Strojno učenje

Tehnike strojnog učenja bez nadzora

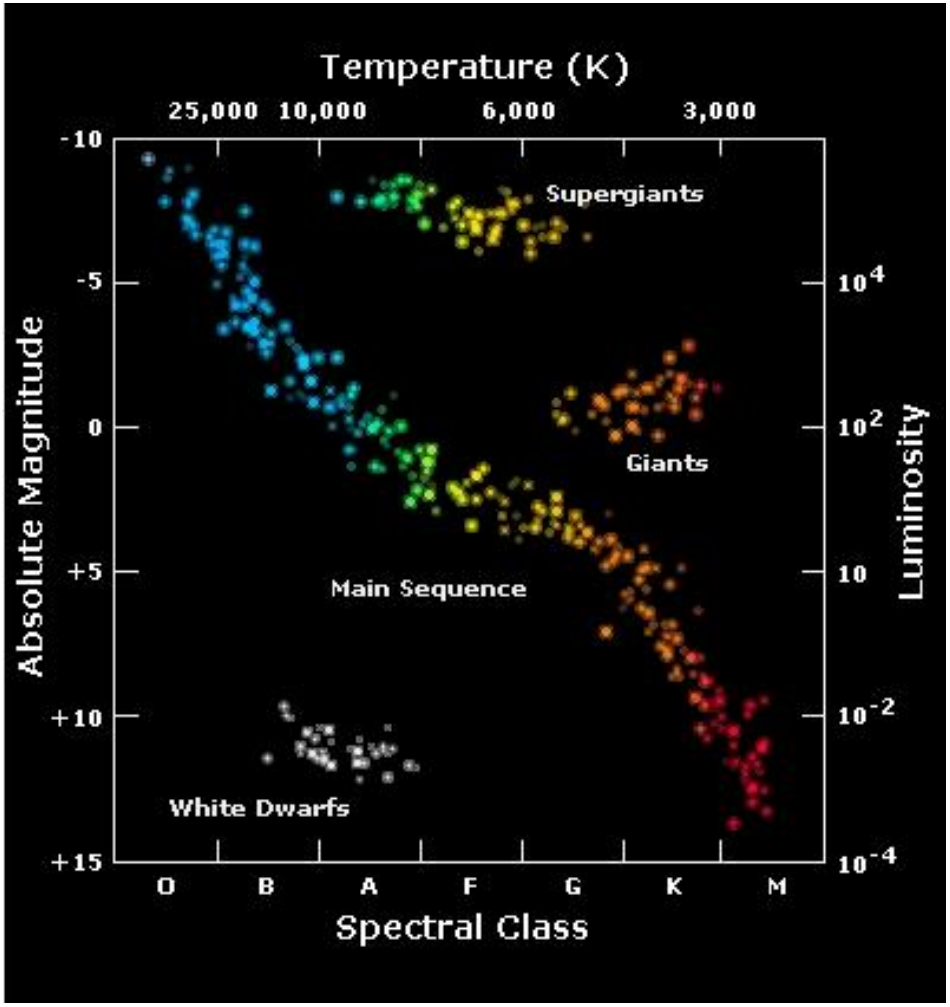
Tomislav Šmuc

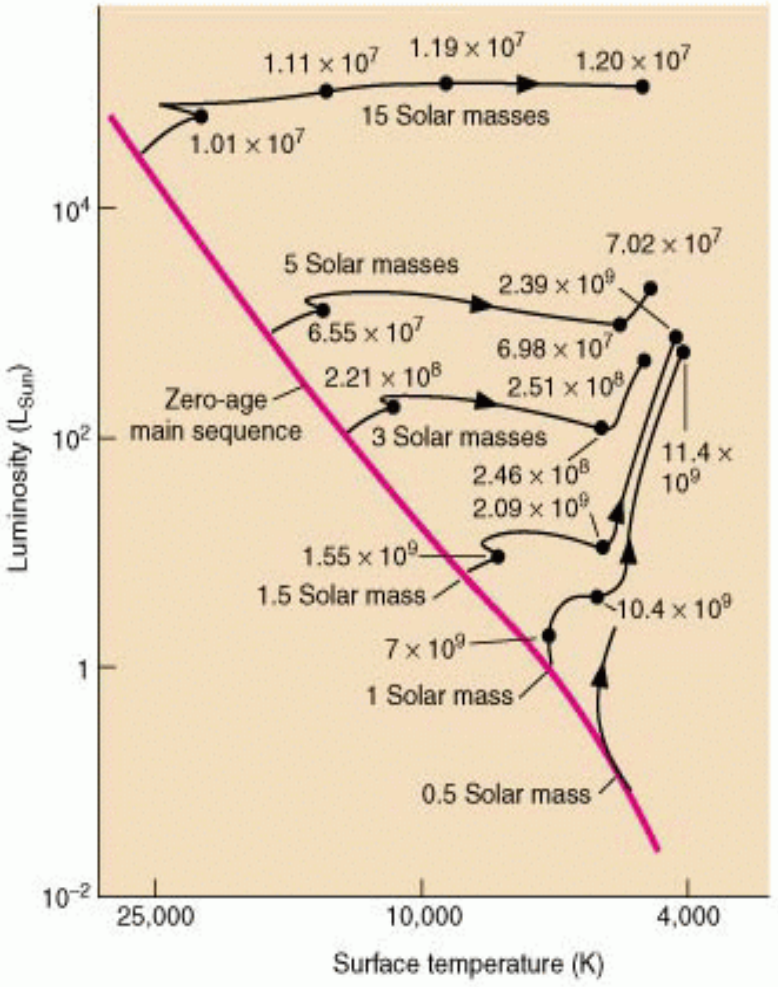
Literatura

- The Elements of Statistical Learning
  Hastie, Tibshirani, Friedman (ch. 14)

- WEKA Manual (Clustering)

Primjer - HRD


Temperature (K)
25,000
10,000
6,000
3,000
-10
-5
0
+5
+10
+15
Absolute Magnitude
Supergiants
Giants
Main Sequence
White Dwarfs
10^4
10^2
10
10^{-2}
10^{-4}
Luminosity
O
B
A
F
G
K
M
Spectral Class


1.11×10^7
1.19×10^7
1.20×10^7
