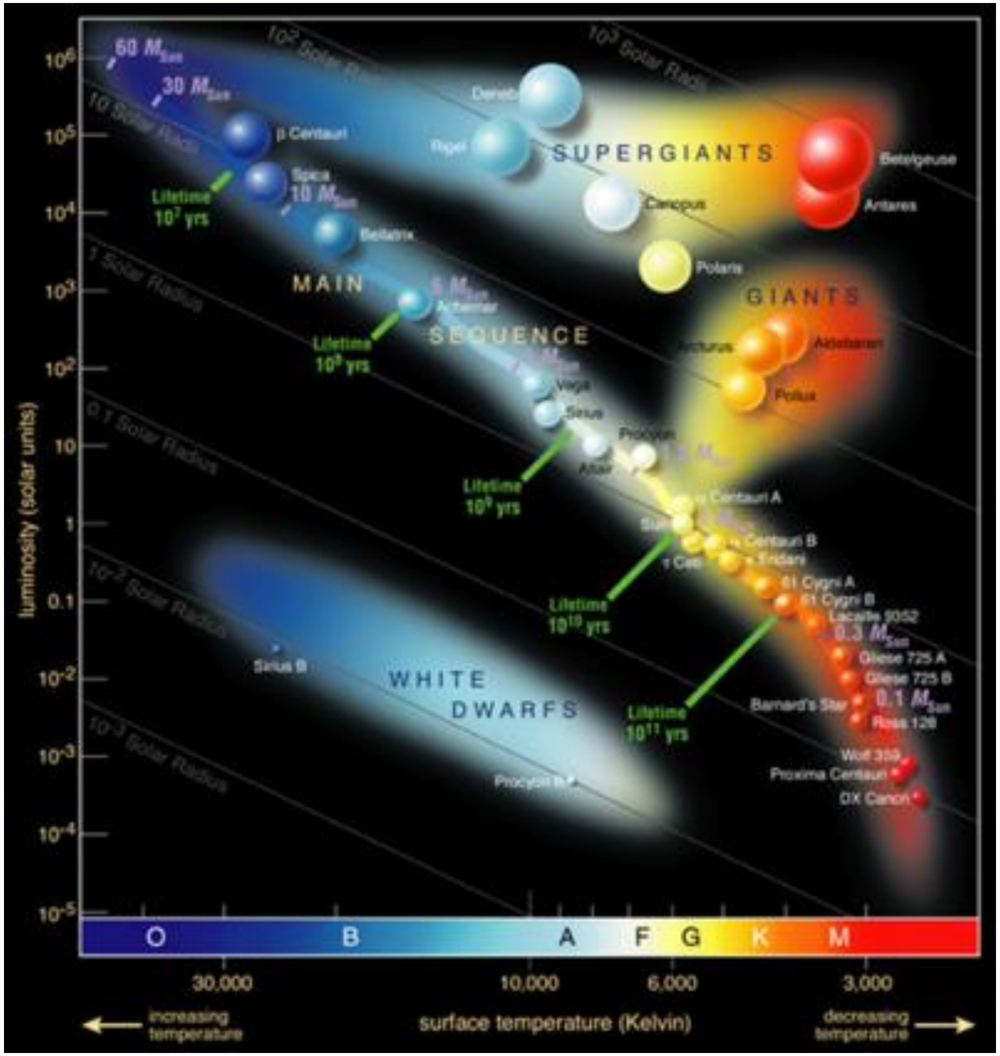
Strojno učenje

Tehnike strojnog učenja bez nadzora

Tomislav Šmuc

Primjer - HRD


60 M_{Sun}
30 M_{Sun}
10 M_{Sun}
β Centauri
Spica
Bellatrix
Achernar
Deneb
Rigel
SUPERGIANTS
Betelgeuse
Antares
Canopus
Polaris
GIANTS
Arcturus
Aldebaran
Pollux
MAIN
SEQUENCE
Vega
Sirius
Procyon
Altair
Sun
Lifetime 10^7 yrs
Lifetime 10^8 yrs
Lifetime 10^9 yrs
Lifetime 10^{10} yrs
Lifetime 10^{11} yrs
61 Cygni A
61 Cygni B
Lacaille 9352
0.3 M_{Sun}
Gliese 725 A
Gliese 725 B
Barnard's Star
0.1 M_{Sun}
Ross 128
Wolf 359
Proxima Centauri
DX Cancri
Sirius B
WHITE
DWARFS
Procyon B
1 Solar Radius
10 Solar Radii
0.1 Solar Radius
10^{-2} Solar Radius
10^{-3} Solar Radius
luminosity (solar units)
10^6
10^5
10^4
10^3
10^2
10
1
0.1
10^{-2}
10^{-3}
10^{-4}
10^{-5}
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)

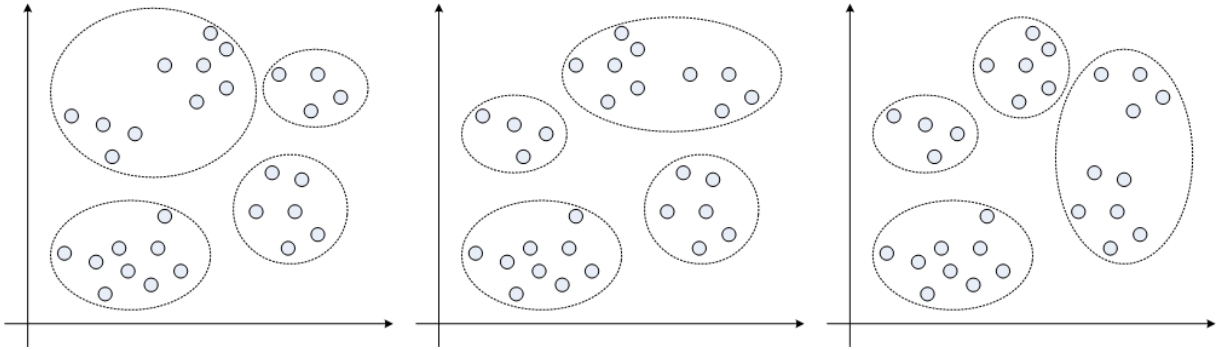
Osnovni problem:

- koliko cluster-a ima ?

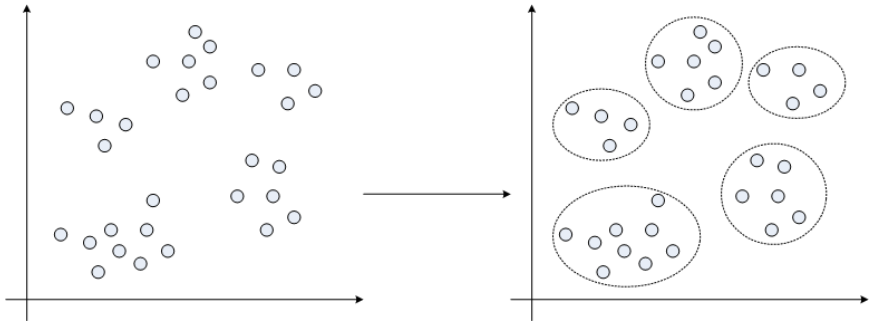
Osnovni problem:

- koliko cluster-a ima ?

- K=4 ?



- K=5 ?



Clustering

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: algoritam

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 x_i najbližoj srednjoj vrijednosti μ_j:

Pripadnost clusteru - C_j (centroid $\boldsymbol{\mu}_j$),

- indikatorska varijablua $\gamma_j{}^i$

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

$\gamma_j{}^i$

	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

3. Izračunaj nove μ_j:

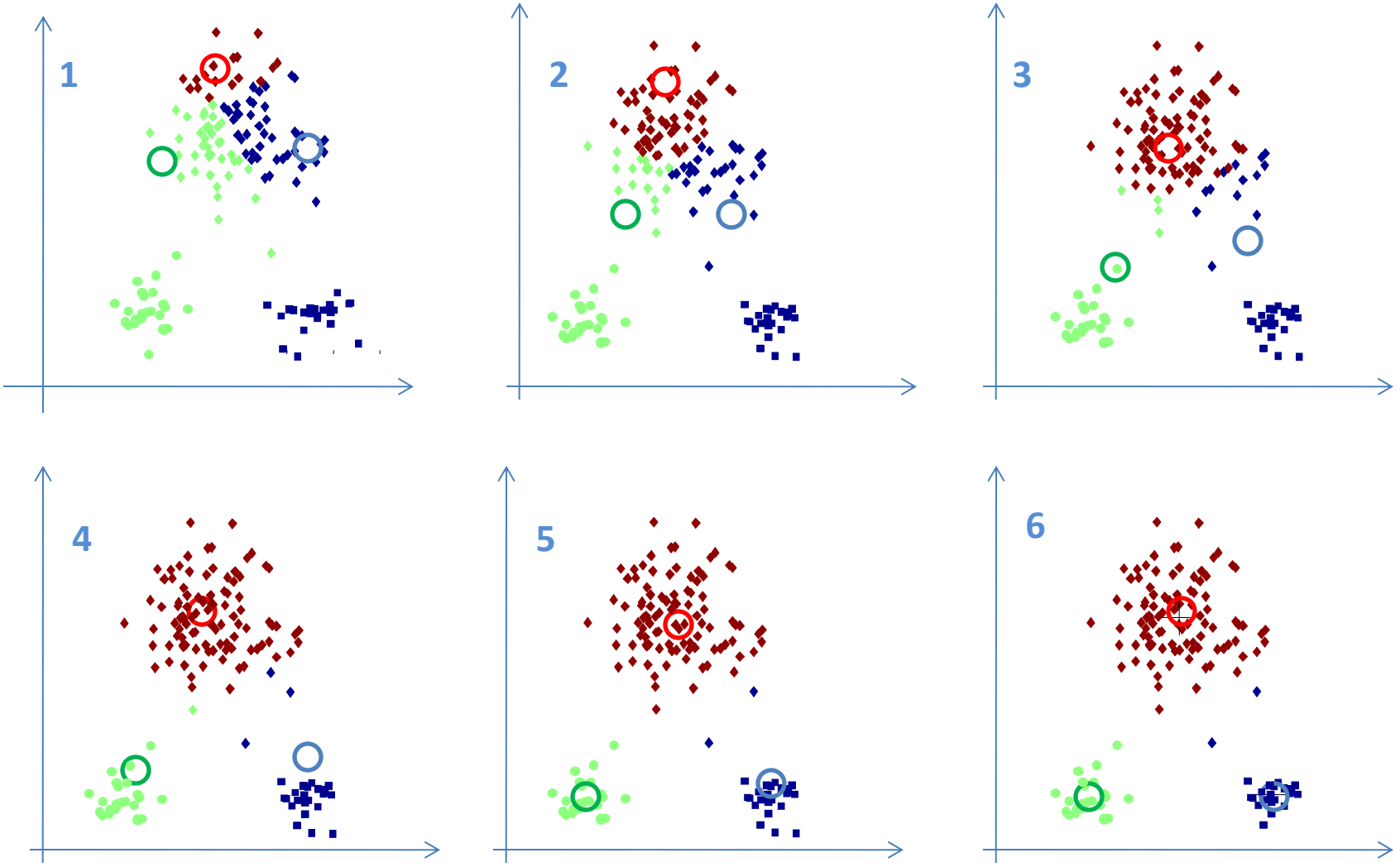
$$\boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{\mathbf{x_i} \in C_j} \mathbf{x_i}$$

4. Ponavljaj 1-3 do konvergencije

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

K=3

K-means: ilustracija

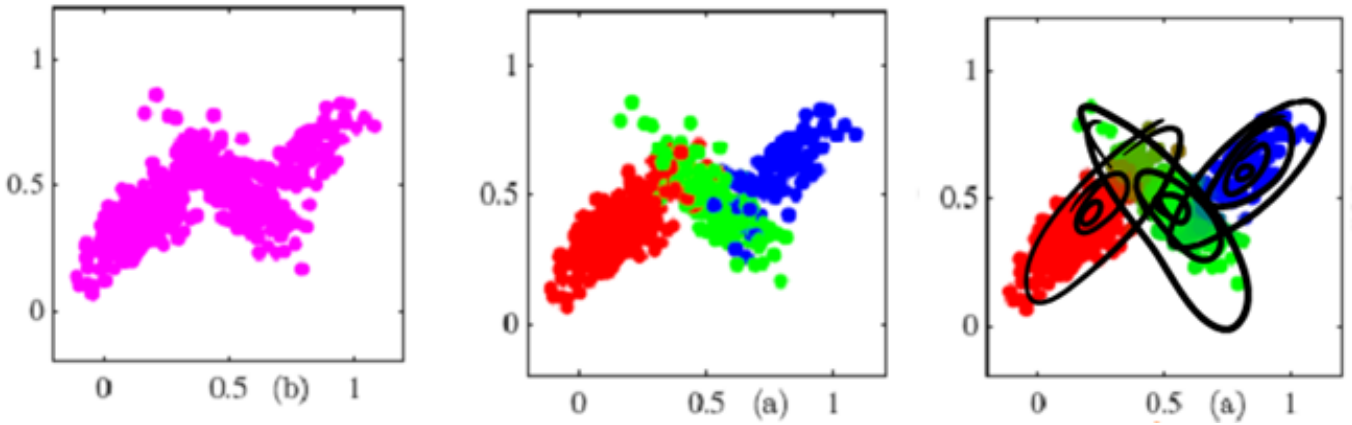


K-means - detalji

- 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

Clustering – K means

(0 ili 1) pripadnost (Hard clustering)



soft clustering (0 ➔1) pripadnost

Mješavina klastera kombinacijom k Gaussove distribucije

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

= 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 $\gamma_c(xi)$ –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. Iniciraj početne parametre $\gamma_{ic}^0, \mu_c^0, \Sigma_c^0$

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

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 - $\gamma_c{}^{t+1}, \mu_c{}^{t+1}, \Sigma_c{}^{t+1}$

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

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

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

Problemi i ograničenja K-means algoritma

- Odabir inicijalnih centroida (slučajan) !?
- Utjecaj “outlier-a” !?
- Karakteristike “stvarnih” cluster-a
  - Oblik, veličina, gustoća
- Koliki je (optimalni) K !?

