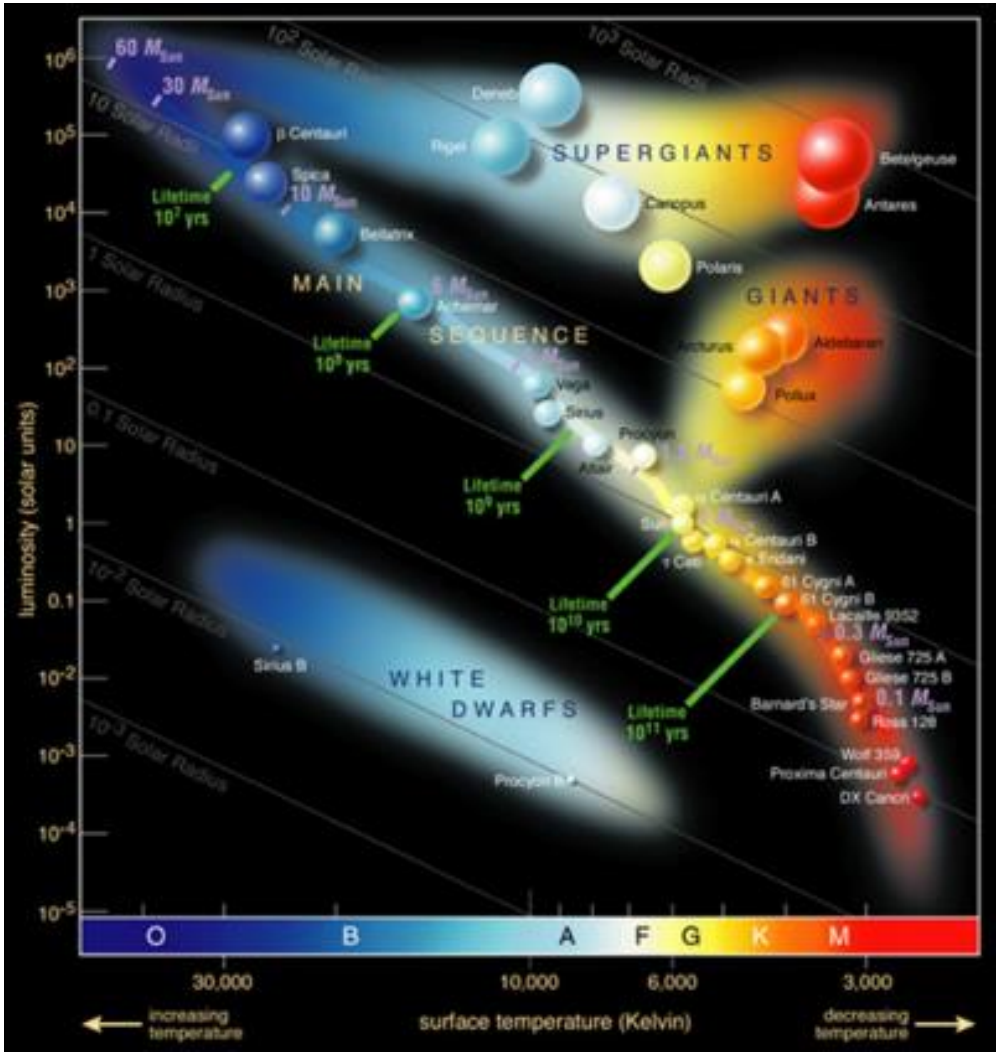
Strojno učenje

Tehnike strojnog učenja bez nadzora

Tomislav Šmuc

Primjer - HRD


SUPERGIANTS
MAIN SEQUENCE
GIANTS
WHITE DWARFS
Betelgeuse
Antares
Canopus
Polaris
Deneb
Rigel
Spica
Bellatrix
Arcturus
Aldebaran
Pollux
Vega
Sirius
Procyon
Altair
Sirius B
Barnard's Star
Wolf 359
Proxima Centauri
Ross 128
Lifetime 10^7 yrs
Lifetime 10^8 yrs
Lifetime 10^9 yrs
Lifetime 10^{10} yrs
Lifetime 10^{11} yrs
60 M_{Sun}
30 M_{Sun}
10 M_{Sun}
0.3 M_{Sun}
0.1 M_{Sun}
10 Solar Radius
1 Solar Radius
0.1 Solar Radius
luminosity (solar units)
O B A F G K M
30,000
10,000
6,000
3,000
increasing temperature
surface temperature (Kelvin)
decreasing temperature

Učenje bez nadzora - Pregled

Clustering

Grupiranje primjera (podataka) u grupe međusobno sličnih primjera

Koji primjeri su slični? (kupci, pacijenti, zvijezde, slike, web-stranice....)

Algoritmi:

- partitivni algoritmi: k-means
- hijerarhijski algoritmi
- SOM - Self-Organization Maps (topologija primjera)

Projekcija podataka, redukcija dimenzija

- pronalaženje latentnih struktura; redukcija dimenzionalnosti

Algoritmi:

- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA),
- Non-negative matrix factorization (NMF)

Clustering

- grupiranje ili segmentacija primjera (podataka) u višedimenzinalnom prostoru
- postoje dijelovi koji su gušće “pokriveni” primjerima
- Centralni pojam – sličnost/udaljenost između primjera
- Ima sličnosti sa učenjem pod nadzorom, no kod klasifikacije – trošak pogreške na neki je način odvojen od samih podataka (klase ili oznake)

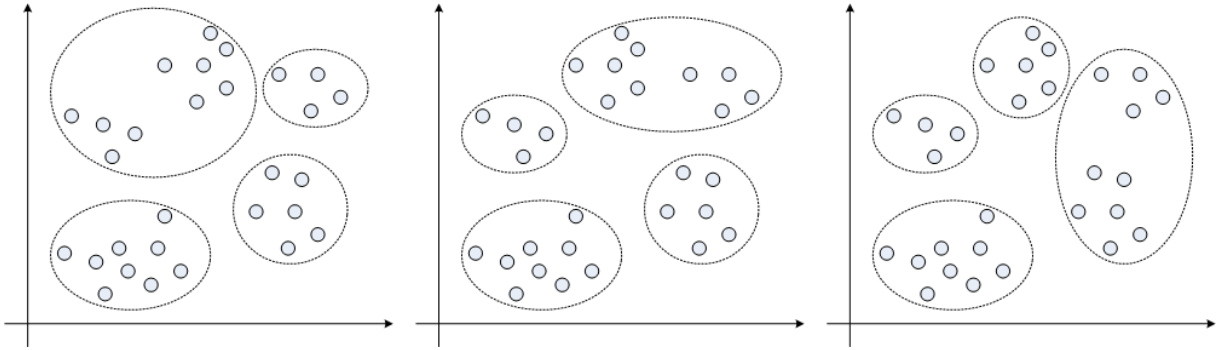
Osnovni problem:

- koliko cluster-a ima ?

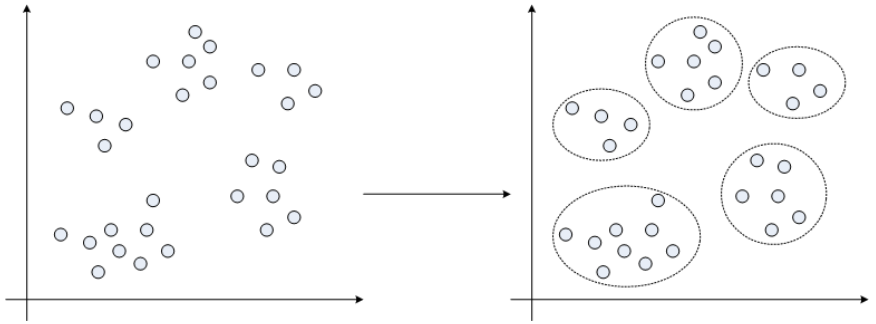
Osnovni problem:

- koliko cluster-a ima ?

- K=4 ?



- K=5 ?



Osnovni cilj:

- odrediti intrinsično grupiranje (neoznačenih) primjera ?

- Kako ćemo odrediti što je dobar rezultat clustering-a?
- Nema apsolutnog kriterija !
  - Nema kriterija koji je odvojen od konačnog cilja clustering-a
- Korisnik – bitan kod određivanja kriterija – poznavanje područja i ciljeva primjene!

