}(x_t)}{\sum_{j=1}^{N} \omega_j^{(k-1)} g_j^{(k-1)}(x_t)}$$

Ebben az úgynevezett “Expectation” lépésben T*N darab valószínűséget határozunk meg.



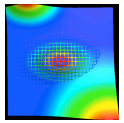
Gaussian Mixture Model

M-step: Módosítsuk a paraméter-terünket, hogy a kezdeti feltétel szerint jobban modellezze a tanulóhalmazt.

$$\mu_{id}^{(k)} = \frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t) x_{td}}{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)}$$

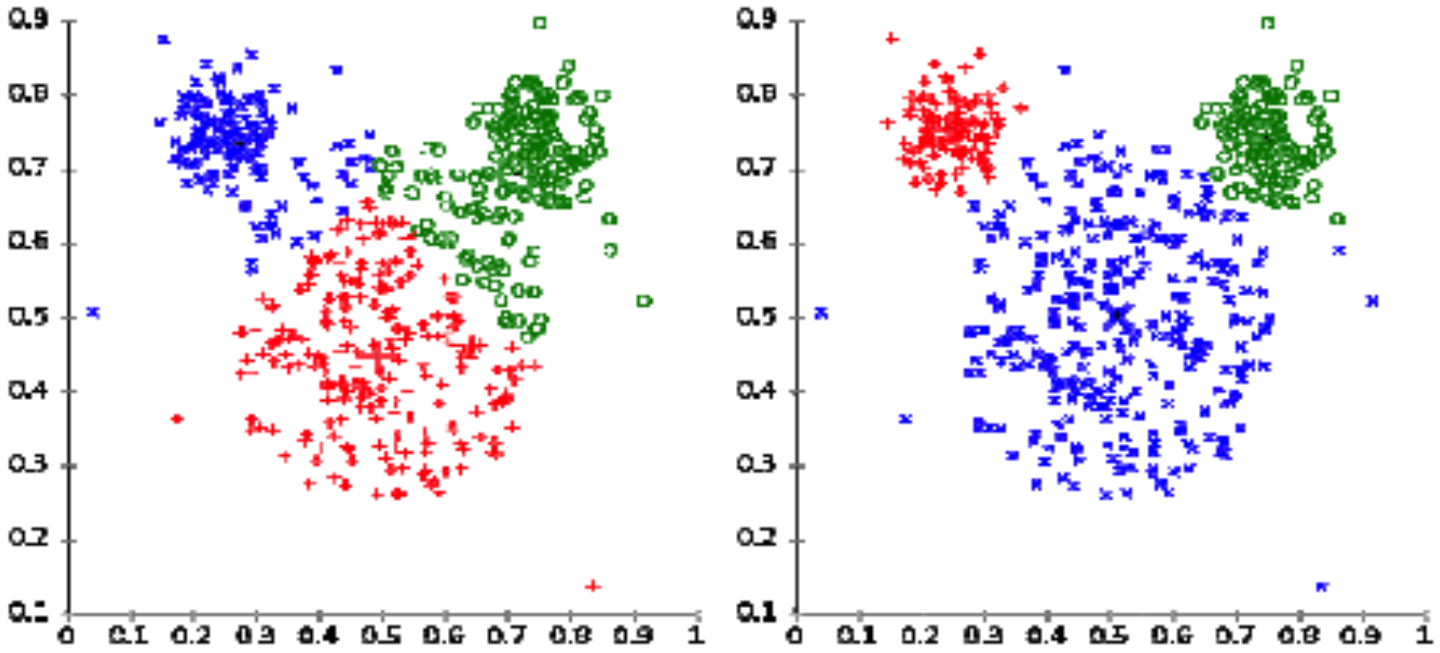
$$\sigma_{id}^{(k)} = \sqrt{\frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)(x_{td} - \mu_{id}^{(k)})^2}{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)}}$$

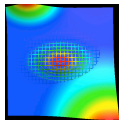
$$\omega_i^{(k)} = \frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)}{T}$$



Gaussian Mixture Model

Egy példa ahol a k-Means-el szemben a GMM képes megfelelően megtalálni a klasztereket. A GMM a K-means-el ellentétben jól kezeli a “gömb”-szerű, de különböző méretű klasztereket.





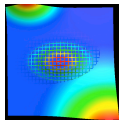
Gaussian Mixture Model

A GMM algoritmus előnye, hogy ötvözi a K-means előnyös tulajdonságait nagydimenziós terekben a szoft klaszterezők előnyeivel.