15 Solar masses
1.01×10^7
10^4
7.02×10^7
5 Solar masses
2.39×10^9
6.55×10^7
6.98×10^7
2.21×10^8
2.51×10^8
Zero-age
main sequence
3 Solar masses
11.4×10^9
2.46×10^8
2.09×10^9
10^2
1.55×10^9
1.5 Solar mass
10.4×10^9
7×10^9
1
1 Solar mass
0.5 Solar mass
10^{-2}
Luminosity (L_{Sun})
25,000
10,000
4,000
Surface temperature (K)

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), 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)

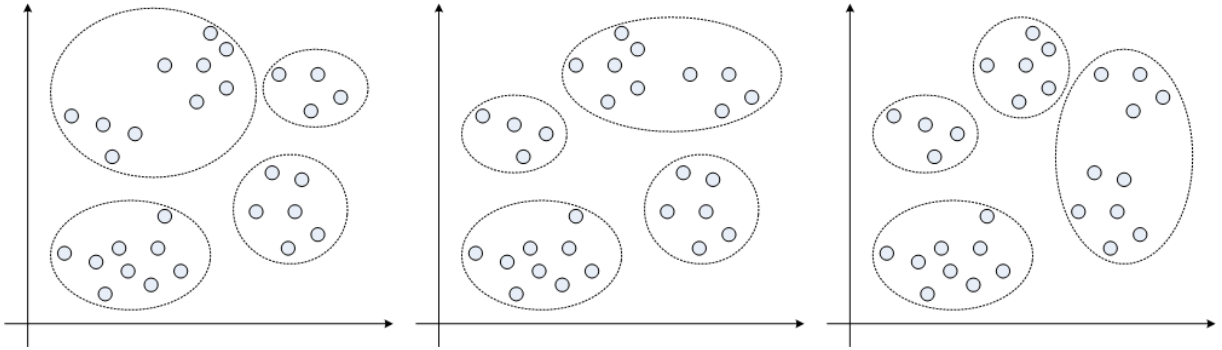
Osnovni problem:

- koliko cluster-a ima ?