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

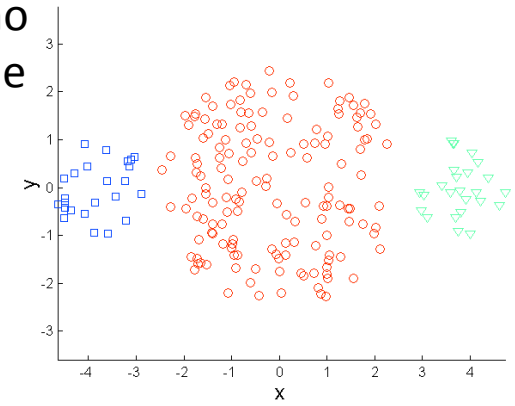
$$DBI = \frac{1}{k}.\sum_{i=1}^{k} R_i$$

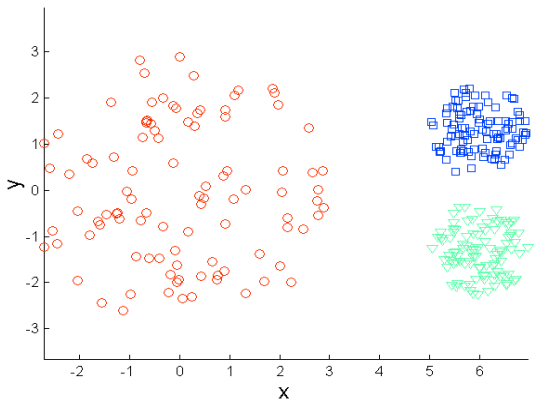
Minimalni DBI => optimalan K;
DBI – usporedba clustering metoda

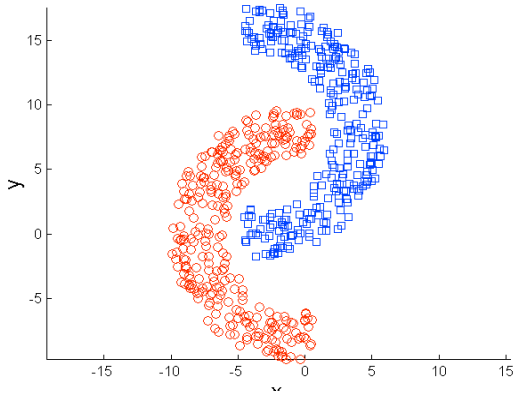
Problemi i ograničenja K-means algoritma: različite veličine

različite veličine | različite gustoće | nekonveksan oblik

Originalno grupiranje

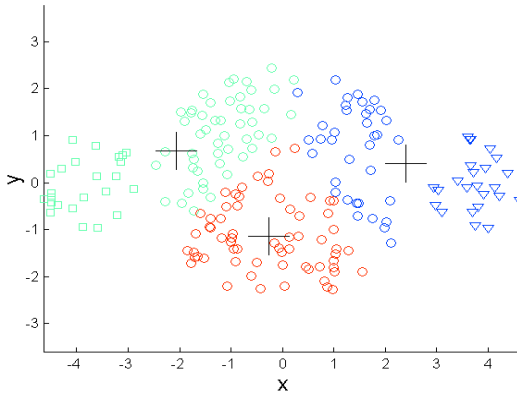


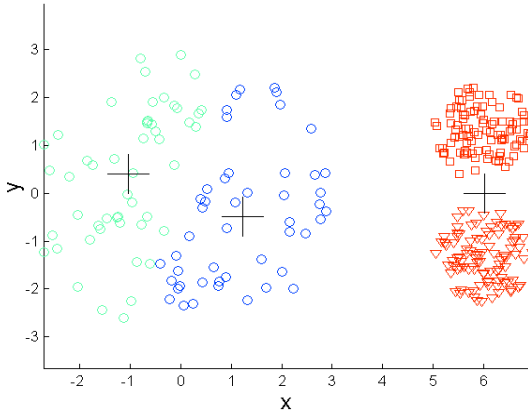


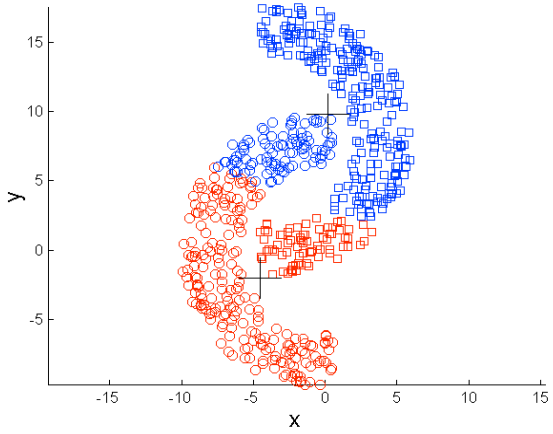


K-means

(K=3)



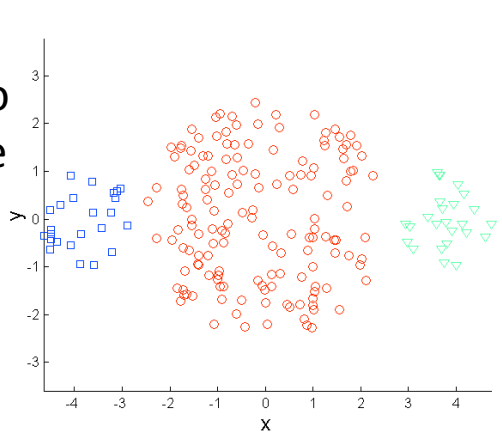


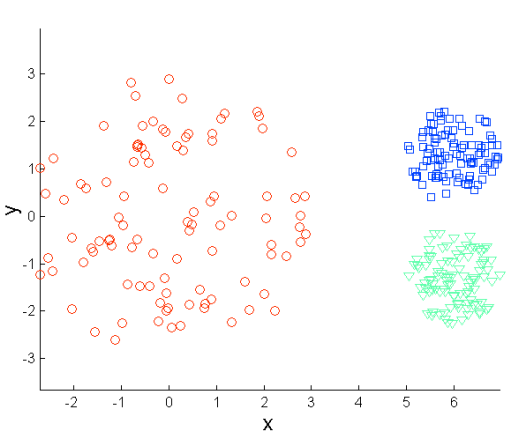


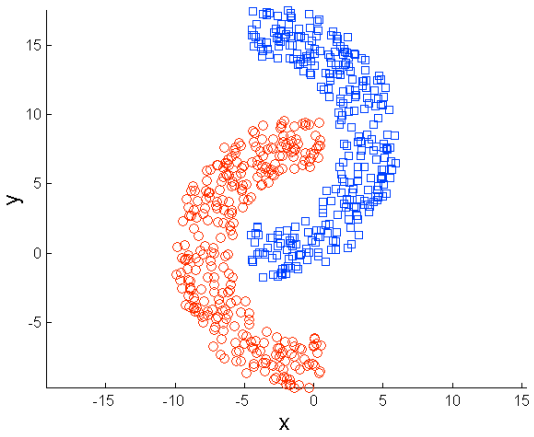
Tipično rješenje: veći K

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

Originalno grupiranje

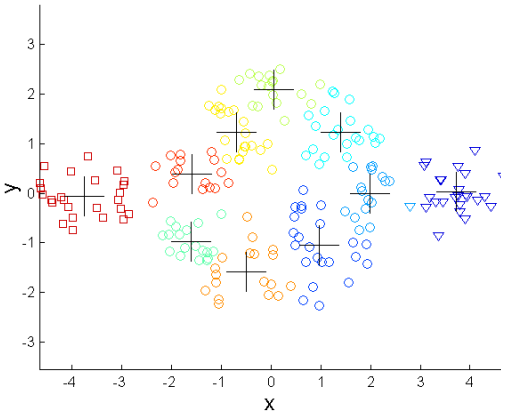


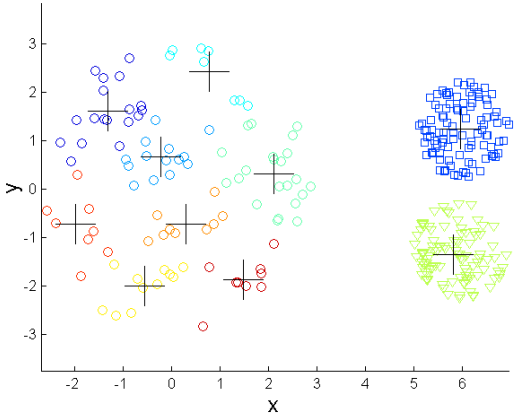


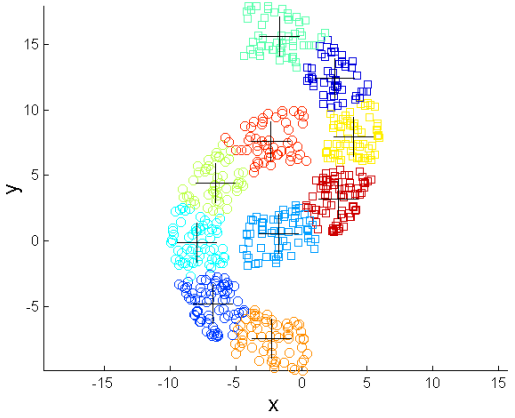


K-means

(K=10)

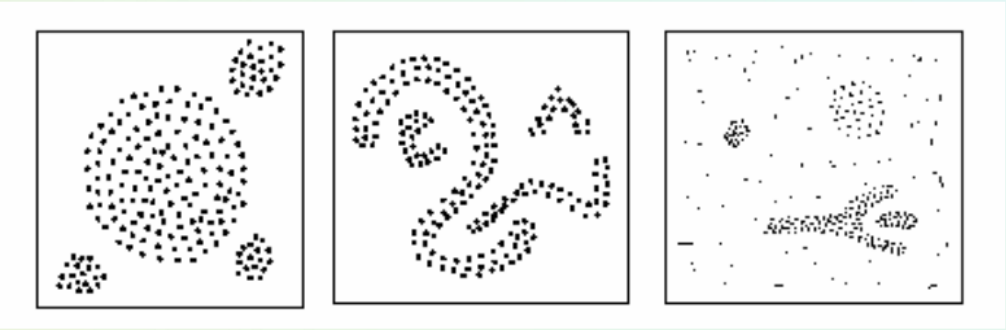






Grupiranje/clustering prema gustoći

Density-Based Clustering



- Clustering based on density (local cluster criterion), such as density-connected points
- Each cluster has a considerable higher density of points than outside of the cluster

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.