- könnyen kezel nehezen besorolható elemeket
- robosztusabb zajokra
- kifinomultabb bag-of-words modellek építhetők segítségével
- kevesebb klaszter is elég a k-Means-el szemben
  (256 vs. 4096)

Számos megvalósítással kapcsolatos nehézsége miatt a k-Means sokkal gyakrabban használt algoritmus, mint a GMM:

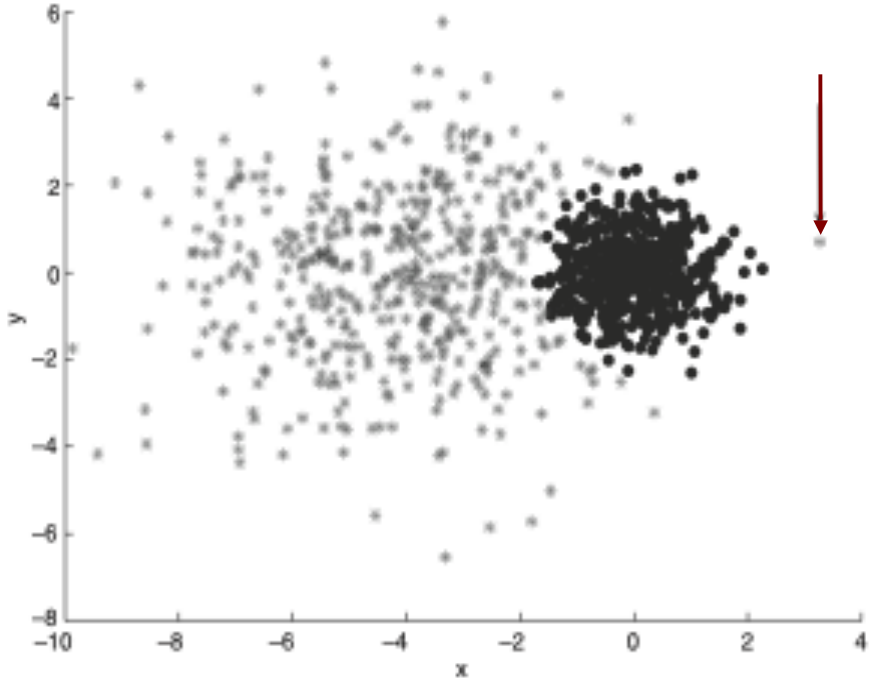
- sokkal számításigényesebb
- talán a K-means-nél is érzékenyebb a kezdőpontokra
- érzékeny a vektortérre
- alulcsordulások miatt sok esetben nem elegendő a fp32 csak bizonyos átalakítások után
- nem egyértelmű mit vesztünk nagydimenziós térben a teljes kovariancia mátrixhoz képest



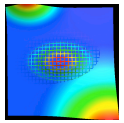
Gaussian Mixture Model

2 eloszlású 2 dimenziós GMM-el klaszterezünk pontokat, fekete és szürke színre festettük a legvalószínűbb eloszlás alapján.

A szélen található pont miért nem a sötétebb osztályhoz tartozik?



Lehetséges ez k-Means algoritmussal?



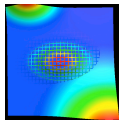
Klaszter analízis

Klaszterezési eljárások jóságának mérésére az ismert adatoktól függően több lehetőség is adódik:

a) amennyiben nem ismerjük az adat eredeti elrendezését, az adott metódushoz igazodva tudunk hibafüggvényt meghatározni:
- K-means : négyzetes hiba a klaszterközépontoktól
- GMM: loglikelihood
- DBSCAN: outlierek 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 2es,
2-es klaszter: 1-es osztály: 4, 2-es osztály: 3, 3-as osztály: 2
tehát a legjellemzőbb osztály az 1es
Purity=(4+4)/(1+4+2+4+3+2) = 0.5
Sajnos a purity növekszik nagyobb klaszterszám mellett....



Klaszter analízis

Kölcsönös információ (mutual information):

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

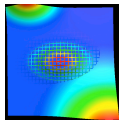
Ahol p_{kj} annak a valószínűsége, hogy egy pont a k-dik klaszterhoz és j-dik osztályba tartozik, p_j a j-dik osztály valószínűsége, p_k pedig a k-dik klaszterba tartozás valószínűsége.

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

Ennek elkerülésére normalizáljuk az entrópia segítségével:

$$H(K)=\sum -p_k \log p_k \qquad H(C)=\sum -p_c \log p_c$$

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



Órai feladat 2:

Vegyük a következő klaszterezést és eredeti osztálybesorolást:
Számítsuk ki a tisztaság mértékét! Van olyan tanult klaszterezési eljárás ami ennél jobban teljesítene?