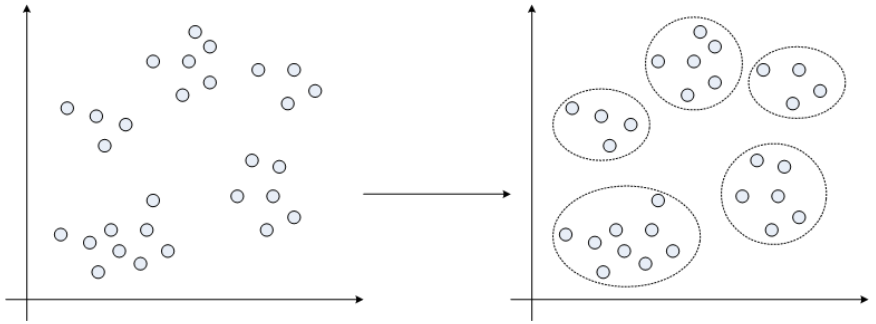
Osnovni problem:

- 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)
    - Aglomerativni i divizivni 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) :

1. **Izaberi K točaka:** c_j *(j=1,K)* – kao inicijalne centroide
2. **ponavljaj**

   2a) Za sve primjere:

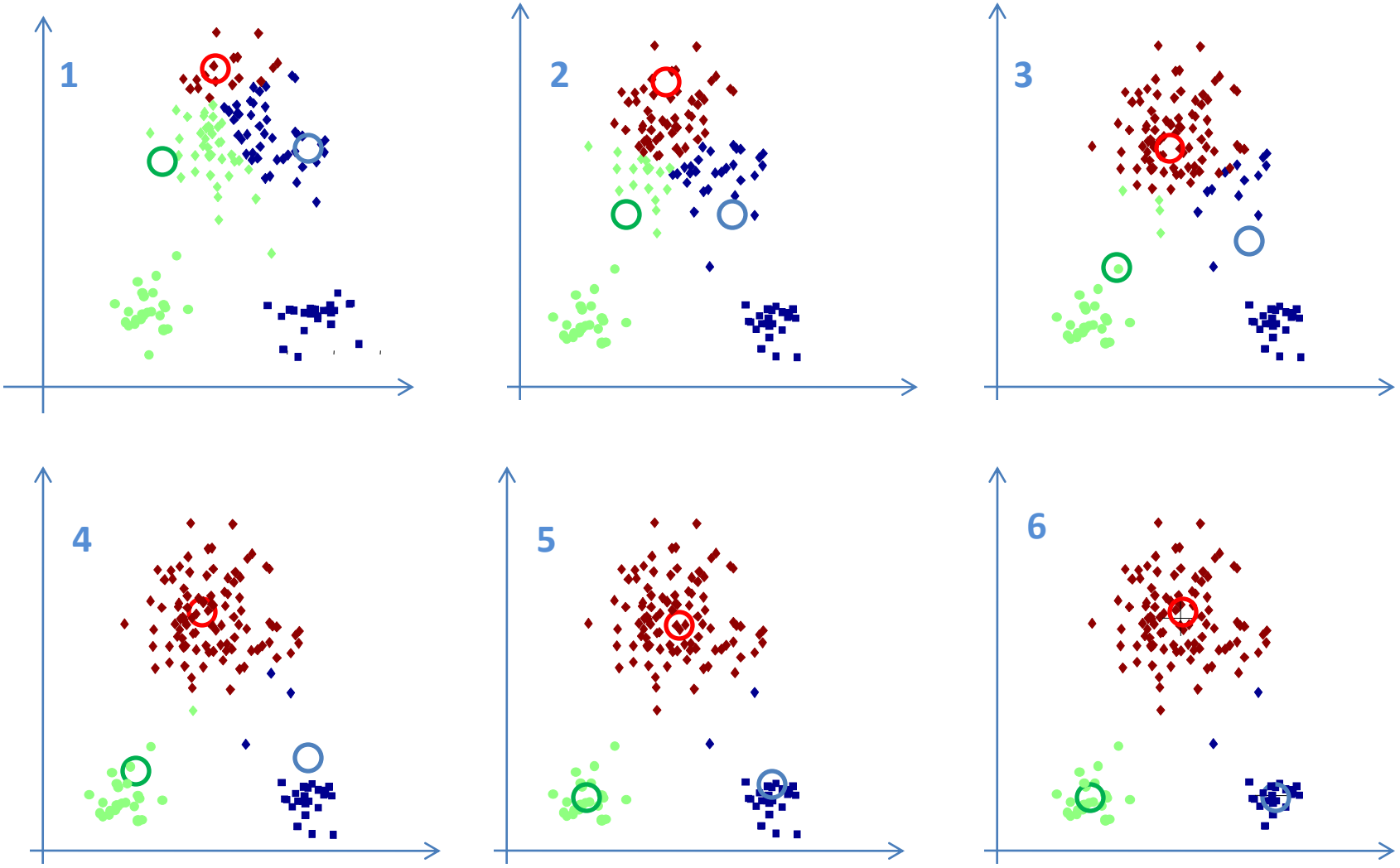
   pridjeli $C(x_i) = j$ – primjeru x_i ukoliko je $d(x_i, c_j)$ najmanja od svih c_j *(j=1,K)*