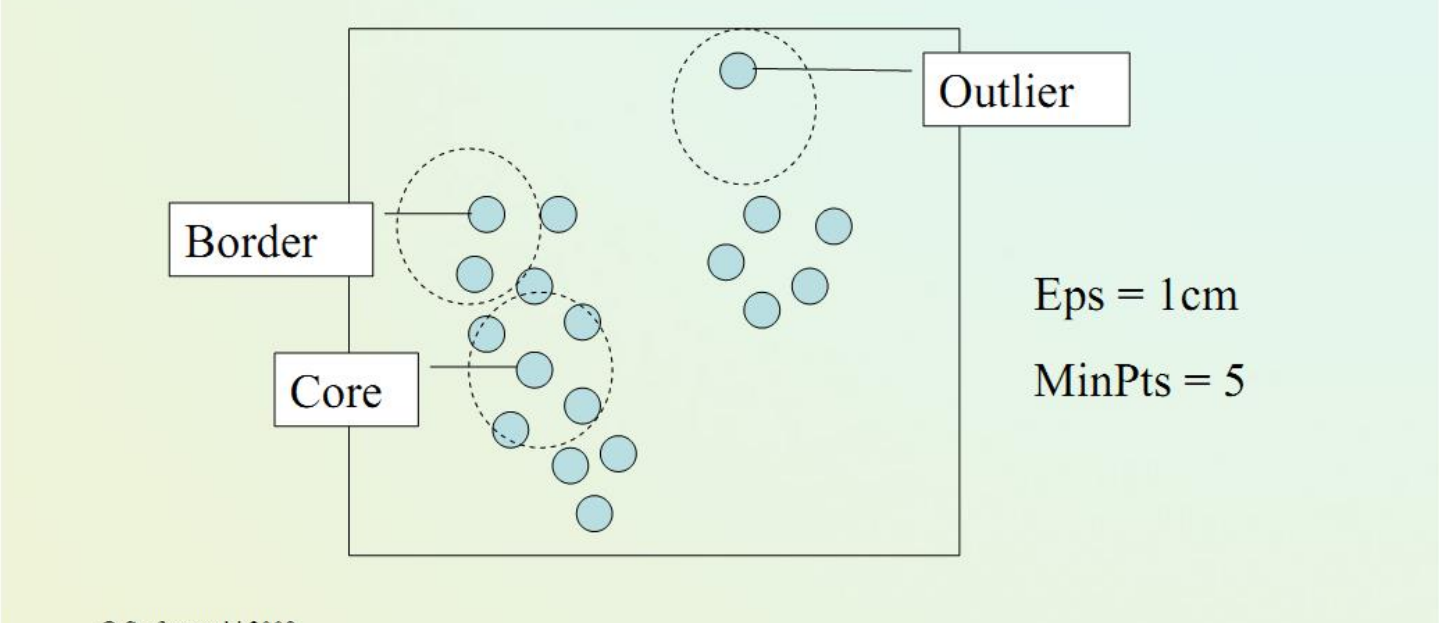
Dostupnost bazirana na gustoći

- **Direct density reachable points:** (**x** je DDR od **y** ako je **x ϵ ε-neighborhood**)
- **Density reachable points:**
  (**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
- **Density connectedness:**
- 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 (istom) gustoćom povezanih točaka

DBSCAN – osnovni koncepti



DBSCAN Algorithm

ALGORITHM 15.1. Density-based Clustering Algorithm

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

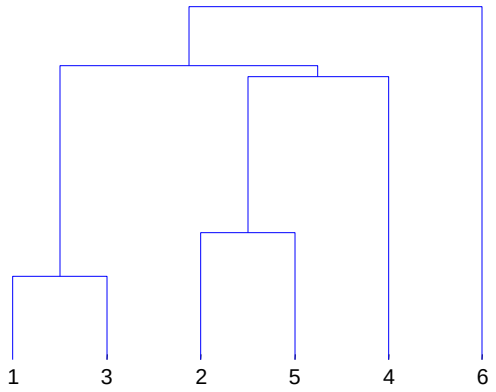
```
1  Core ← ∅
2  foreach x_i ∈ D do // Find the core points
3      Compute N_ε(x_i)
4      id(x_i) ← ∅ // cluster id for x_i
5      if N_ε(x_i) ≥ minpts then Core ← Core ∪ {x_i}
6  k ← 0 // cluster id
7  foreach x_i ∈ Core, such that id(x_i) = ∅ do
8      k ← k + 1
9      id(x_i) ← k // assign x_i 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
```