Moguće primjene clustering-a

- Redukcija potrebnih podataka – slični podaci ili replike nisu potrebne
- Pronalaženje “prirodnih grupa/cluster-a” i njihovo opisivanje (nova saznanja)
- Korisno grupiranje
- Za detekciju outlier-a, grešaka, šuma (indirektno)

Clustering – osnovni pristupi

- **Partitivni algoritmi**
  - Primjeri se grupiraju u distinktne grupe (jedan primjer – jedna grupa/cluster)
    - algoritam *K-means (*K – srednjih vrijednosti)

- **Hijerarhijski algoritmi**
  - Pronalaze se i definiraju grupe i podgrupe primjera (hijerarhija cluster-a)
    - Skupljajući (agglomerative) i razdvajajući(divisive) algoritmi

- **Ne-ekskluzivni algoritmi**
  - Tzv. fuzzy sets pristupi: jedan primjer može istodobno pripadati dvama ili više cluster-a (stupanj pripadnosti)
    - Algoritam: Fuzzy C-Means

- **Probabilistički algoritmi**
  - Pretpostavlja i određuje (parametarski definiranu) distribuciju iz koje su generirani primjerci
    - EM algoritmi (Expectation maximization) - Gaussian mixture model: u osnovi varijanta *K*-means algoritma

Partitivni clustering - definicije

Odrediti "**encoding" – funkciju** koja određuje pripadnost primjera x_i određenom clusteru *k:*

C(i) =k

Da bi odredili *C(i),* moramo definirati funkciju koju ćemo optimirati – koja najbolje odražava ono što želimo postići:

- odrediti homogene/bliske grupe primjera

1. Definirajmo udaljenost – različitost primjera

$$d(x_i, x_{i'}) = \frac{1}{2}\sum_{j=1}^{p} w_j \cdot (x_{i,j} - x_{i',j})^2$$

Ako želimo da sve varijable podjednako utječu na udaljenost između primjera

$w_j \approx 1/\bar{d}_j$ gdje je $\bar{d}_j = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N} d_j(x_{i,j}, x_{i',j})$

Partitivni clustering - definicije

Definirajmo slijedeće funkcije

- *W(C)*: udaljenost (različitost – dissimilarity) između primjera **iste grupe (clustera**)
- *B(C)*: udaljenost (različitost – dissimilarity) između primjera **različitih grupa (clustera)**

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k} d(x_i, x_{i'})$$

$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')\neq k} d(x_i, x_{i'})$$

3. Mi želimo da *W(C)* bude minimalno:

$$W(C) = T - B(C)$$

$$T = W(C) + B(C)$$

Ukupna međusobna udaljenost između primjera određenog skupa T je konstantna !

K-means (Lloyd S, 1957)

Uz zadani K (broj clustera) :

0. **Inicijalizacija**: Izaberi k srednjih vrijednosti $\boldsymbol{\mu}_j$ (slučajni odabir)
1. **Izračunaj udaljenosti:**
   1. Za *j=1,…,k i i=1,…,n* izračunaj $\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|$
2. **Pridjeli $\boldsymbol{x}_i$ najbližoj srednjoj vrijednosti $\boldsymbol{\mu}_j$:**

   Pripadnost clusteru - C_j (centroid $\boldsymbol{\mu}_j$),
   - indikatorska varijabla γ_j^i

$$\gamma_j^i = \begin{cases} 1 & \text{ako } j = \underset{j'}{\operatorname{argmin}} \left\| \mathbf{x}_i - \boldsymbol{\mu}_{j'} \right\| \\ 0 & \text{inace} \end{cases}$$

3. **Izračunaj nove $\boldsymbol{\mu}_j$:**

$$\boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{\mathbf{x_i} \in C_j} \mathbf{x_i}$$

4. **Ponavljaj 1-3 do konvergencije**

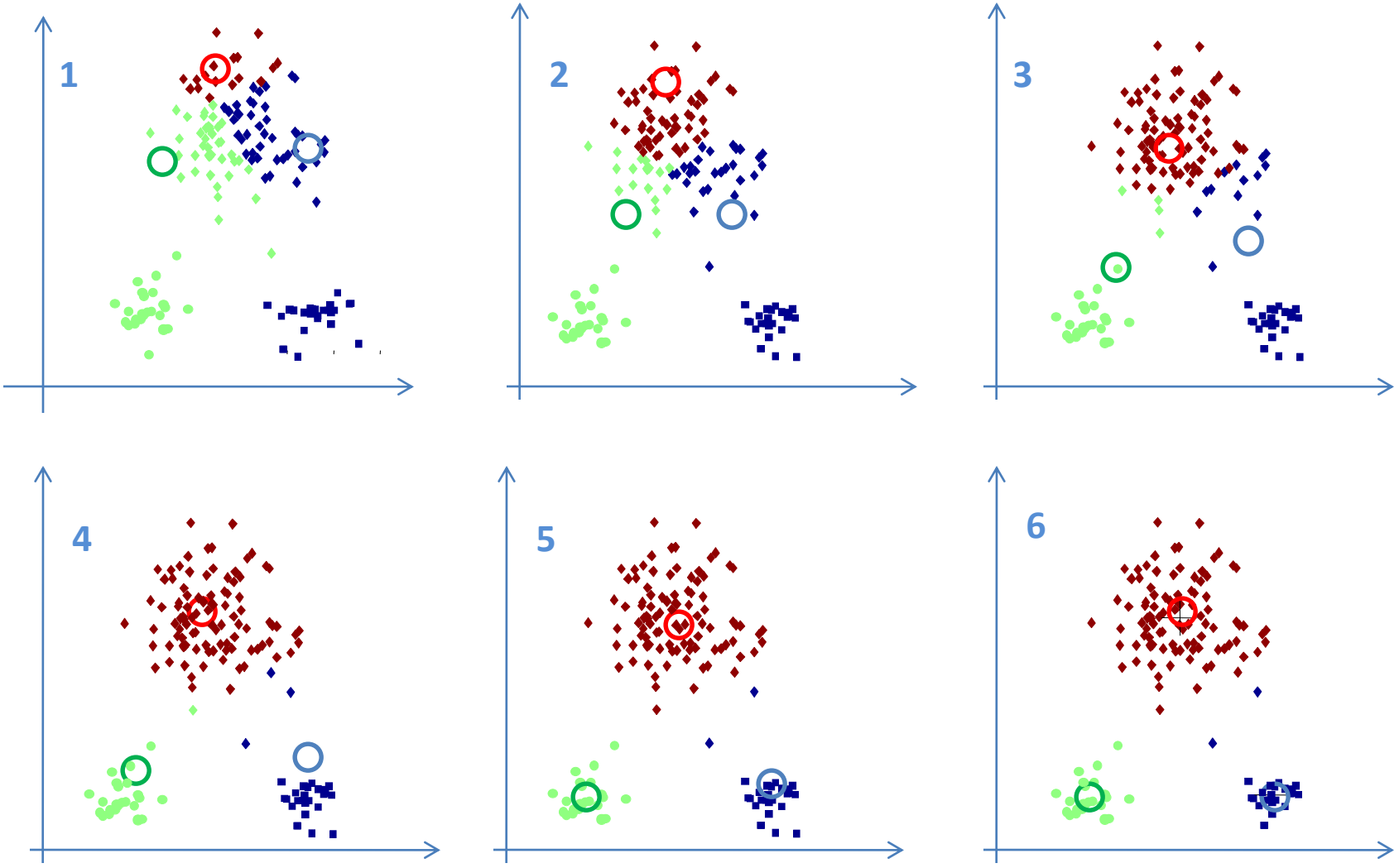
γ_j^i

	j=1	j=2	j=3
x1	0	0	1
x2	1	0	0
x3	0	1	0
x4	0	1	0
x5	0	0	1
x6	1	0	0

- γ_j^i - članovi matrice $\boldsymbol{\gamma}$ (n*K) – jedna 1 po retku (primjer $\boldsymbol{x}_i$)

K=3

K-means: ilustracija



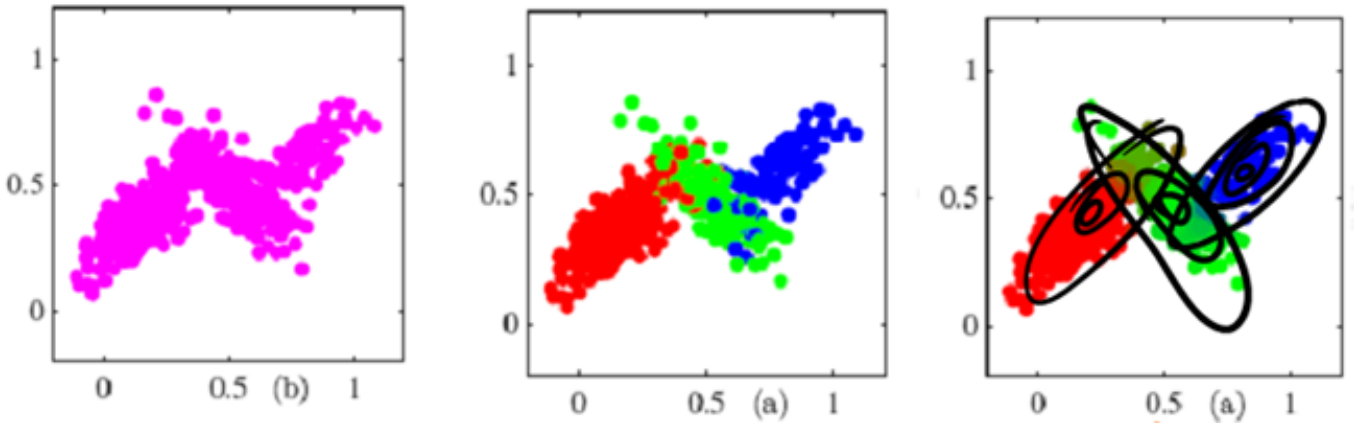
K-means - detalji

- Inicijalni centroidi - uglavnom slučajno određeni
  - Centroid se tipično određuje kao srednja vrijednost točaka u cluster-u
- Udaljenost primjera
  - tipično Euklidska, ali i druge mjere: korelacija, “kosinusna” sličnost
- Konvergencija – uvijek konvergira, za najčešće korištene mjere udaljenosti
  - najveće promjene su u prvim iteracijama.
  - stopping kriterij: obično kada je broj promjena < od nekog zadanog broja
- Složenost: O(n * K * I * d)
  - n = broj točaka, K = broj cluster-a,
    I = broj iteracija, d = broj atributa/varijabli