   2b) Izračunaj nove centroide c_j za svaki od cluster-a (srednja vrijednost svih primjera istog clustera)

4. **dok se c_j ne prestanu mijenjati**

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) !?
- 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

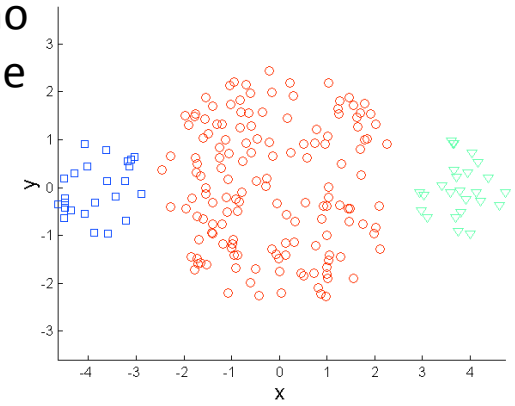
$$DBI = \frac{1}{k}.\sum_{i=1}^{k} R_i$$

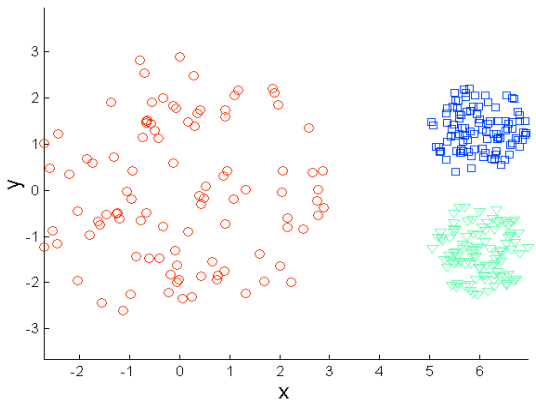
Minimalni DBI => optimalan K;
DBI – usporedba clustering metoda