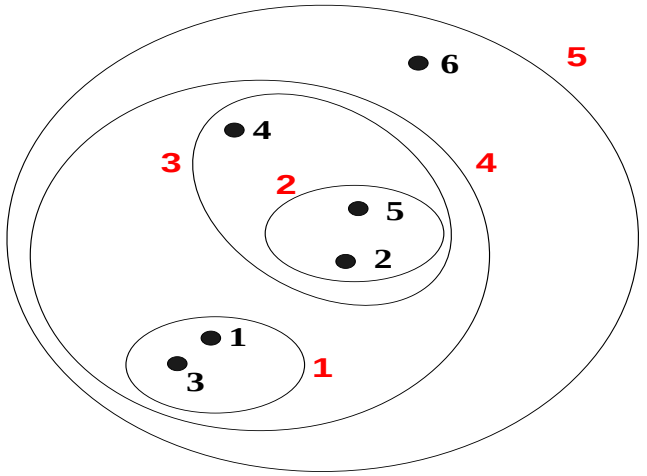
DENSITYCONNECTED $(\mathbf{x}, k)$:

```
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

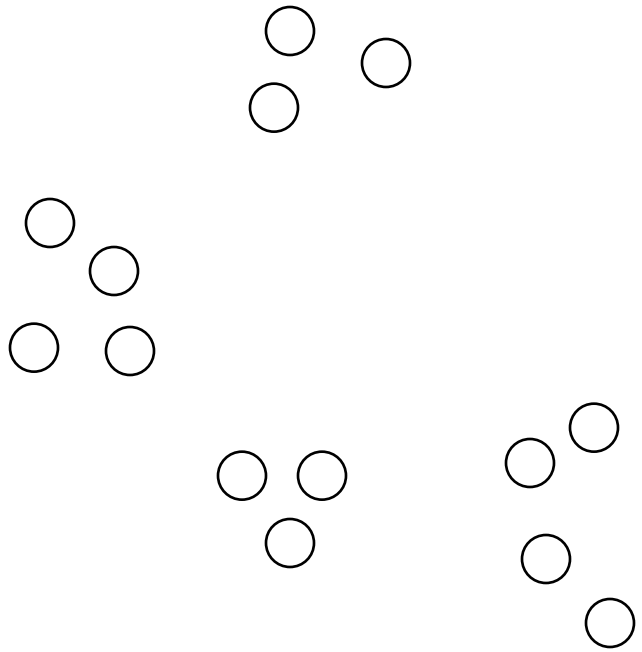
Aglomerativni HC algoritam

- **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

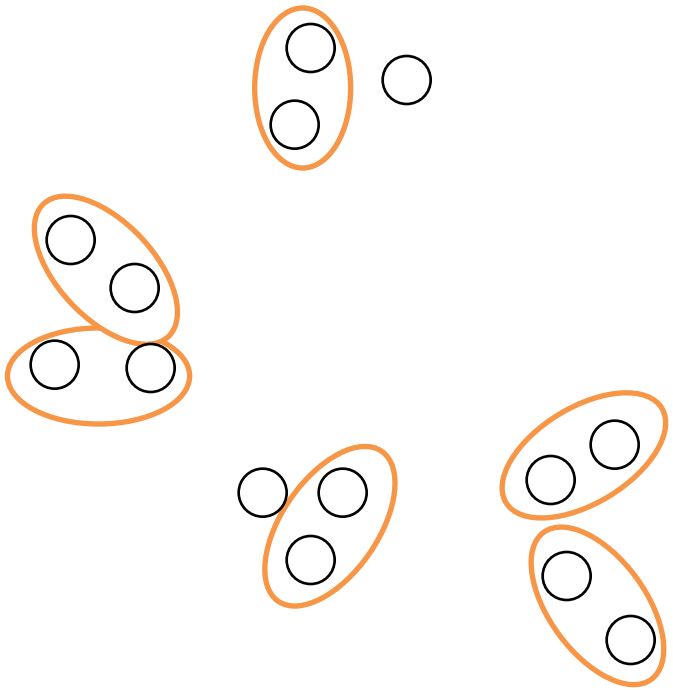


	p1	p2	p3	-...	pn
p1	0	1.1	2.1	...	
P2	1.1	0	3.2	...	
P3	2.1	3.2	0	...	
...	...	...			

Matrica udaljenosti

Aglomerativni HC algoritam

- i. korak

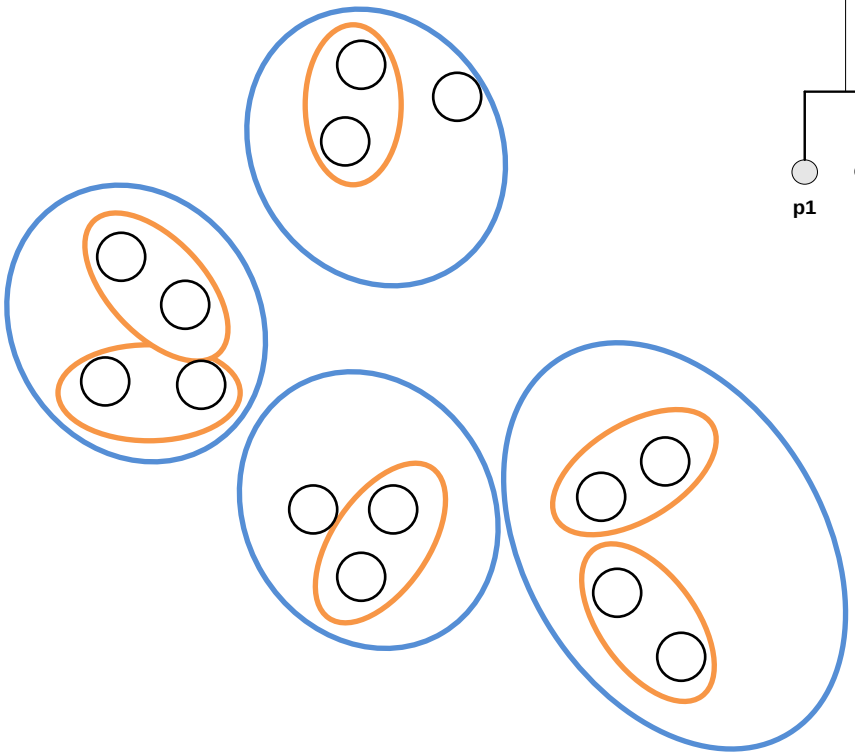


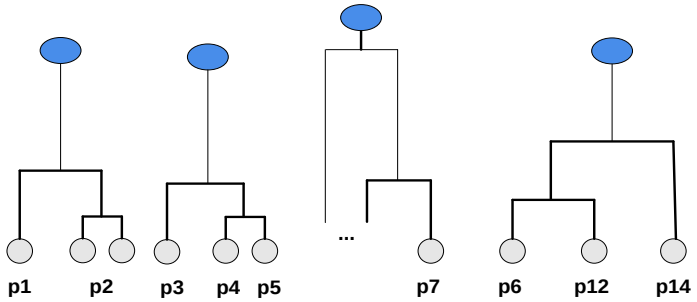
	c1	c2		-...	cj
C1	0	1.1		...	
c3	2.1	3.2		...	
...	...	...			

Matrica udaljenosti

Aglomerativni HC algoritam

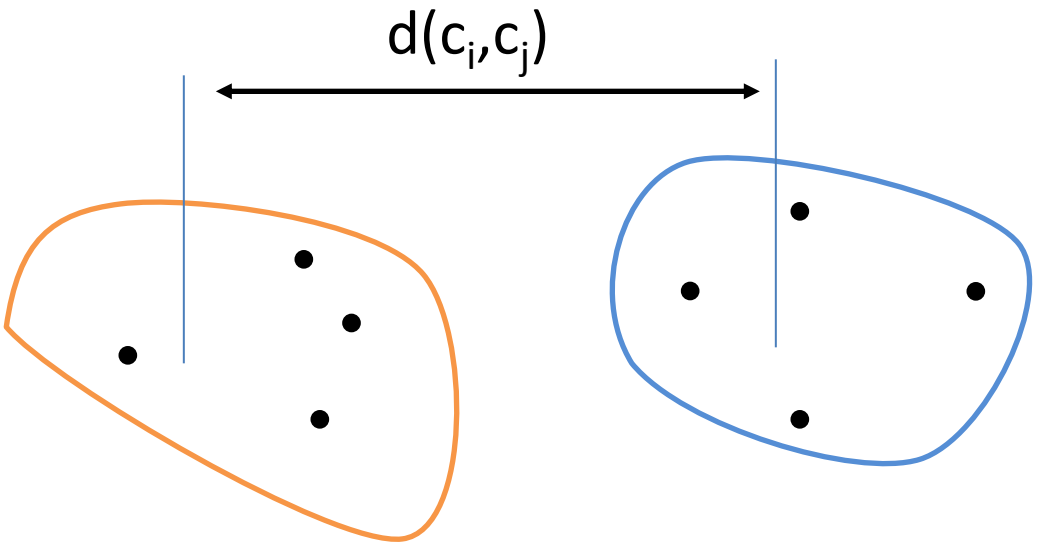
- Zadnji korak (K=4)





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

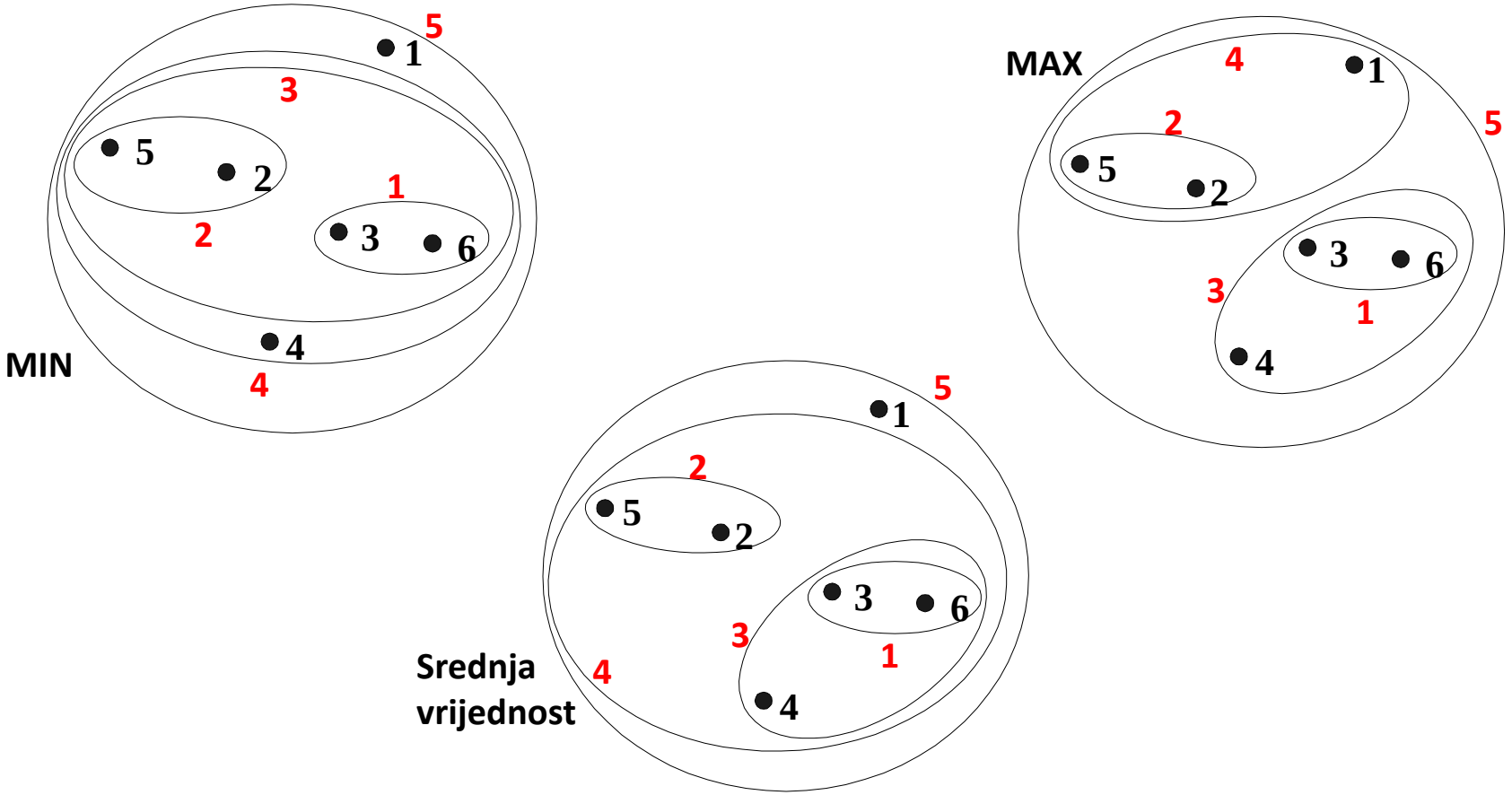
- Zavisno o odabranoj metodi dobit ćemo različiti rezultat !

- **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

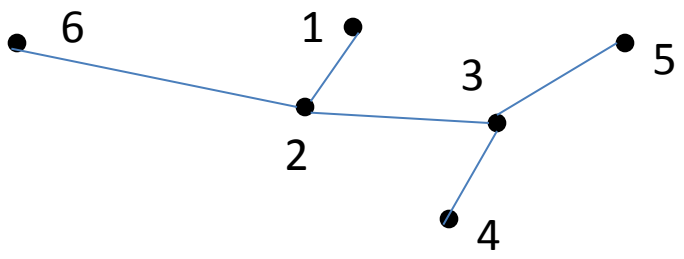
- (N= br primjera – N^2 matrica udaljenosti/sličnosti)

$O(N^3)$ – 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 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)

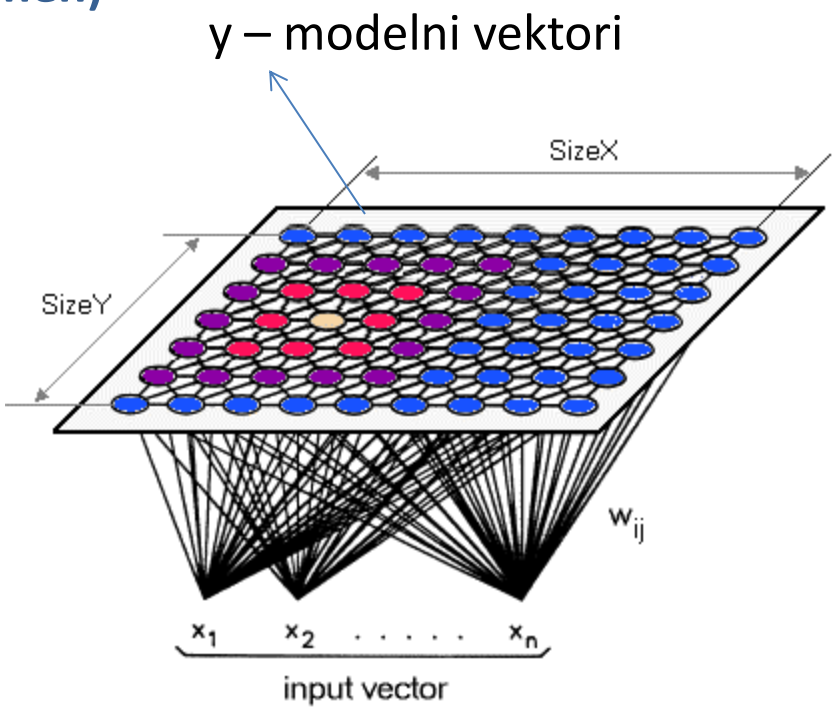
- Cilj: topologija primjera => mapiranje primjera u niže-dimenzionalni prostor, uz uvjet da udaljenosti između primjera budu što je više moguće sačuvane

- 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 (Teuvo Kohonen)

- 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 (mxm, oblik čvorova...)
- incijaliziraj početnu **veličinu susjedstva *D(0)***
- zadaj $0 < \eta(t) \le \eta(t-1) \le 1$ 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

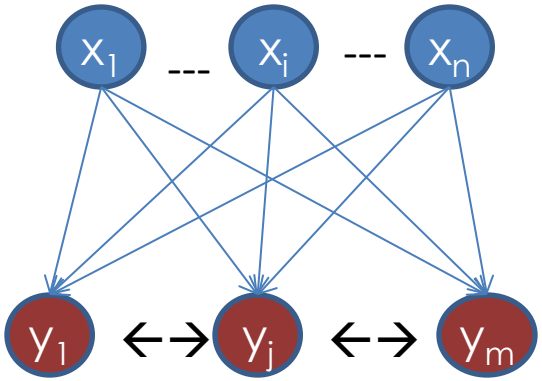
$$\sum_{k=1}^{n} (x_{i,k} - y_{j,k})^2$$

  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***

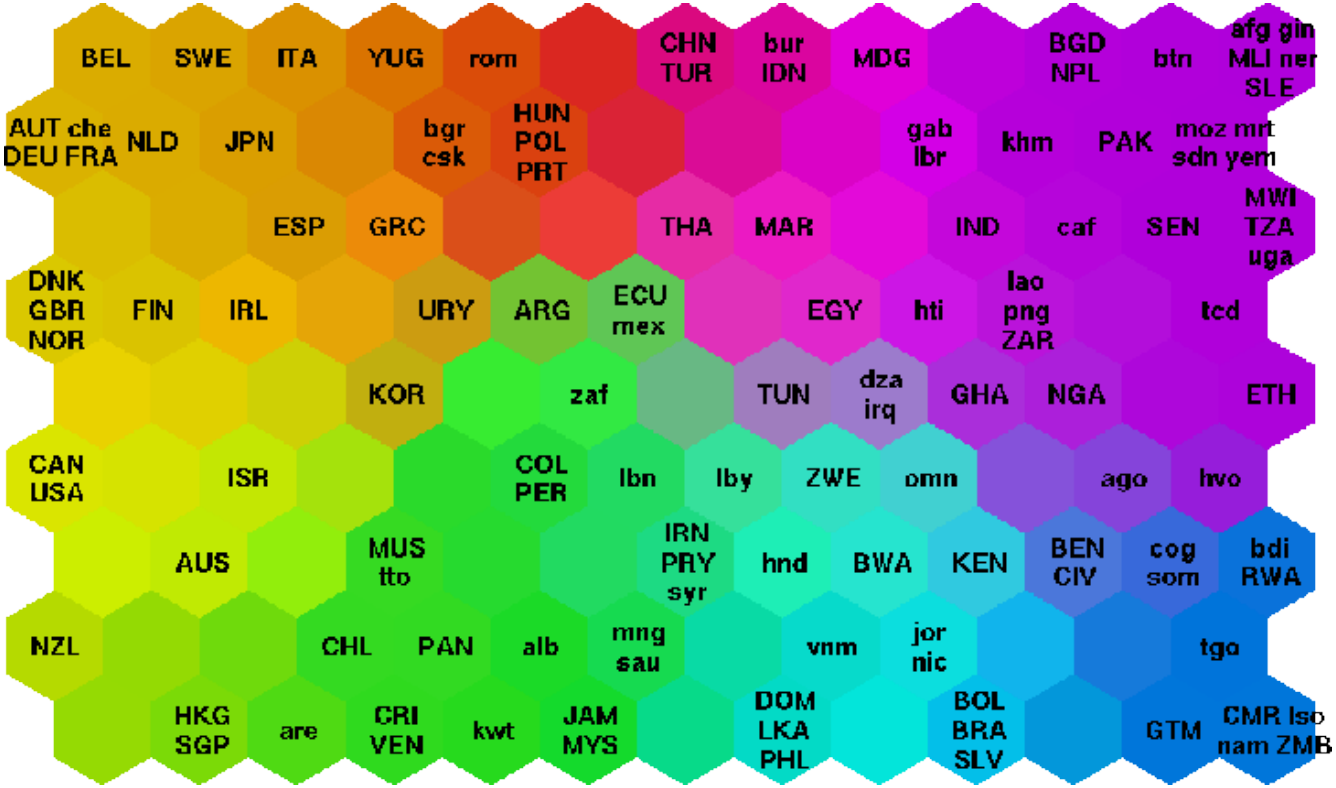
Arhitektura



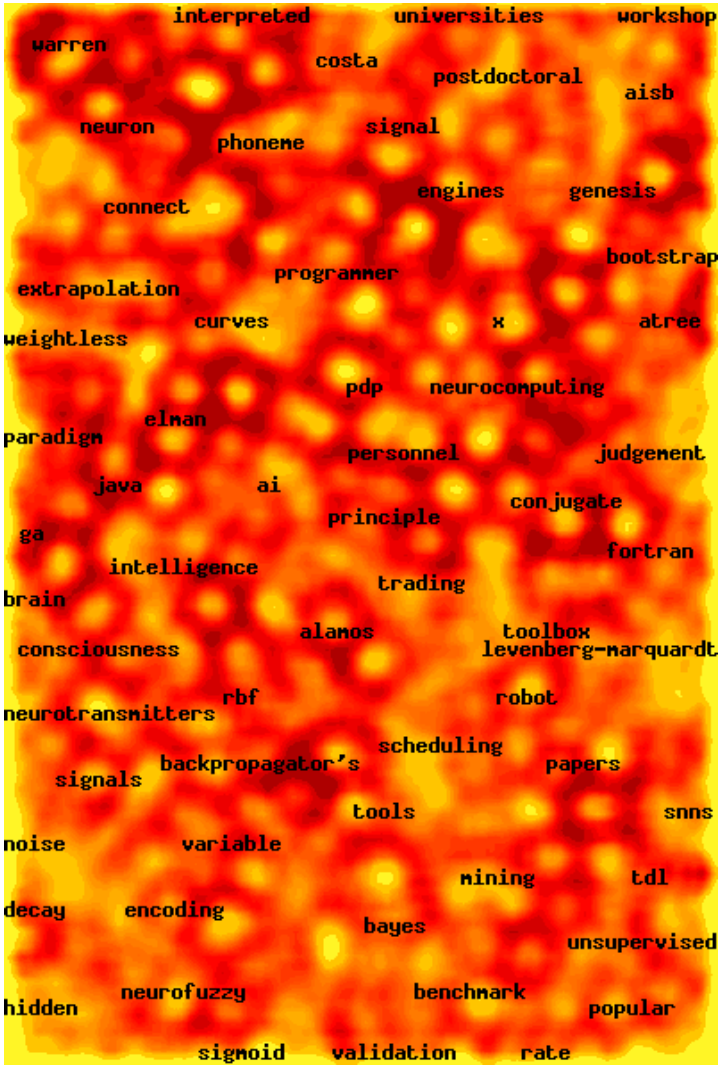
SOM – značaj i primjene

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

SOM:
World poverty map



SOM (Self-organizing-maps)



SOM: comp.ai.neural-nets newsgroup

Metode redukcije dimenzionalnosti

Redukcija dimenzionalnosti

Problem više-dimenzionalnih prostora ("curse of dimensionality", Bellman,1961)

- Problemi vezani uz multivarijantnu analizu vezani uz povećanje dimenzionalnosti

Implikacije više-dimenzionalnosti

- eksponencijalni porast primjera ako želimo sačuvati "gustoću" primjera
- Uz "sačuvanu" gustoću primjera (N primjera/intervalu) i *d* dimenzija, ukupni broj primjera je N^d
- Eksponencijalno raste i kompleksnost ciljne funkcije: funkcija u višedimenzionalnom vjerojatno će biti puno kompleksnija od one u niže-dimenzionalnom prostoru

Dva pristupa – redukciji dimenzionalnosti

- Selekcija podskupa varijabli (onih koje su više-informativne)
- **<u>Ekstrakcija manjeg broja novih dimenzija-varijabli</u>** - kombiniranjem postojećih

Redukcija dimenzionalnosti

Ekstrakcija manjeg broja novih dimenzija-varijabli - kombiniranjem postojećih/originalnih varijabli

Formulacija:

- Treba pronaći za prostor $x_i \in R^N$ mapiranje $y=f(x):R^N \rightarrow R^M$ *uz* $M<(<)N$, tako da za neki transformirani primjer (vektor) većina informacije, ili strukture ostane sačuvana

- U principu bi optimalna mapiranja trebala biti nelinearnog karaktera

- Tradicionalno: najčešće su korištene linearne transformacije

Kad su podaci => matrica uzoraka

- Vrlo tipično za strojno učenje – podaci u formi mjerenja, **vektori svojstava** ili uzorci