Problemi i ograničenja K-means algoritma

- Odabir inicijalnih centroida (slučajan)
- Koliki je (optimalni) K
- Utjecaj “outlier-a”
- „Hard assignment” – točka pripada samo jednom klasteru
  - Funkcionira dobro kada su klasteri otprilike slične veličine
  - Probabilističke varijante (modeliraju svaki klaster kao Gausovu distribuciju) rješavaju taj problem
- Geometrija podataka - oblik “stvarnih” cluster-a
  - k-means radi dobro u slučajevima kada se radi o ovalnim/okruglim grupama, ne ponaša se dobro kada se radi o ne-konveksnim oblicima
  - Kernel k-means varijanta rješava takve slučajeve

(0 ili 1) pripadnost (Hard clustering)



soft clustering (0 ➔1) pripadnost

Mješavina klastera kombinacijom k-Gaussovih distribucija

$$p(\mathbf{x}_i \mid \theta) = \sum_{j=1}^{K} p(j) N(x_i \mid \mu_j, \sigma_j^2)$$

= vjerojatnost da je $\mathbf{x}_i$ „posljedica" (težinska kombinacija) K Gaussovih distribucija.

Nepoznanice – parametri modela koje treba odrediti:

- $p(j)$ – vjerojatnosti klastera (j) (vrijedi) $\sum_{j=1}^{K} p(j) = 1$,
- $\theta_j = (\mu_j, \sigma_j)$

EM algoritam - Expectation Maximization

- Kad bi znali $\gamma_c(x_i)$ – vjerojatnost pripadanja x_i klasteru *c,* bilo bi jednostavno odrediti μ_c, σ_c klastera. No, da bi odredili $\gamma_c(x_i)$ trebaju nam μ_c, σ_c !

EM algoritam

1. Inicijaliziraj početne parametre $\gamma_{ic}^0, \mu_c^0, \Sigma_c^0$

Ponavljaj sve dok ne vrijedi $\gamma_c{}^{t+1}(\boldsymbol{x}_i) - \gamma_c{}^t(\boldsymbol{x}_i) < \varepsilon \ \forall \ \boldsymbol{x}_i$

2. **E korak** (**E**xpectation) – u iteraciji *t*, izračunaj očekivanja vrijednosti indikatora $\gamma_c{}^t(\boldsymbol{x}_i)$ (da primjer $\mathbf{x}_i$ pripada klasi *c*) i normaliziraj:

$$\tilde{\gamma}_c^t(\mathbf{x}_i) = \frac{p_c^t \cdot N(\mathbf{x}_i \mid \mu_c^t, \Sigma_c^t)}{\sum_j^K p_j^t \cdot N(\mathbf{x}_i \mid \mu_j^t, \Sigma_j^t)}$$

$$\gamma_c^t(\mathbf{x}_i) = \frac{\tilde{\gamma}_c^t(\mathbf{x}_i)}{\sum_j^K \tilde{\gamma}_j^t(\mathbf{x}_i)}$$

3. **M korak** (**M**aximization) – Osvježi parametre - $p_c{}^{t+1}, \mu_c{}^{t+1}, \Sigma_c{}^{t+1}$

$$\boldsymbol{\mu}_c^{t+1} = \frac{\gamma_c^t(\mathbf{x}_i) \cdot \mathbf{x}_i}{\sum_j^K \gamma_j^t(\mathbf{x}_i)}$$

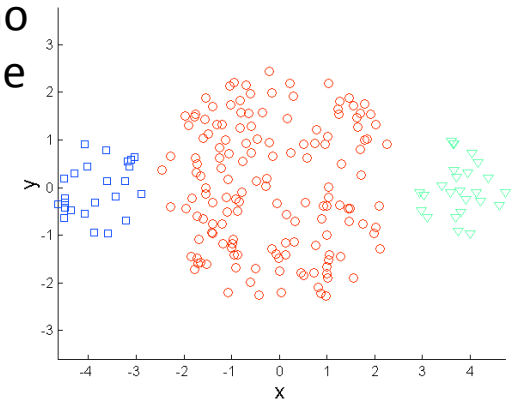
$$\Sigma_c^{t+1} = \sum_{i=1}^n \gamma_c^t(\mathbf{x}_i)(\mathbf{x}_i - \boldsymbol{\mu}_c^t)^T(\mathbf{x}_i - \boldsymbol{\mu}_c^t)$$

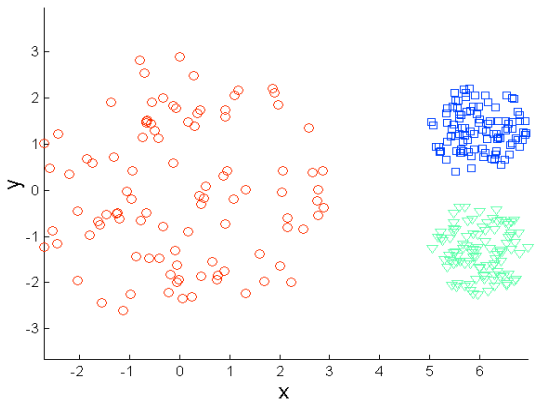
$$p_c^{t+1} = \frac{1}{n}\sum_{i=1}^n \gamma_c^t(\mathbf{x}_i)$$

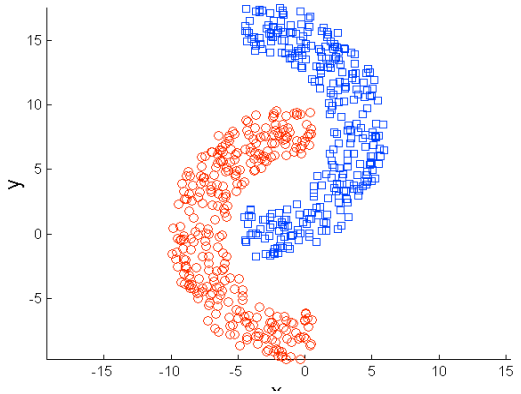
Problemi i ograničenja K-means algoritma: različite veličine

različite veličine | različite gustoće | nekonveksan oblik

Originalno grupiranje

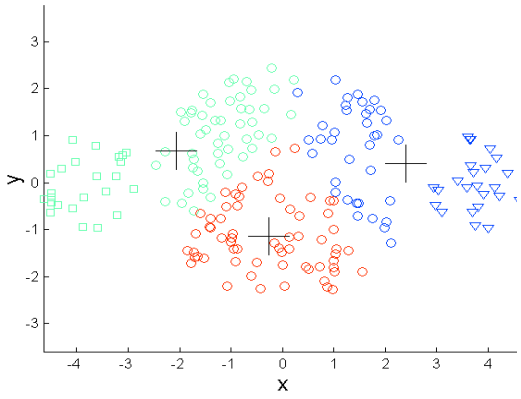


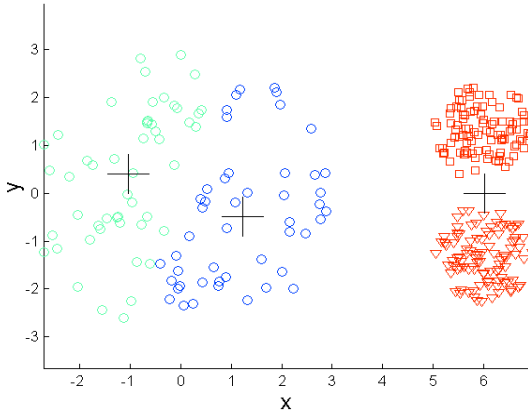


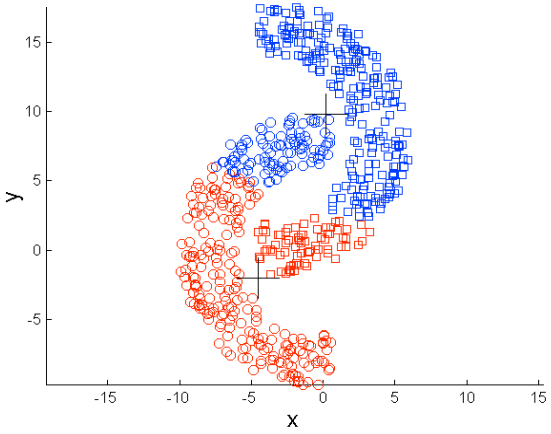


K-means

(K=3)