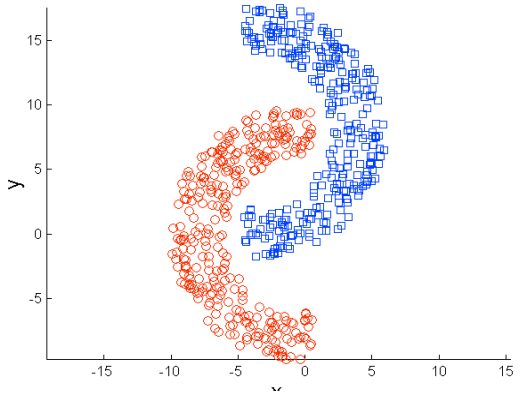
Problemi i ograničenja K-means algoritma: različite veličine

različite veličine | različite gustoće | nekonveksan oblik

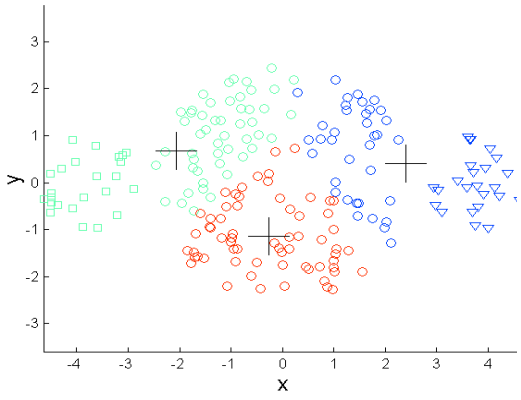
Originalno grupiranje

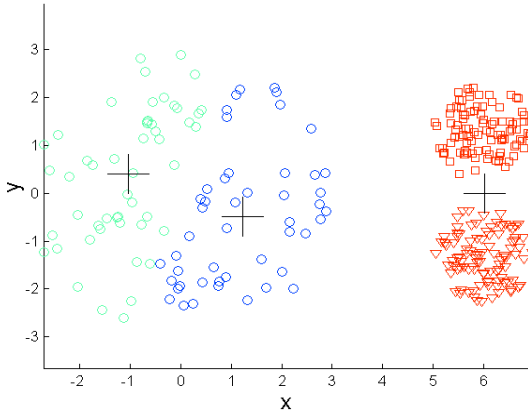


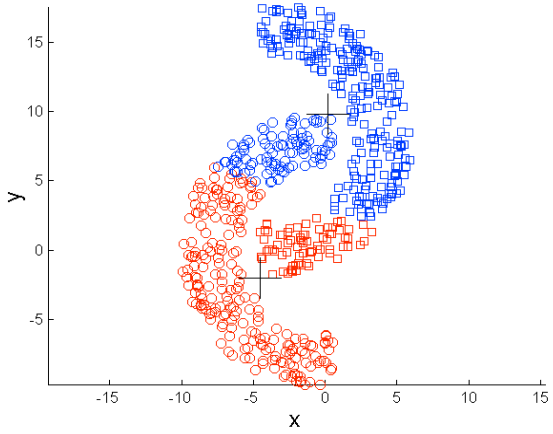




K-means (K=3)