- Vektori svojstava (m-dimenzionalni Euklidski prostor)

$$\mathbf{x_i} \in \mathcal{X} \subseteq \mathbb{R}^m, \quad i = 1, \ldots, n$$

- **Matrica uzoraka**

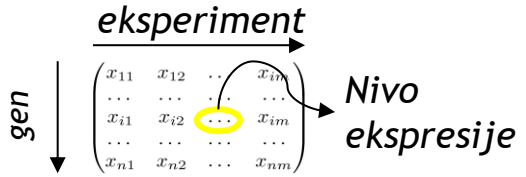
$$\mathbf{X} \in \mathbb{R}^{n \times m}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1' \\ \ldots \\ \mathbf{x}_i' \\ \ldots \\ \mathbf{x}_n' \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{im} \\ \ldots & \ldots & \ldots & \ldots \\ x_{i1} & x_{i2} & \ldots & x_{im} \\ \ldots & \ldots & \ldots & \ldots \\ x_{n1} & x_{n2} & \ldots & x_{nm} \end{pmatrix}$$

i-ti uzorak

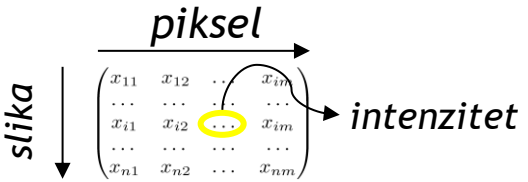
Primjeri matrice uzoraka

- **Mjerenja**
  nivo ekspresije
  - i: primjer, gen
  - j: mjerenje – eksperiment (bolesti)

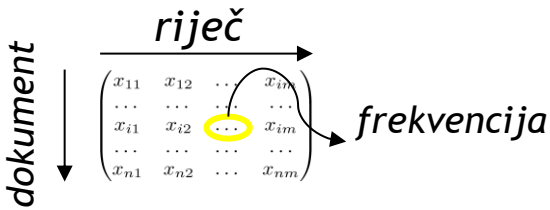
$$\mathbf{X} \in \mathbb{R}^{n \times m}$$



- **Digitalne slike**
  vektori (nijanse sivog)
  - i: slika
  - j: piksel intenzitet na lokaciji $j=(k,l)$

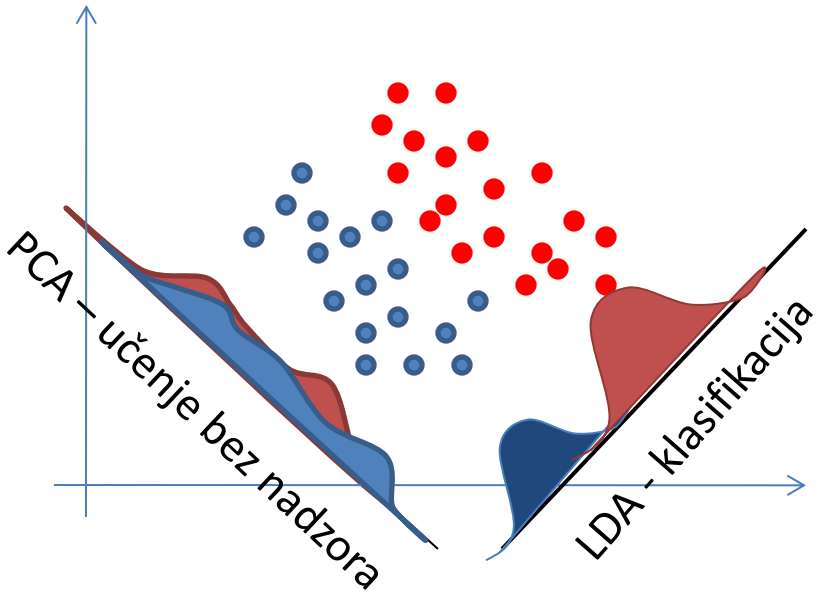


- **Korpus dokumenata**
  tzv. bag-of-words reprezentacija
  - i: dokument
  - j: riječ iz riječnika



Metoda osnovnih komponenata
(PCA - Principal Component Analysis)

PCA - Principal Component Analysis (metoda osnovnih komponenti)



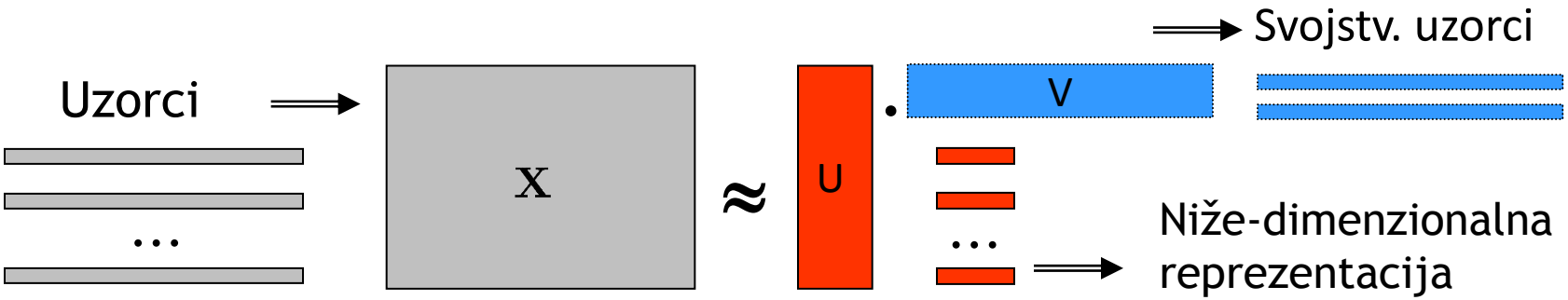
Projekcija zavisi od funkcije cilja koja se optimira

PCA – funkcija cilja je da reprezentacija primjera u niže-dimenzionalnom prostoru mora biti što točnija (sačuvanje udaljenosti izm. Primjera)

LDA – funkcija cilja je da reprezentacija primjera u niže-dimenzionalnom prostoru ima što bolja klasifikacijska svojstva (razdvajanje klasa)

Redukcija dimenzija - PCA

- **Osnovna ideja PCA:** redukcija dimenzionalnosti (*m* ➔ p: p<<m)
- **Pretpostavka (uobičajena):** *m* je velik broj zavisnih varijabli (koreliranih)
  - Intuitivno – želimo zadržati što je više moguće originalnih odnosa/varijacije između podataka/uzoraka
- **Postupak**: transformacija u novi skup (međusobno nekoreliranih) varijabli koje su i rangirane tako da **one prve u rangu zadržavaju najveći dio varijacije koja je prisutna u svim originalnim varijablama**.



- **Podaci**: matrica uzoraka (slike, tekstovi, muzika, ekspresije gena,)
- **PCA = Latentna struktura**: niže-dimenzionalni podprostor
- **Dekompozicija prostora** ➔ svojstvene vrijednosti, svojstveni vektori

Redukcija dimenzija – PCA – Optimalna rekonstrukcija

- **Pearson, 1901**: **PCA = Ortogonalna linearna projekcija s minimalnom greškom – u smislu najmanjih kvadrata**
- Izraziti uzorke u novoj ortogonalnoj bazi

$$\mathbf{x}_i = \sum_j w_{ij}\mathbf{v}_j, \quad w_{ij} \equiv \langle \mathbf{x}_i, \mathbf{v}_j \rangle \quad \{\mathbf{v}_1, \ldots, \mathbf{v}_d\}$$
$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij}$$

- **Niže-dimenzionalna aproksimacija** (kao linearna projekcija)

$$\hat{\mathbf{x}}_i = \underbrace{\sum_{j=1}^{q} w_{ij}\mathbf{v}_j}_{\textit{Vektori koje zadržavamo}} + \underbrace{\sum_{j=q+1}^{m} c_j\mathbf{v}_j}_{\textit{Vektori koje odbacujemo}}, \quad q \leq m$$

PCA - Principal Component Analysis (metoda osnovnih komponenti)

Uz dani skup točaka $\mathbf{x} \in \mathbf{R}^m$ želimo projicirati svaki **x** u *d<m* (niže-dimenzionalni) podprostor sa $\mathbf{z}=[z_1,z_2,\ldots,z_d] \in \mathbf{R}^d$ tako da je:

$$\mathbf{x} = \sum_{i=1}^{m} z_i u_i$$

- vektori $\mathbf{u}_i$ zadovoljavaju kriterij ortonormalnosti:

$$\mathbf{u}_i^{\mathrm{T}} \mathbf{u}_j = \delta_{ij}$$