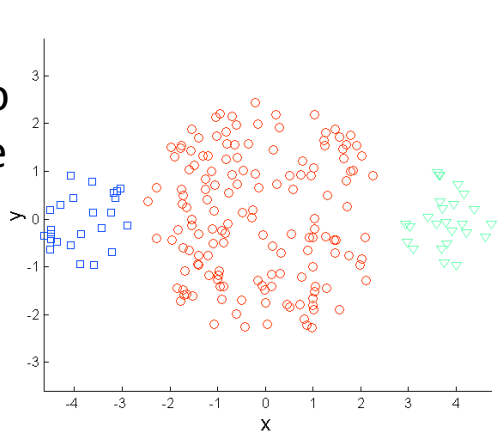


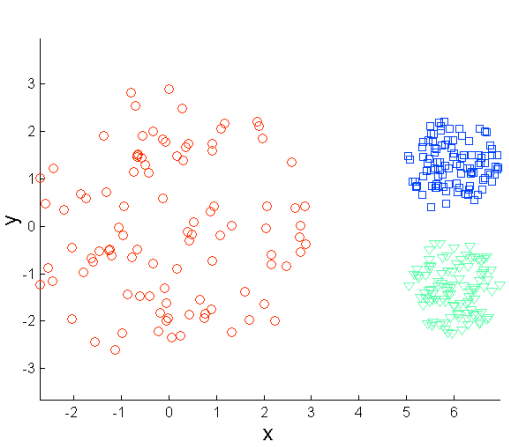


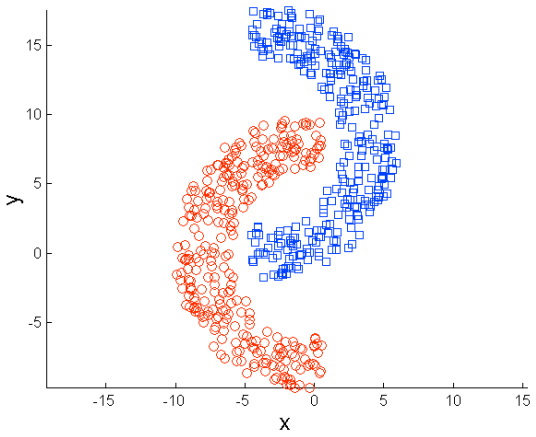
Tipično rješenje: veći K

- Dijelovi (pravih) cluster-a: treba ih još povezati !?

Originalno grupiranje

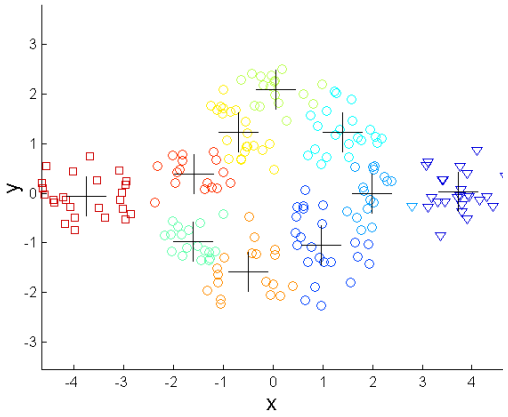


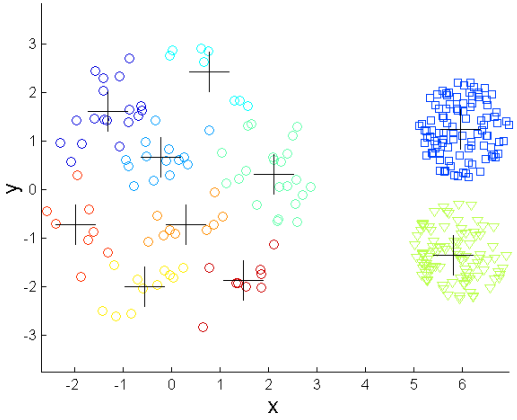


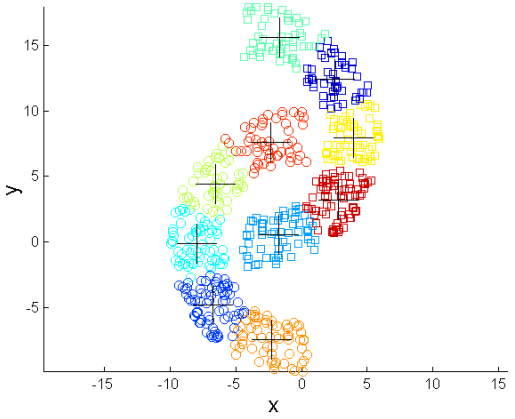


K-means

(K=10)







Clustering – Kernel K-means

Osnovna ideja:

- umjesto Euklidske udaljenosti u originalnom prostoru varijabli koristimo kernel (udaljenosti u mapiranom višedimenzionalnom prostoru)

$$d(x_n, \mu_k) = \|\varphi(x_n) - \varphi(\mu_k)\|$$

$$\|\varphi(x_n) - \varphi(\mu_k)\|^2 = \|\varphi(x_n)\|^2 + \|\varphi(\mu_k)\|^2 - 2\varphi(x_n)^T\varphi(\mu_k)$$

$$= k(x_n, x_n) + k(\mu_k, \mu_k) - 2\, k(x_n, \mu_k)$$

gdje je $k(.,.)$ kernel funkcija, a φ pripadna (implicitna) mapa

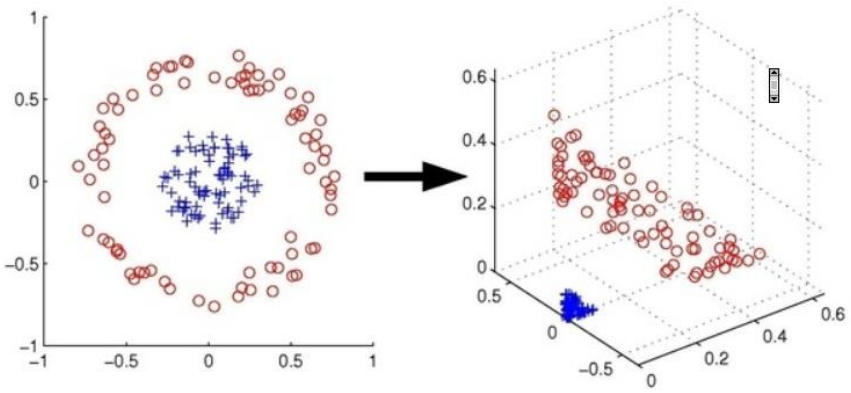
Podsjetnik:

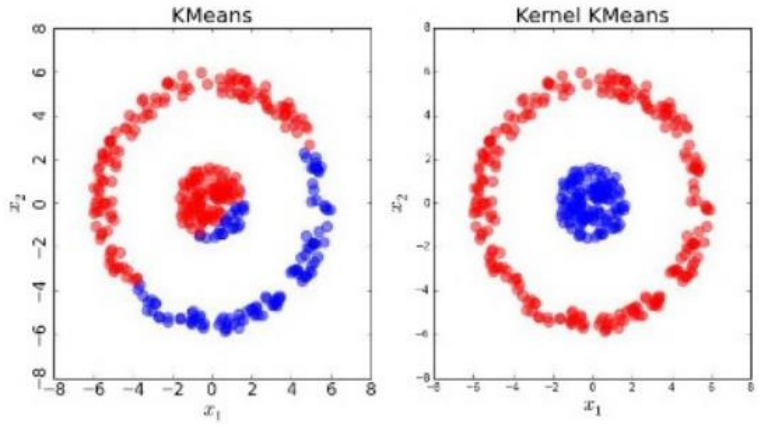
φ – je višedimenzionalni (beskonačno dim.!?) prostor varijabli, za koji je $k(.,.)$ funkcija udaljenosti itmeđu točaka

Ne moramo imati izračunate/spremljene $\varphi(x_n)$ i $\varphi(\mu_k)$ => algoritam traži samo kernel izračune !

Clustering – Kernel K-means

x $\varphi(x)$





Evaluacija clustering rezultata

- Najčešća mjera suma kvadratne pogreške (SSE):
  - Za svaki primjer, greška je kvadrat udaljenosti do centroide c_j cluster-a kojem primjer x_i pripada

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_j} d^2(c_j, x_i)$$

  - Uz dana 2 cluster-a – odabrat ćemo onaj s manjom greškom!
  - Jedan od načina kako smanjiti SSE - povećati K broj cluster-a
    - bolje mjere mogu razlikovati dobar rezultat sa manjim K, od relativno lošijeg rezultata sa većim K

Mjere "dobrote" clustering-a (en. cluster validity measures):

- Davis Bouldin Index, Dunn's Validity index, C-index...

Davis Bouldin Index (DBI)

Funkcija (sume) "raspršenja" primjera unutar (intra) cluster-a i separacije između clustera

Ako su $C=\{C_1,\ldots..,C_k\}$ clusteri na skupu N primjera i definiramo:

$$R_{\substack{ij \\ i\neq j}} = \frac{\mathrm{var}(C_i)+\mathrm{var}(C_j)}{\| c_i - c_j \|} \qquad \text{i} \qquad R_i = \max_{j=1,..k, i\neq j} R_{ij}$$

c_i centroid C_i

$$DBI = \frac{1}{k}.\sum_{i=1}^{k} R_i$$

Minimalni DBI => optimalan K;
DBI – pogodan za usporedbu clustering metoda

Grupiranje/clustering prema gustoći

- Svaki klaster karakterizira značajno veća gustoća unutar negoli van klastera
- Algoritmi su bazirani na pojmu susjedstva i povezanosti primjera na bazi lokalne gustoće primjera

DBSCAN algoritam - koncepti

Tipologija točaka

1. X je **centralna točka (core point)** ako **n $_{\epsilon}$> minpts** (t.j. ima u svom **ε –okolišu > minpts** točaka)
2. X je **granična točka (border point)** ako **n $_{\epsilon}$< minpts, ali X pripada** ε –okolišu barem jedne centralne točke
3. X je **usamljena točka(iznimka)/**(**noise/outlier point)** ako nije 1. ili 2.

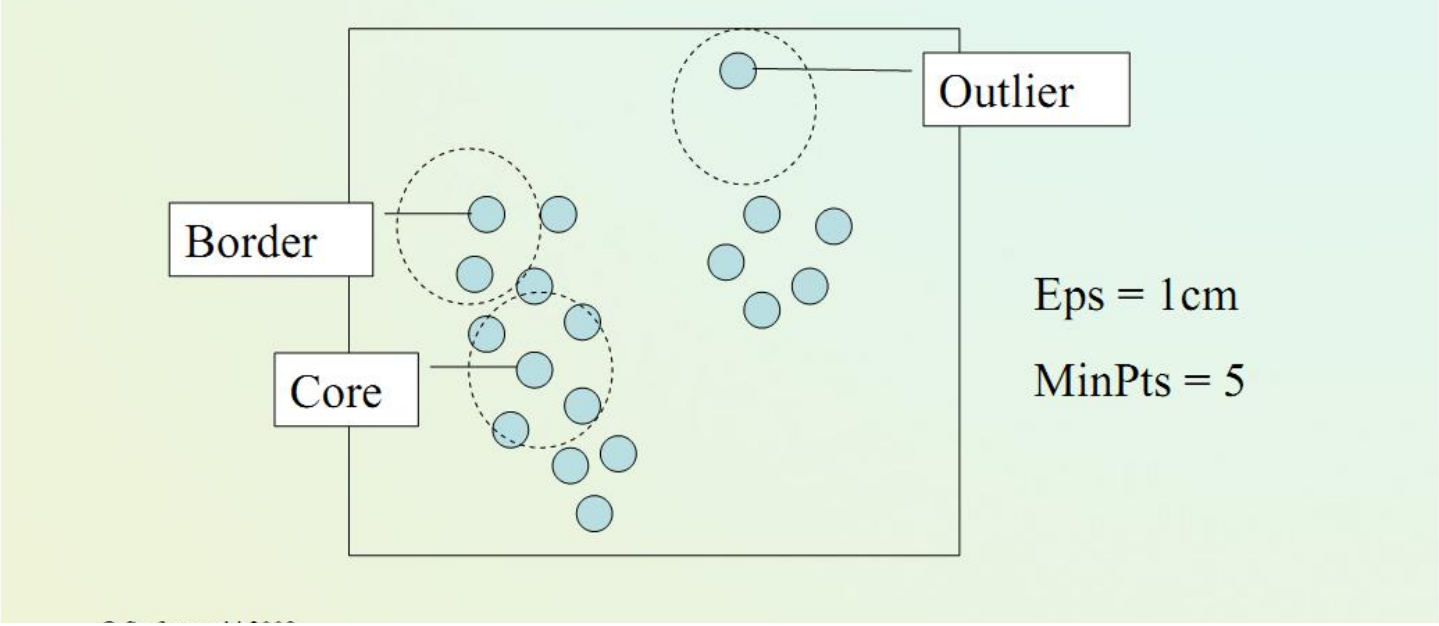
Dostupnost bazirana na gustoći (density reachable)

- **Direktno dostupne točke (Direct density reachable points - DDR):**
  => (**x** je DDR od **y** ako je **x ϵ ε-neighborhood**)
- **Indirektno dostupne točke (Density reachable points - DR):**
  => **x** je DR od **y** ako postoji lanac točaka:
  $x_0;x_1; \ldots ;x_l$, takav da je $x=x_0$ and $y=x_l$, a x_i je DDR od x_{i-1} za sve i= 1..l
- **Povezanost primjera na bazi gustoće (Density connectedness - DC):**
- Bilo koje dvije točke **x** i **y** su DC ako postoji centralna točka **z**, takva da su i **x** i **y** DR od strane **z**.

Grupa/klaster na bazi gustoće / Density based cluster

- ….definiran je kao: maksimalni skup gustoćom povezanih točaka

DBSCAN – osnovni koncepti



DBSCAN Algorithm

ALGORITHM 15.1. Density-based Clustering Algorithm

DBSCAN $(\mathbf{D}, \epsilon, minpts)$:

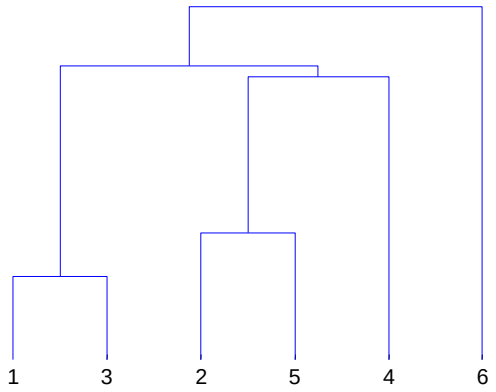
```
1  Core ← ∅
2  foreach x_i ∈ D do // Find the core points
3      Compute N_ε(x_i)
4      id(x_i) ← ∅ // cluster id for x_i
5      if N_ε(x_i) ≥ minpts then Core ← Core ∪ {x_i}
6  k ← 0 // cluster id
7  foreach x_i ∈ Core, such that id(x_i) = ∅ do
8      k ← k + 1
9      id(x_i) ← k // assign x_i to cluster id k
10     DENSITYCONNECTED (x_i, k)
11 C ← {C_i}_{i=1}^k, where C_i ← {x ∈ D | id(x) = i}
12 Noise ← {x ∈ D | id(x) = ∅}
13 Border ← D \ {Core ∪ Noise}
14 return C, Core, Border, Noise
```

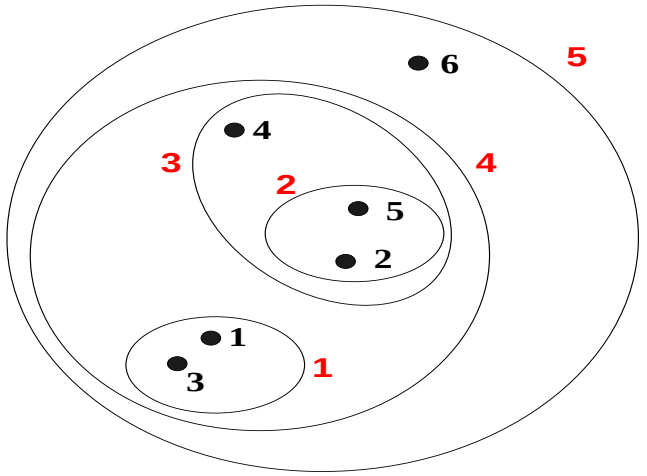
DENSITYCONNECTED $(\mathbf{x}, k)$:

```
15 foreach y ∈ N_ε(x) do
16     id(y) ← k // assign y to cluster id k
17     if y ∈ Core then DENSITYCONNECTED (y, k)
```

Hijerarhijski clustering

- Hijerarhija grupa/cluster-a, organiziranih poput obrnutog stabla ~ dendrograma
- Dendrogram
  - dijagram kojim se prikazuje redoslijed spajanja primjera/clustera





HC - Zbog čega može biti interesantan?

- Moguće je da udaljenosti između primjera, a time i HC daje nekakvu smislenu hijerarhiju – taksonomiju koncepata
  - Npr. u biologiji – sekvence prema sličnosti – filogenetska stabla organizama)

- Broj clustera (udruživanja) može biti proizvoljan

Hijerarhijski clustering

- Dva osnovna tipa
  - Aglomerativnog tipa (spajanje : “bottom up”)
    - Početak – točke su osnovni clusteri
    - U svakom koraku – spajamo najsličniji par cluster-a
    - Kraj - kada dostignemo zadani broj K (ili minimalno jedan veliki cluster)
  - Razdvajajući (en. divisive) (dijeljenje: “top-down”)
    - Početak – jedan veliki cluster = svi primjeri
    - U svakom koraku, dijelimo cluster sve dok ne dodjemo do nivoa zadanog broja K clustera (ili je svaki cluster jedan primjer)

- Pri spajanju ili dijeljenju – koristimo matrice sličnosti ili udaljenosti između primjera

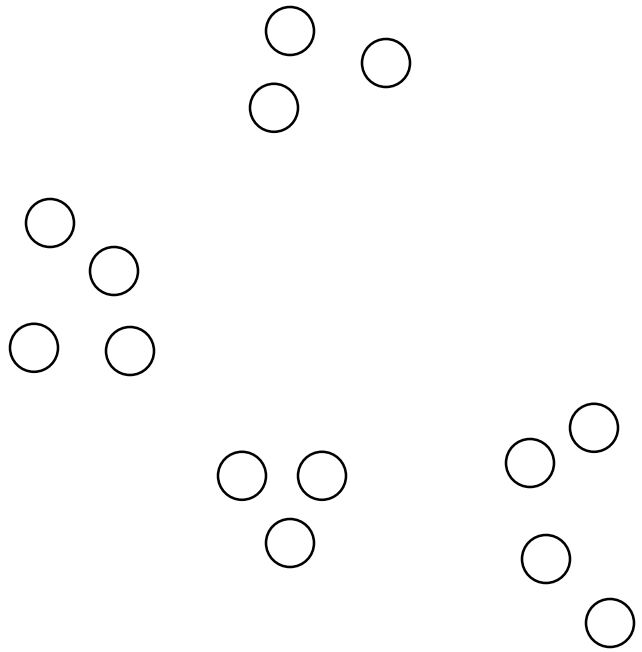
Aglomerativni HC algoritam

- **Izračunaj matricu udaljenosti/sličnosti**
- **Svaki primjer je cluster**
- **ponavljaj**
  - Spoji dva najbliža cluster-a
  - Ponovno izračunaj udaljenosti/sličnosti u matrici
- **dok** ne preostane samo K cluster-a (jedan cluster)