Mi želimo koristiti d<m. Ostali koeficijenti b_i i $\mathbf{u}_i$ omogućavaju aproksimaciju $\tilde{\mathbf{x}}$

$$\tilde{\mathbf{x}} = \sum_{i=1}^{d} z_i \mathbf{u}_i + \sum_{i=d+1}^{m} b_i \mathbf{u}_i$$

PCA - Principal Component Analysis (metoda osnovnih komponenti)

Za svaki **x** – greška koju činimo zbog redukcije dimenzionalnosti je:

$$\mathbf{x} - \tilde{\mathbf{x}} = \sum_{i=d+1}^{m} (z_i - b_i)\mathbf{u}_i$$

Želimo naći $\mathbf{u}_i$ koeficijente b_i, i vrijednosti z_i **s** najmanjom greškom

Za čitav skup podataka - uz relaciju ortonormalnosti vrijedi:

$$E_d = \frac{1}{2}\sum_{k=1}^{n} \left\| \mathbf{x}^k - \tilde{\mathbf{x}}^k \right\|^2 = \frac{1}{2}\sum_{k=1}^{n}\sum_{i=d+1}^{m} \left\| z_i^k - b_i^k \right\|^2$$

PCA - Principal Component Analysis (metoda osnovnih komponenti)

Izvod - minimizacija E_d po **u,** vodi na koncu do:

$$\mathbf{C}\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

Gdje je **C** matrica kovarijance

$$\mathbf{C} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}^k - \bar{\mathbf{x}})(\mathbf{x}^k - \bar{\mathbf{x}})^{\mathrm{T}}$$

A vektori $\mathbf{u}_i$ su svojstveni vektori matrice **C.** Konačno, E_d je minimalno za slučaj kad se iz rekonstrukcije odbace svojstveni vektori čije su svojstvene vrijednosti najmanje:

$$E_d = \frac{1}{2}\sum_{i=d+1}^{m}\lambda_i$$

Redukcija dimenzija – PCA preko SVD

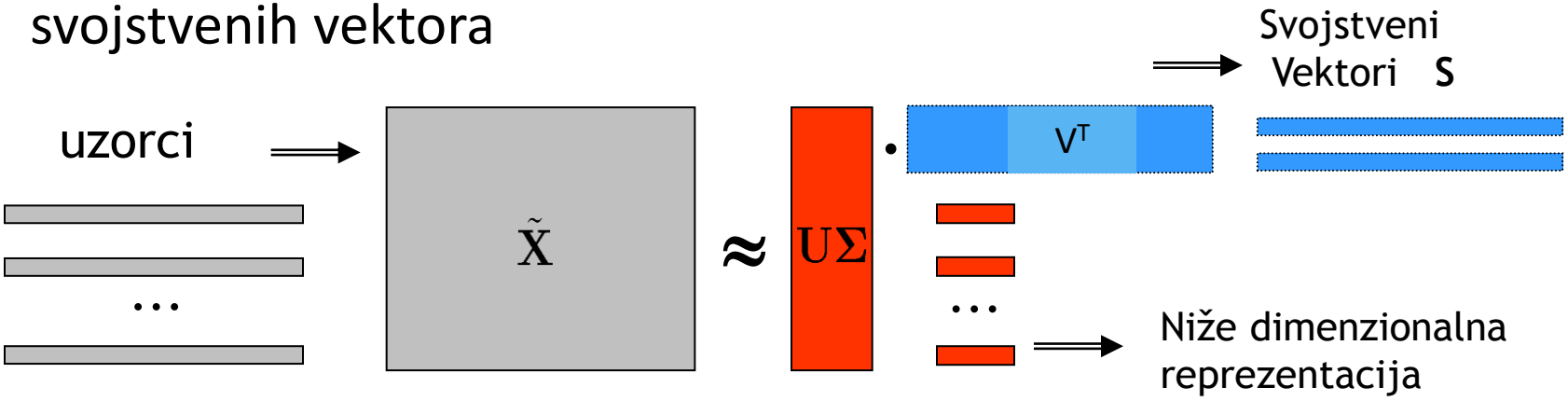
- SVD matrice uzoraka se može koristiti za izračunavanje PCA

$$\tilde{\mathbf{X}} = \mathbf{U}\sum\mathbf{V}^T \Rightarrow$$

$$S = \frac{1}{n}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} = \frac{1}{n}(\mathbf{V}\sum\mathbf{U}^T)(\mathbf{U}\sum\mathbf{V}^T) = \frac{1}{n}\mathbf{V}\sum^2\mathbf{V}^T$$

Redovi $\mathbf{V}$ su svojstveni vektori $\mathbf{S}$

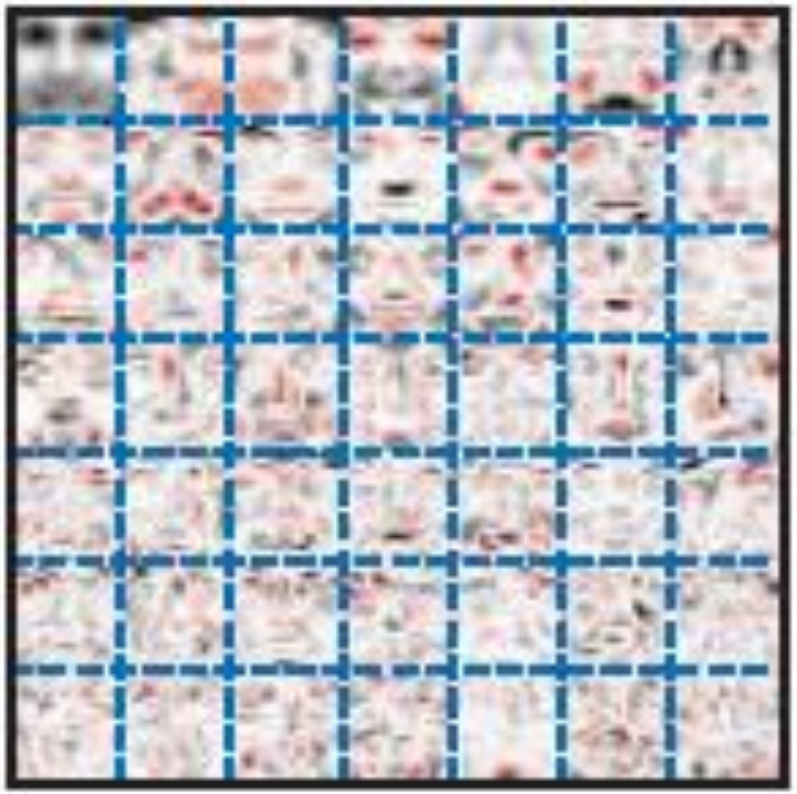
- $\tilde{\mathbf{X}}V = \mathbf{U}\sum$ ➔ PC težine ➔ unutarnji produkt uzoraka i svojstvenih vektora



Redukcija dimenzija – kako izgledaju PC?

- Primjena na slike lica
  - Uzorak ⟹ intenzitet sivog po pikselima slike
- Eigenfaces

Prosječno lice



[Lee & Seung 1999]

⟸ osnovne komponente

PCA - primjene

Redukcija dimenzija – kao priprema podataka za druge algoritme ili analizu podataka (tipično za probleme vezane uz prepoznavanje slika (lica))

Otkrivanje niže-dimenzionalnih struktura u više-dimenzionalnom prostoru (tzv. data manifolds)

Vizualizacija podataka, interpretacija - 2D-3D grafovi podataka

Eliminacija šuma iz podataka - otkrivanje i eliminacija outlier-a

Ograničenja

Linearnost

-> postoje nelinearne varijante (kernel PCA)...

Dekoreliranost osnovnih komponenti nije i nezavisnost

-> metoda nezavisnih komponenti (ICA)

Aditivni model (ograničenja na predznak koeficienata (tzv. Loadings)

-> NMF metoda (nonnegative matrix factorization)

Ne-Negativna Matrična Dekompozicija

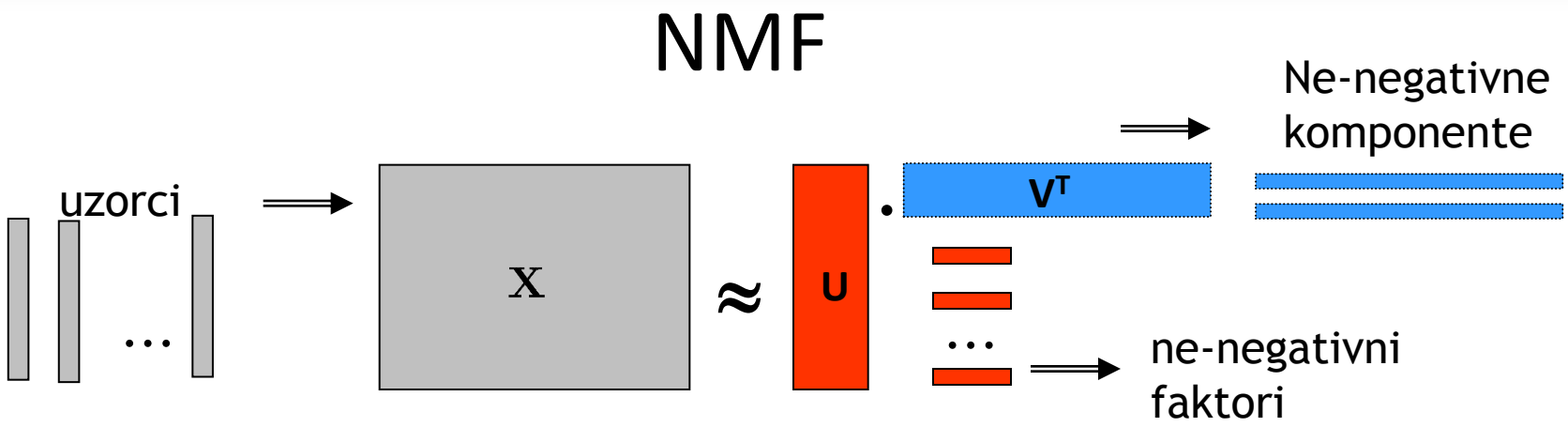
(NMF - Non-Negative Matrix Decomposition)

NMF

- Aproksimativna (niži rang) matrična dekompozicija s ne-negativnim faktorima

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{U}\mathbf{V}^T, \quad \mathrm{U} \in \Re_{\geq o}^{m\,x\,q}, \mathrm{V}^{\mathrm{T}} \in \Re_{\geq o}^{q\,x\,n}$$

- Motivacija
  - Primjena za ne-negativne matrice (matrice uzoraka sa ne-negativnim mjerenjima)
  - Prethodno znanje – latentni faktori su pozitivni !
  - Efekt faktora – kumulativni/aditivni (nema poništavanja efekata zbog negativnih doprinosa!)



- **Podaci**: Matrica uzoraka
- **Latentna struktura**: niže-dimenzionalni podprostor definiran **ne-negativnim baznim vektorima**