Osnovna operacija – računanje udaljenosti/sličnosti između dva cluster-a: Različiti pristupi

Aglomerativni HC algoritam

- Početak

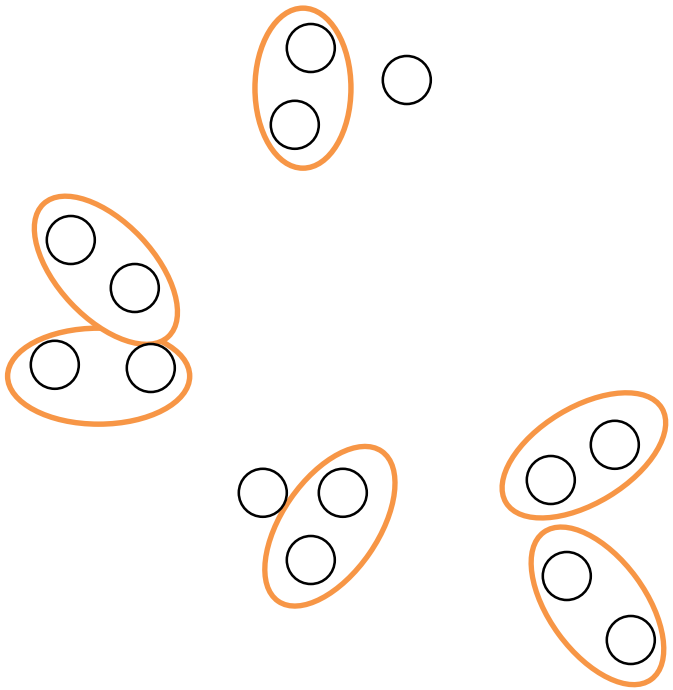


	p1	p2	p3	-...	pn
p1	0	1.1	2.1	...	
P2	1.1	0	3.2	...	
P3	2.1	3.2	0	...	
...	...	...			

Matrica udaljenosti

Aglomerativni HC algoritam

- i. korak

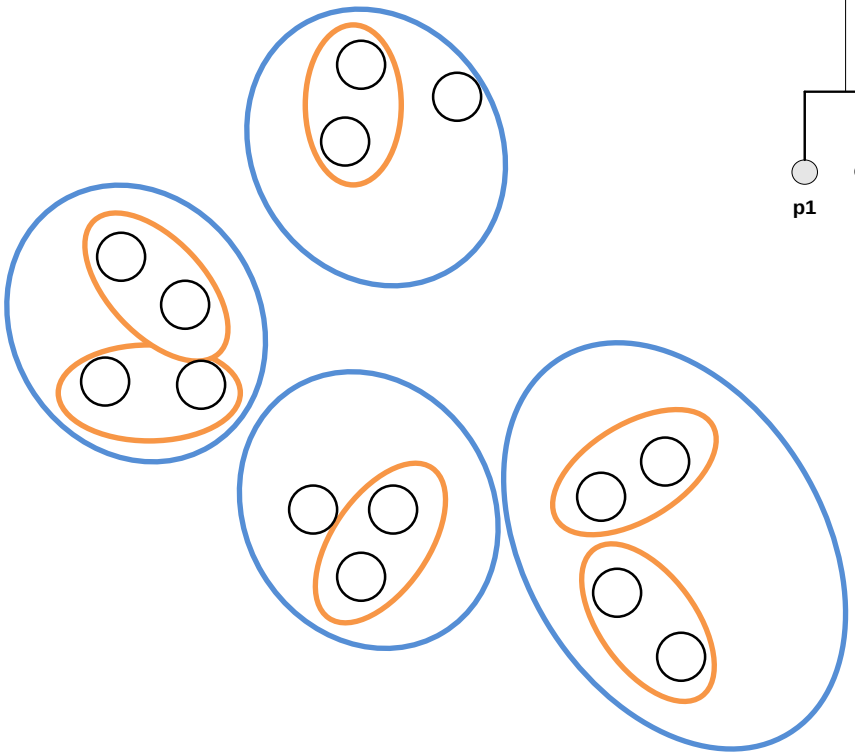


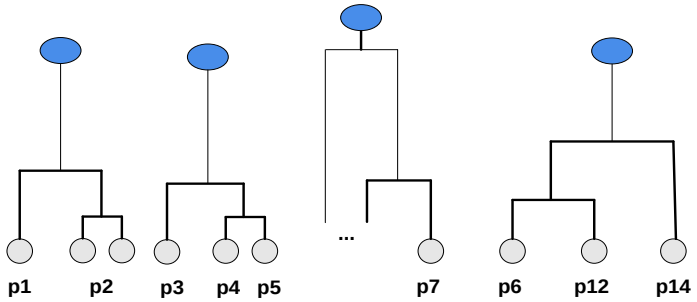
	c1	c2		-...	cj
C1	0	1.1		...	
c3	2.1	3.2		...	
...	...	...			

Matrica udaljenosti

Aglomerativni HC algoritam

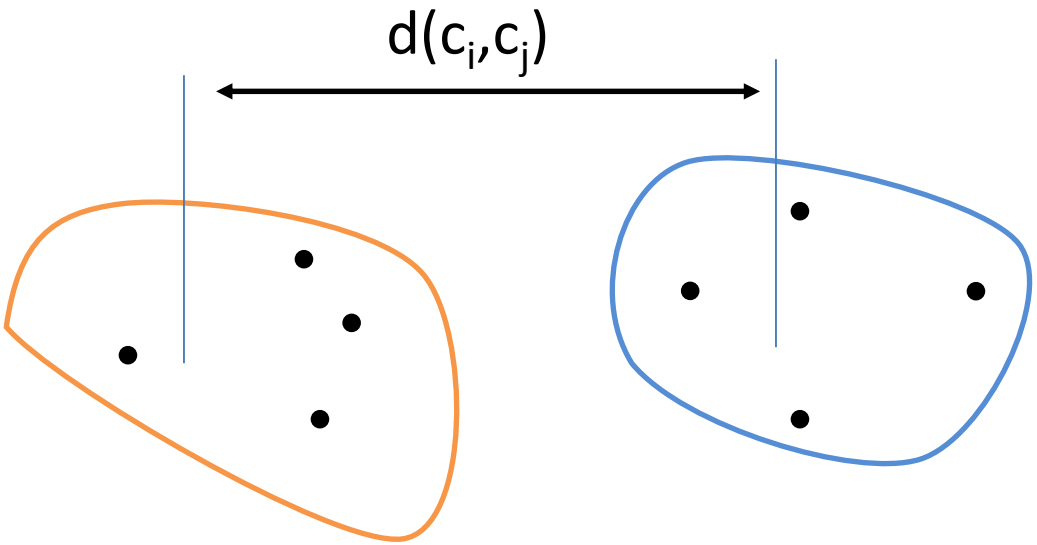
- Zadnji korak (K=4)





AHC - Osnovno pitanje

- Kako računamo matricu udaljenosti/sličnosti između cluster-a?



- MIN $d(x_i,x_j)$
- MAX $d(x_i,x_j)$
- Udaljenost između centroida c_i i c_j
- Srednja udaljenost primjera c_i naspram primjera c_j
- Druge složenije metode

AHC - Osnovno pitanje

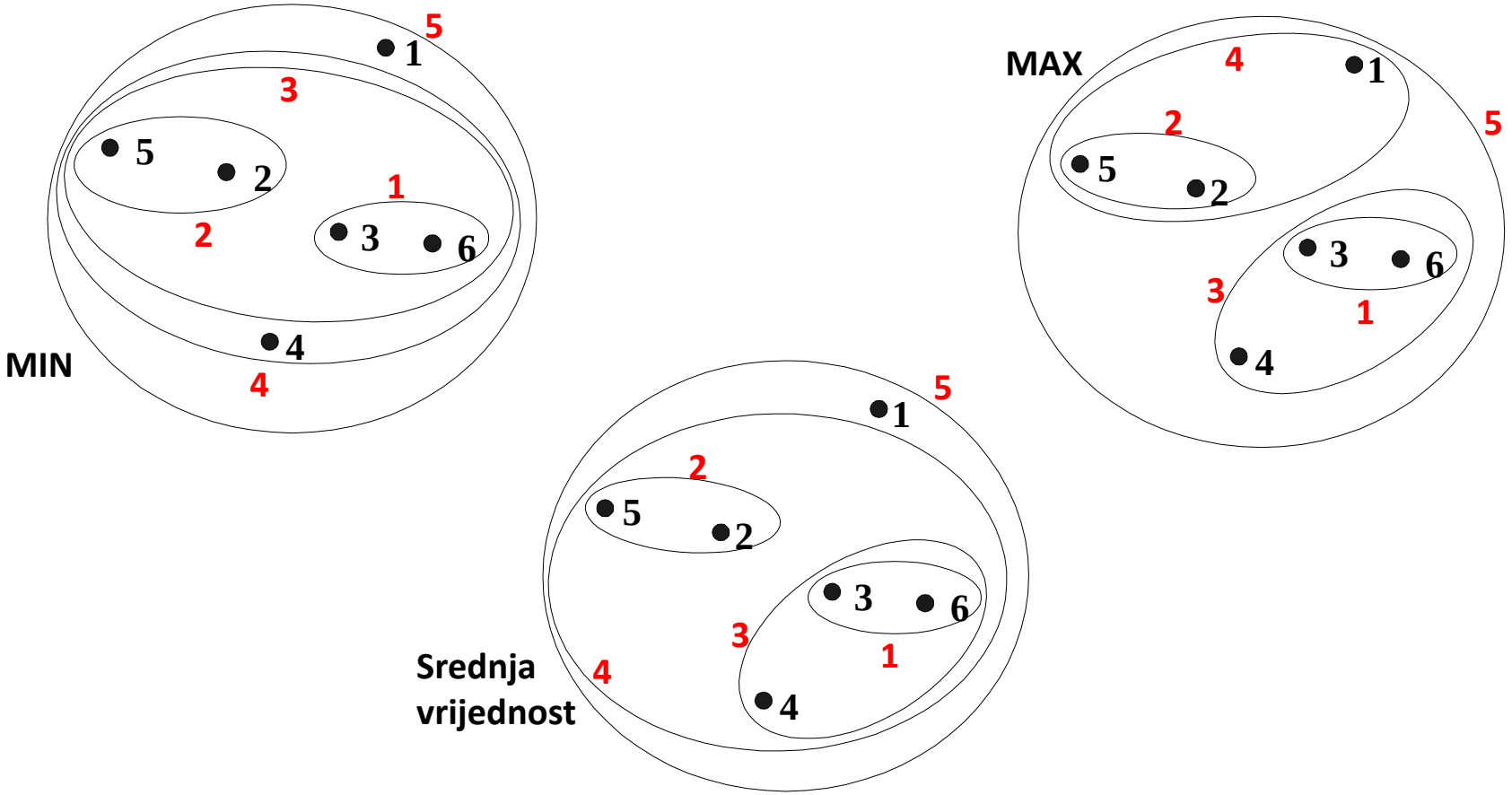
- Zavisno o odabranoj metodi dobit ćemo različiti rezultat !

- **MIN $d(x_i,x_j)$**
  - dobro: dobro aproksimira eliptične oblike cluster-a
  - loše: osjetljiva na šum i "outlier" točke

- **MAX $d(x_i,x_j)$**
  - dobro: manje osjetljiva na šum i outlier-e
  - loše: - "sklona" mrvljenju većih cluster-a
    - "sklona" stvaranju globularnih cluster-a

- **Srednja udaljenost primjera c_i naspram primjera c_j**
  - Kompromis između MIN i MAX
  - Dobro: manje osjetljiva na šum i outlier-e
  - Loše: - "sklona" stvaranju globularnih cluster-a

AHC: različite metode računanja udaljenosti/sličnosti => različiti konačni rezultati



AHC: složenost

$O(N^2)$ – prostorna

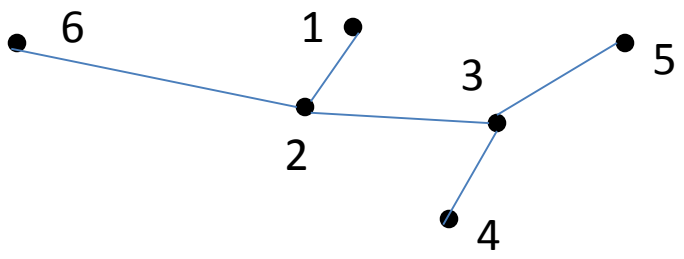
- (N= br primjera – N^2 matrica udaljenosti/sličnosti)

$O(N^3)$ – vremenska

- N koraka, N^2 proračuna matrice, te pronalaženje najsličnijih cluster-a
- Neki algoritmi postižu $O(N^2 \log(N))$

DHC : MST (Minimum Spanning Tree)algoritam

1. Inkrementalno gradi MST
   - Početak: Stablo je jedan (prvi - slučajni) primjer x_p
   - **Ponavljaj**
     - dodaj novi primjer x_j u stablo tako da nađeš minimalni $d(x_p, x_j)$ između svih parova x_p – unutar stabla i x_j - van stabla
     - Dodaj x_j u stablo i stavi vezu između x_p i x_j
   - **dok** nisu sve točke u stablu



Razdvajajući hierarhijski clustering
DHC : MST (Minimum Spanning Tree)algoritam

2. Koristi MST da bi napravio cluster-e:

   MST je cluster

   - **Ponavljaj**

     - napravi novi cluster tako da nađeš najveću udaljenost (najmanja sličnost) koja još nije prekinuta u nekom od postojećih dijelova MST (cluster-a)

   - **dok** nisu sve svi dijelovi stabla (clusteri) svedeni na primjere

SOM – (Teuvo Kohonen, 1981)

- Cilj: topologija primjera => mapiranje primjera u niže-dimenzionalni prostor, uz uvjet da udaljenosti između primjera budu što je više moguće sačuvane

- SOM (Kohonenove mape) – projekcija višedimenzionalnog prostora
  - na 1D ili 2D grid/mapu čvorova (neuroni!)

- Veza prema stvarnoj biologiji:
  - slična percepcija vodi na ekscitiranje u istim područjima mozga

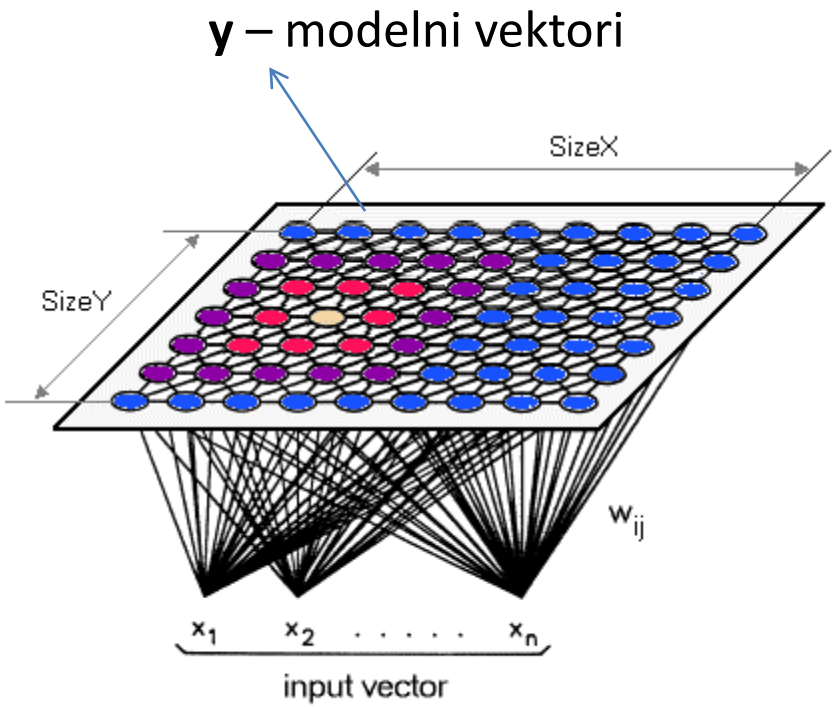
SOM – samo-organizirajuća mapa

- SOM algoritam – uči mapiranje s ulaznih primjera na 2D/1D mrežu neurona

- **y** - modelni vektori se nalaze na mapi (1D ili 2D)

- Sačuvanje originalne topologije primjera (~ sačuvanje udaljenosti između primjera)

- clustering alat kod kojeg je vizualizacija bitan aspekt



- SOM – ima i generalizacijska svojstva:
  Novi primjer – “asimilira” se u određenom čvoru mreže !

SOM Algoritam

•Odabrati topologiju mreže (dimenzija mxm!; oblik čvorova...)