- **Dekompozicija**: ne-konveksni opt. problem

Redukcija dimenzionalnosti: NMF

$$\hat{\mathbf{X}} = \mathbf{U}\mathbf{V}^T$$

- **Kvadratna pogreška (Frobeniusova norma)**

$$E(\mathbf{X}, \hat{\mathbf{X}}) = \sum_{i=1}^{n}\sum_{i=1}^{m}(x_{ij} - \hat{x}_{ij})^2 = \left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_F$$

- **Na bazi Kullback-Leibler divergencije**
  (tipično se koristi za udaljenosti dviju distribucija)

$$E_{div}(\mathbf{X}, \hat{\mathbf{X}}) = \sum_{i=1}^{n}\sum_{i=1}^{m}(x_{ij}\log\frac{x_{ij}}{\hat{x}_{ij}} - x_{ij} + \hat{x}_{ij})$$

Uz uvjet $\sum_{i,j}\hat{x}_{ij} = const.$

Redukcija dimenzionalnosti: NMF

Metoda/algoritam za faktorizaciju matrica (pozitivnih) u 2 ne-negativne matrice:

$$\mathbf{X} = \mathbf{U}\mathbf{V}^T \qquad \begin{bmatrix} x_{11} & x_{12} \ldots . x_{1n} \\ x_{21} & x_{22} \ldots . x_{2n} \\ \ldots\ldots \\ \ldots\ldots \\ x_{m1} & x_{m2} \ldots . x_{mn} \end{bmatrix} = \begin{bmatrix} u_{11}\, u_{12} \ldots . u_{1k} \\ u_{21}\, u_{22} \ldots . u_{2k} \\ \ldots \\ u_{m1}\, u_{\mathrm{m}2} \ldots . u_{\mathrm{m}k} \end{bmatrix} . \begin{bmatrix} v_{11}\, v_{12} \ldots . v_{1n} \\ v_{21}\, v_{22} \ldots . v_{2n} \\ \ldots \\ v_{k1}\, v_{k2} \ldots . v_{kn} \end{bmatrix}$$

- Minimiziraj $$J = \frac{1}{2}\left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_F$$

- Gdje je $\|\mathbf{X}\|_F \Rightarrow \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|x_{ij}|^2}$ - Frobeniusova norma
  + uvjet da vrijede ograničenja na ne-negativnost: $\boldsymbol{u_{ij} \geq 0}$ **i** $\boldsymbol{v_{ij} \geq 0}$!!

➔ vodi na Lagrange-ovu funkciju !!

Lagrangeova funkcija

$$L = J + tr(\alpha \mathbf{U}^T) + tr(\beta \mathbf{V}^T)$$

Uz Derivaciju L =0

$$\frac{\partial L}{\partial \mathbf{U}} = -\mathbf{XV} + \mathbf{UV}^T\mathbf{V} + \boldsymbol{\alpha}$$

$$\frac{\partial L}{\partial \mathbf{V}} = -\mathbf{X}^T\mathbf{U} + \mathbf{VU}^T\mathbf{U} + \boldsymbol{\beta} \quad \Rightarrow \quad \alpha_{ij}u_{ij} = \beta_{ij}v_{ij} = 0 \quad \Rightarrow$$

i uz Kuhn - Tuckerove uvjete:

$$(\mathbf{XV})_{ij}u_{ij} - (\mathbf{UV}^T\mathbf{V})_{ij}u_{ij} = 0$$

$$(\mathbf{X}^T\mathbf{U})_{ij}v_{ij} - (\mathbf{VU}^T\mathbf{U})_{ij}v_{ij} = 0$$

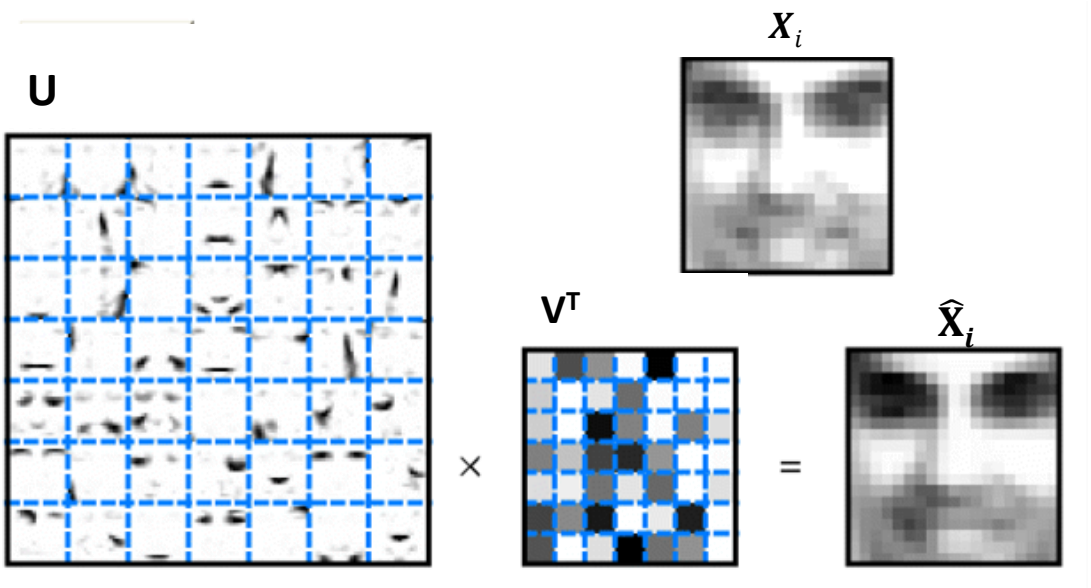
Gornji izrazi vode na iterativni algoritam za „osvježavanje” u_{ij} i v_{ij}

$$u_{ij} \leftarrow u_{ij}\frac{(\mathbf{XV})_{ij}}{(\mathbf{UV}^T\mathbf{V})_{ij}}$$

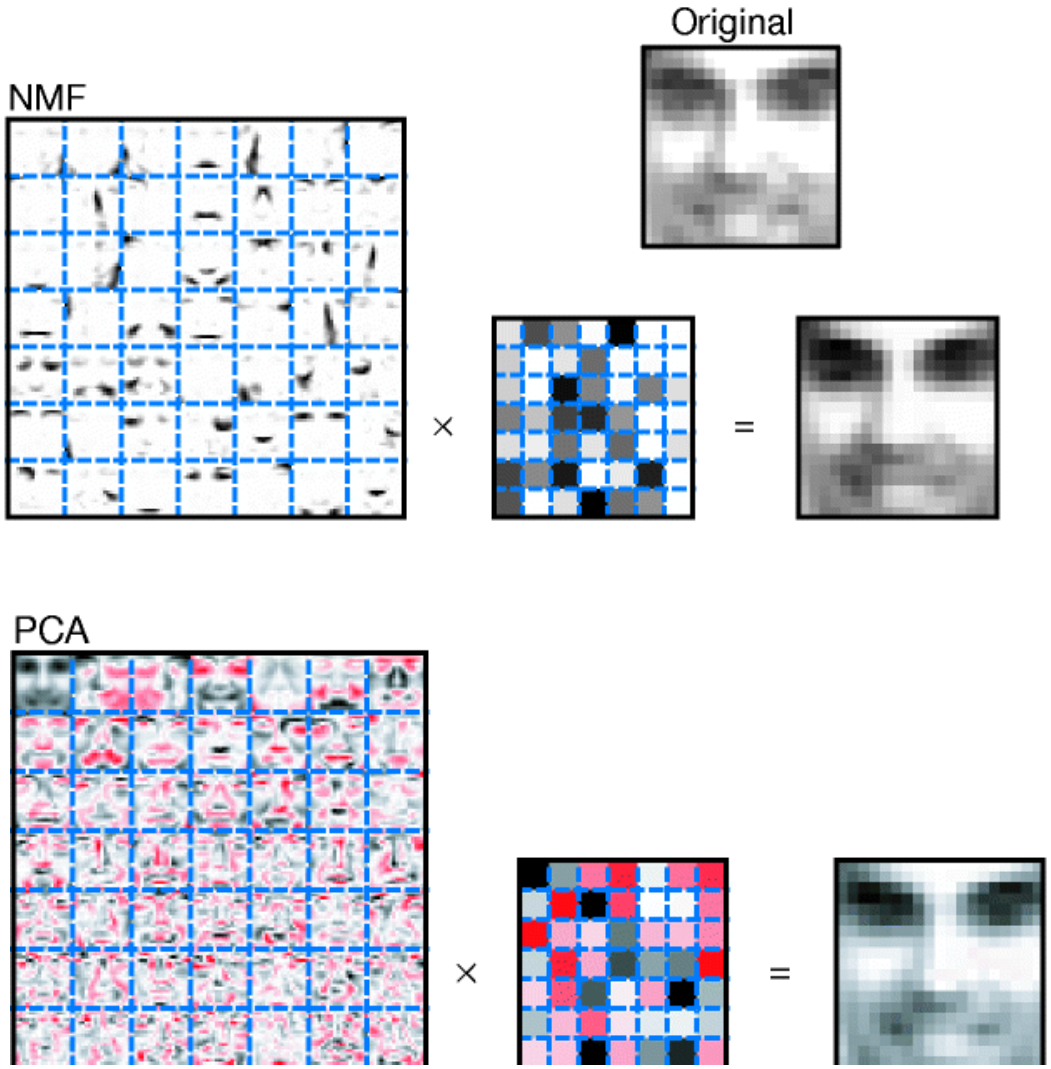
$$v_{ij} \leftarrow v_{ij}\frac{(\mathbf{X}^T\mathbf{U})_{ij}}{(\mathbf{VU}^T\mathbf{U})_{ij}}$$

NMF - Primjena u slikama

- NMF automatska detekcija **dijelova** u slikama ili mjerenjima
- Digitalne slike ➔ matrice uzoraka sa ne-negativnim vrijednostima (intenzitet)
- Faktori korespondiraju sa lokaliziranim uzorcima
- **Eigenfaces** (NMF) ➔ dijelovi lica:



PCA vs NMF



Metoda Nezavisnih Komponenata
(ICA - Independent Component Analysis)

ICA - Independent Component Analysis

Model

$$\mathbf{x} = \mathbf{A} \cdot \mathbf{s} + \varepsilon \qquad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Signal / Uzorak

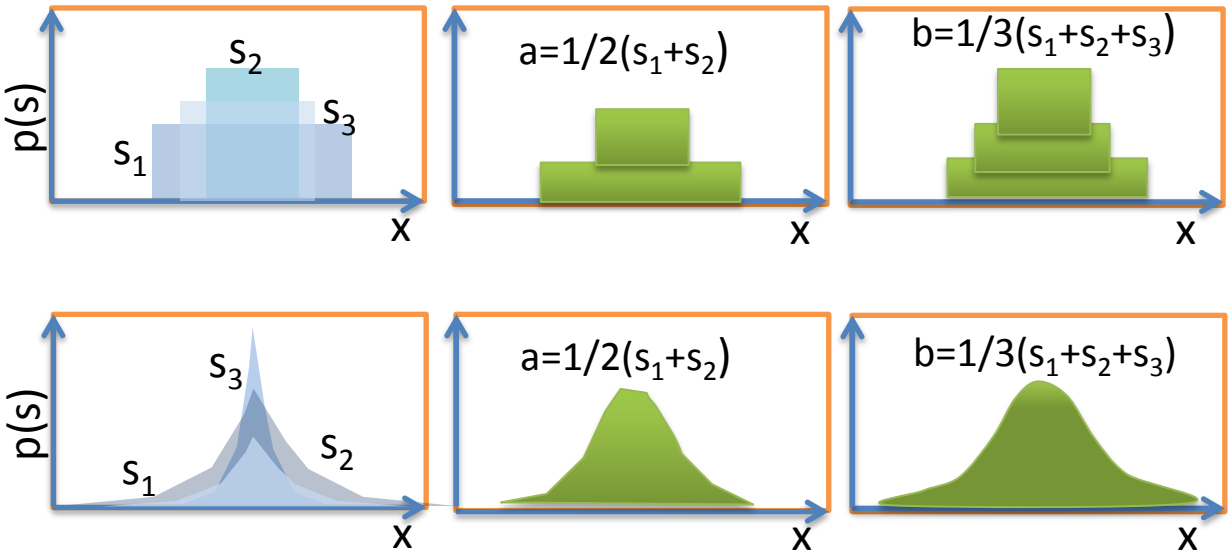
Matrica miješanja

Nezavisne komponente

- Nepoznati su i **A** i **s** (odnosno $\mathbf{W}=\mathbf{A}^{-1}$ => **s=WX)**
- Latentne varijable **s -** ne-Gaussovske (apriorne) distribucije
- **s – međusobno nezavisne**
  (nezavisnost ≠ nekoreliranost – PCA)