- incijaliziraj početnu **veličinu susjedstva *D(0)***
- zadaj $0 < \eta(t) \leq \eta(t-1) \leq 1$ faktor učenja
  (uglavnom promjenjiv – smanjuje se s *t*)

- **Inicijaliziraj modelne vektore y_j**
- **dok nije zadovoljen kriterij zaustavljanja**

a. Odaberi ulazni primjer x_i

b. Odredi euklidske udaljenosti između x_i i čvora y_j na mreži

$$\sum_{k=1}^{n} (x_{i,k} - y_{j,k})^2$$

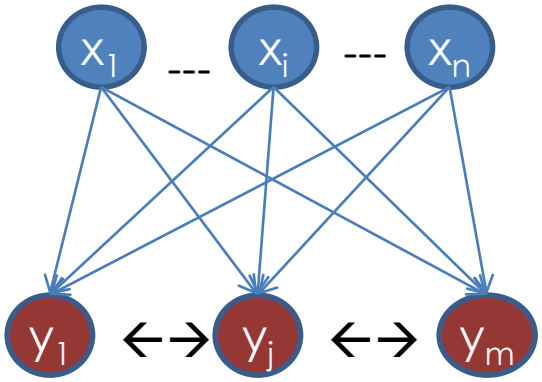
c. Odredi čvor *j** prema kojem udaljenost ima minimalnu vrijednost u odnosu na x_i

d. Promijeni sve modelne vektore na mreži koji su unutar susjedstva ***D(t)*** od y_{j*} koristeći:

$$y_j(t+1) = y_j(t) + \eta(t)(x_i - y_j(t))$$

- **povećaj *t***

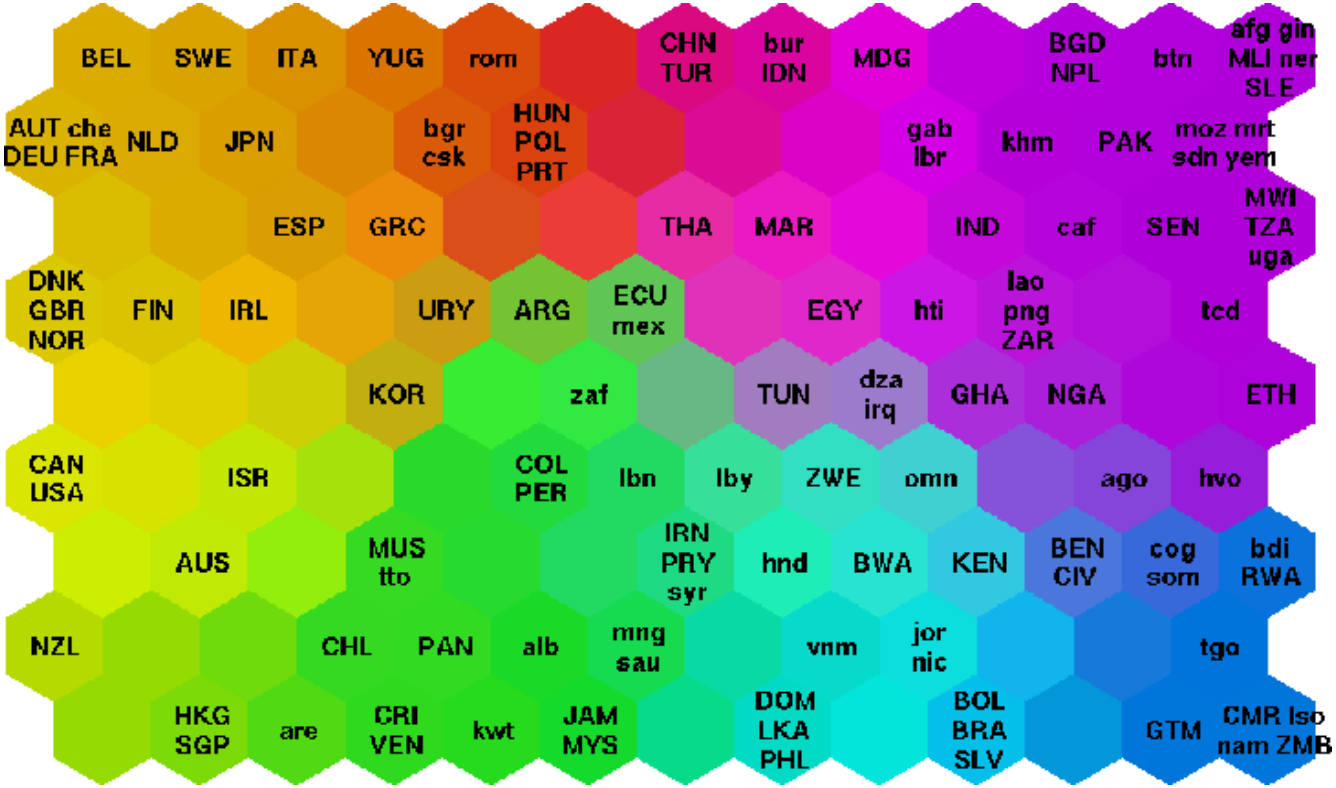
Arhitektura



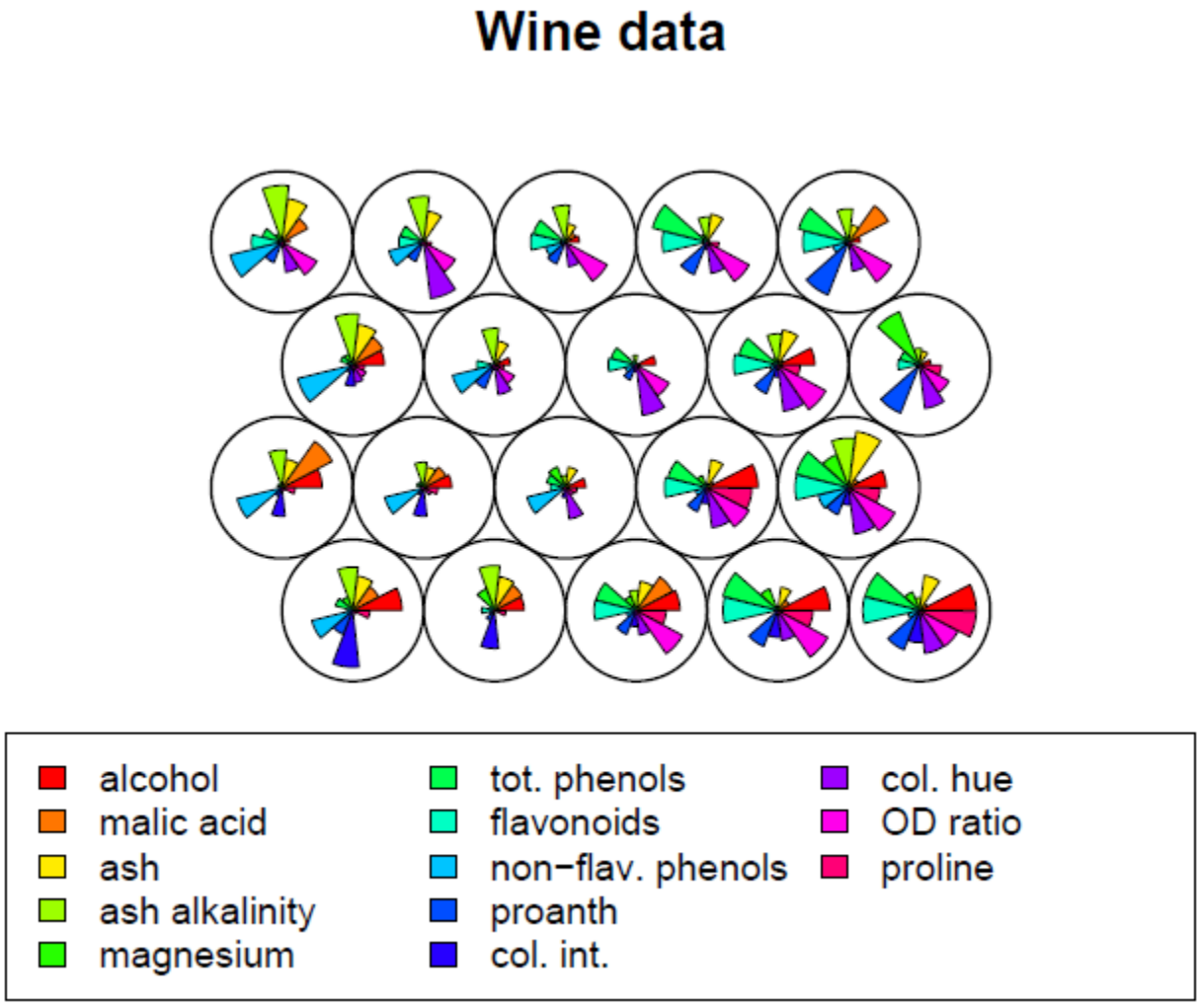
SOM – značaj i primjene

- NN model rada mozga
- Vizualizacija velikih skupova podataka
- Vid reprezentacije znanja

SOM:
World poverty map



SOM (Self-organizing-maps)

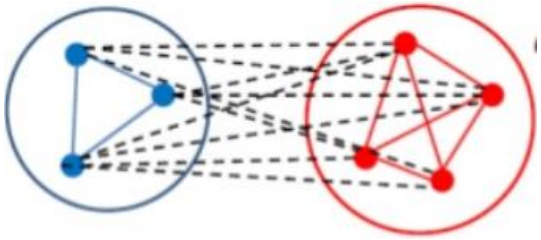


R-package: *kohonen*

Druge metode, Drugi tipovi podataka

Spektralno grupiranje (spectral clustering)

- Clustering ~ particioniranje grafa
- graf => na bazi matrice udaljenosti izmedju primjera (k-nn graf)
- Optimizacija particioniranja grafa => Normalized cut problem



Grupiranje nad mrežnim podacima (grafovi)

$\Rightarrow$ Community detection algoritmi

Literatura

Data Mining and Analysis (Zaki & Meira)
- Ch 13 - ch 17

The Elements of Statistical Learning
- Ch 14

Introduction to Machine Learning (E. Alpaydin)
- Ch 7

- Lloyd, S. P. (1957). Least squares quantization in PCM. Technical Report RR-5497, Bell Lab, September 1957.