ICA – maksimiziranje ne-gaussovskog ponašanja

- ICA funkcionira za slučajeve kada je maksimalno jedan od izvora Gaussovskog tipa, ostali moraju biti ne-Gaussovske distribucije
- Linearno miješanje nezavisnih slučajnih varijabli čini ih sve bliže Gaussovskoj raspodjeli (Centralni granični teorem)



➔ Za odvajanje komponenti (ne-Gauss.) – treba ići što dalje od Gaussove distribucije

➔ Mjere za razdvajanje: viši momenti distribucija (3,4 – kurtosis), entropija

ICA –nezavisnost komponenti

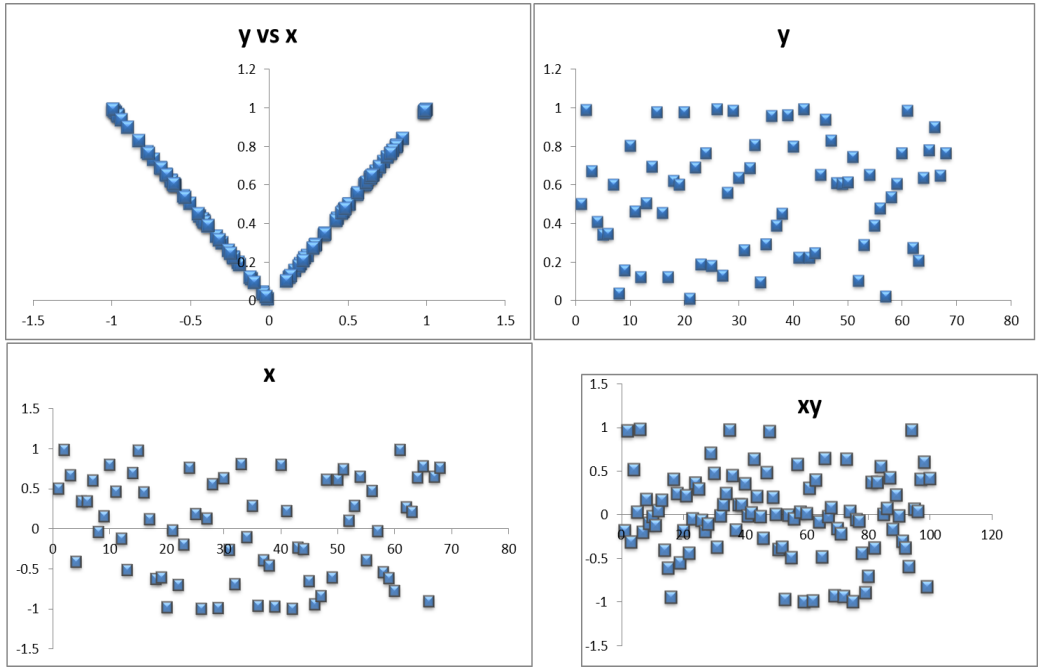
Primjer: Nezavisnost ≠ nekoreliranost

$x\epsilon[-1,1]$ – uniformna distribucija
$\mathrm{y} = -x\colon\ x\ \leq 0$
$\mathrm{y} = x\colon\ x\ > 0$

$E[\mathrm{x}] = 0$
$E[\mathrm{y}] = 0.5$
$E[\mathrm{xy}] = 0$

$\mathrm{cov}[\mathrm{x},\mathrm{y}] = E[xy] - E[x]E[y]$
$\mathrm{cov}[\mathrm{x},\mathrm{y}] = 0$ => **x,y ne-korelirani**

$E[xy|x \leq 0] = -1/3$
$E[xy|x > 0] = +1/3$
$E[y|x] \neq E[y]$ **=> y zavisno o x !**



ICA – maksimiziranje ne-gaussovskog ponašanja

- Linearno miješanje nezavisnih slučajnih varijabli čini ih "više Gaussijanskim" (Central Limit Theorem)

$$\langle \mathbf{v}, \mathbf{x} \rangle = \langle \mathbf{v}, \mathbf{W}\mathbf{s} \rangle = \langle \mathbf{W}'\mathbf{v}, \mathbf{s} \rangle = \langle \mathbf{z}, \mathbf{s} \rangle, \ \mathbf{z} = \mathbf{W}'\mathbf{v}$$

Vektor težina nad varijablama u originalnom prostoru

Inducirane kombinacije težina Za nezavisne komponente

- Cilj: naći kombinaciju težina koja je maksimalno „ne-Gaussijanska"

Redukcija dimenzionalnosti: ICA – kontrastne funkcije

- Mjere odstupanja od Gaussove distribucije **minimiziramo tzv. kontrastne funkcije**
- **„šiljatost"** (**kurtosis** - kumulant 4 reda):

$$K(z) = E[z^4] - 3(E[z^2])^2$$

> 0: **super Gausijanska d.**
< 0: **sub Gausijanska d.**

- **Negativna entropija (negentropy)**

$$J(z) = H[z_{gauss}] - H[z]$$

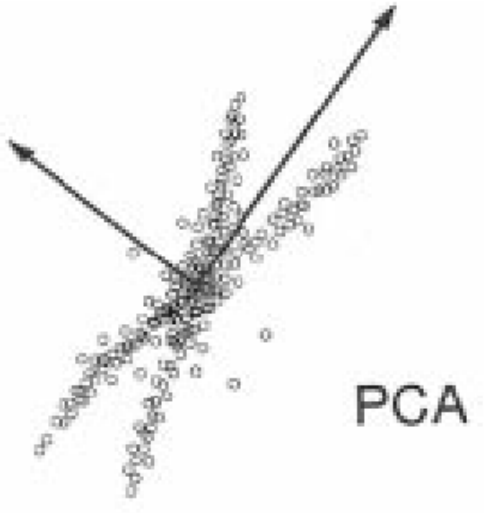
Gaussian iste Varijance kao distr. x

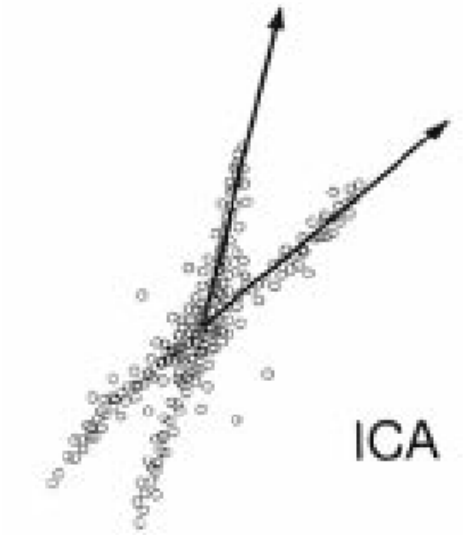
$\Rightarrow$ Koristi činjenicu da Gauss. ima maksim. entropiju (za dane μ, σ) !

- **Najčešće se koriste neke aproksimativne funkcije *G(z)* i na njima temelji određivanje *z***

$$J(z) = (E[G(z)] - E[G(z_{gaus})])^2$$

PCA vs ICA





- PCA: dekoreliranje varijabli

  (statistika drugog reda)

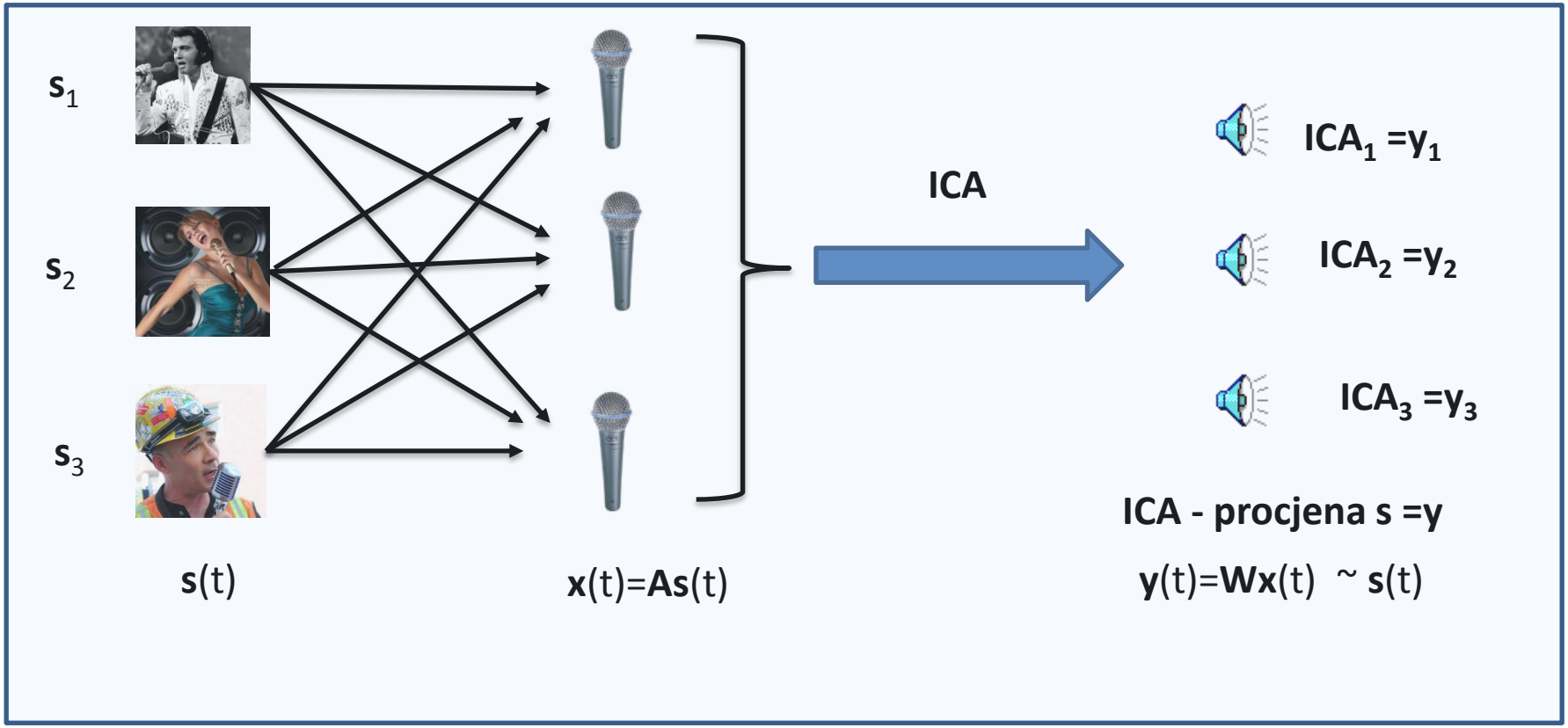
- ICA: nezavisnost

  (uključivanje momenata distribucije višeg reda)

Redukcija dimenzionalnosti: ICA

ICA = metoda kojom se rješava problem slijepe separacije signala
(**Blind Source Separation - BSS**)

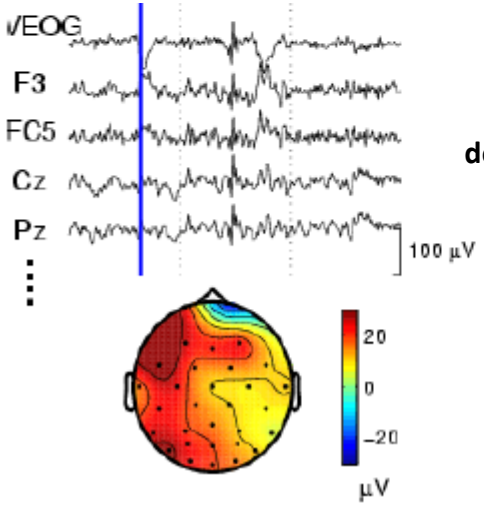
"cocktail party problem"



ICA – Slijepo razdvajanje signala

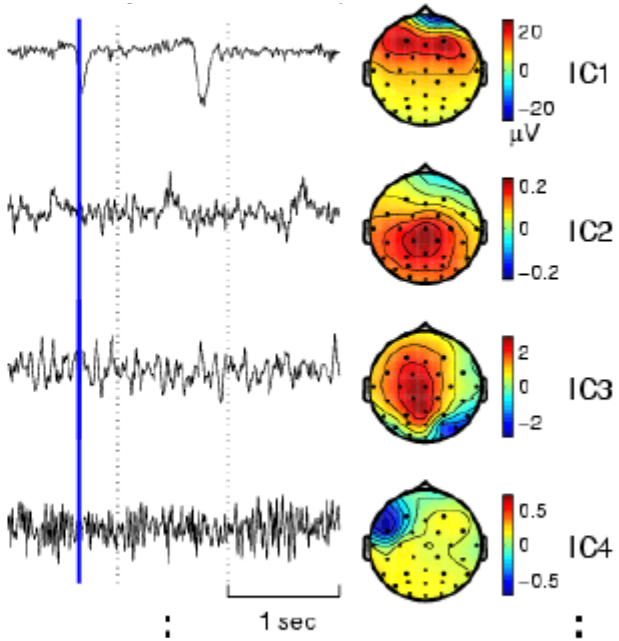
Primjer rješenja

EEG kanali (točke snimanja)



Signali x

ICA dekompozicija



Nezavisni izvori - aktivacije

y=Wx

Matrica miješanja

W^{-1}

Literatura

The Elements of Statistical Learning

Hastie, Tibshirani, Friedman (ch. 14)

PCA

- I. T. Jolliffe, *Principal Component Analysis*, Springer, 2nd, 2002.

ICA

- A. J. Bell and T. J. Sejnowski, *An information-maximisation approach to blind separation and blind deconvolution*, Neural Computation, 7(6), 1995.
- A. Hyvärinen and E. Oja, *Independent component analysis: a tutorial*, Neural Computation, 13(4-5), pp. 411-420, 2000

NMF

- D. D. Lee and H. S. Seung. *Learning the parts of objects by non-negative matrix factorization*. Nature 401, 788-791 (1999).
- D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, NIPS 13, pp. 556-562, 2001.