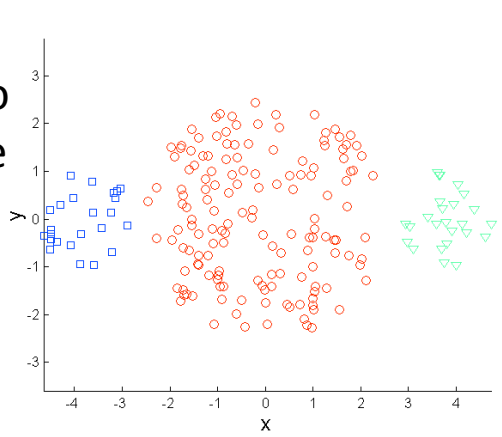


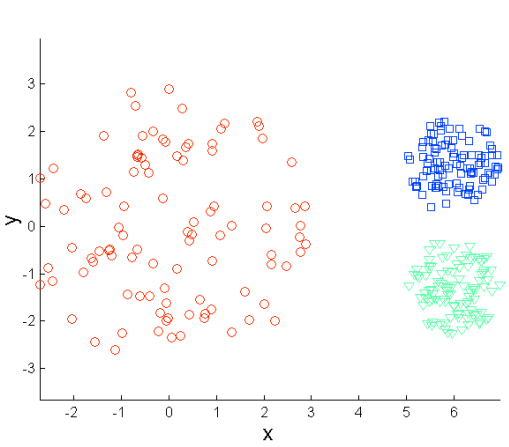
Clustering – K means

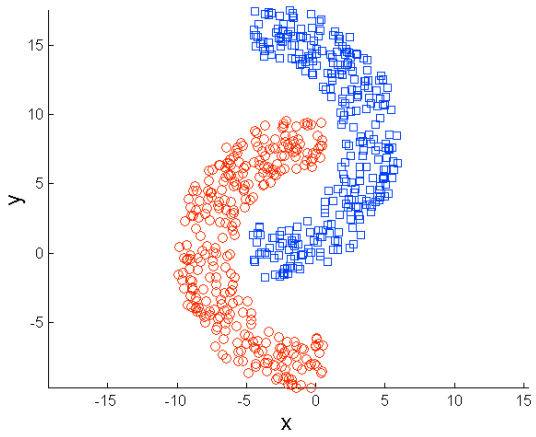
Tipično rješenje: veći K

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

Originalno grupiranje

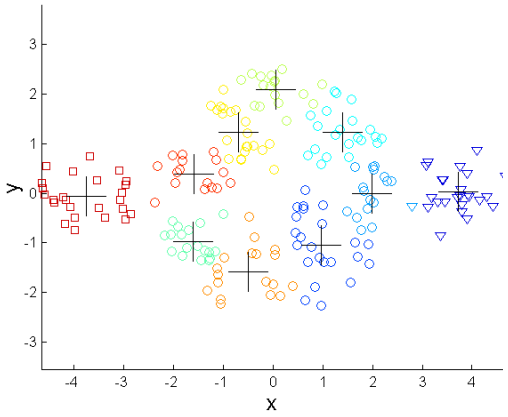


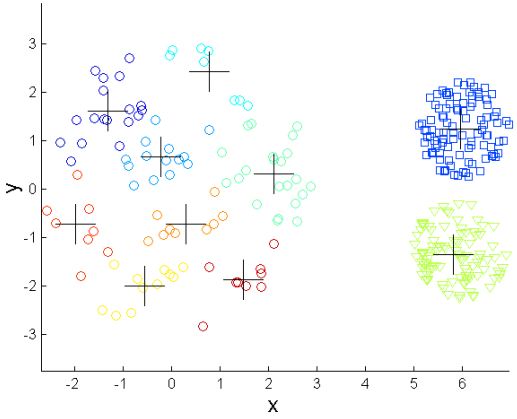


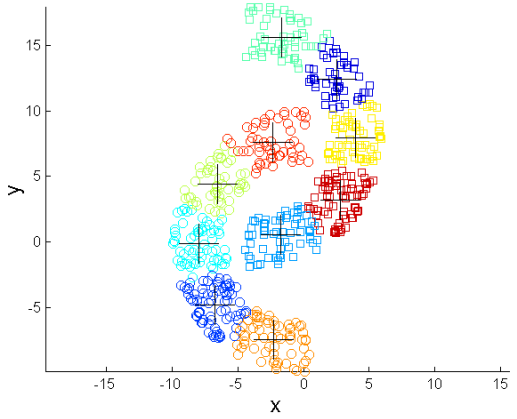


K-means

(K=10)

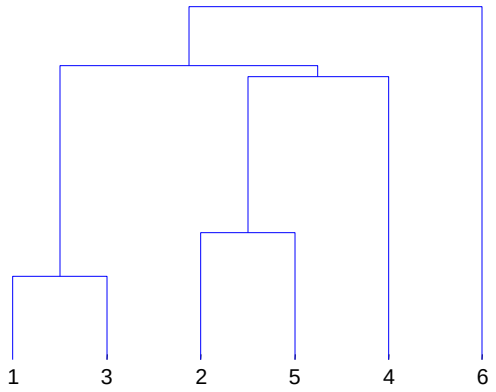


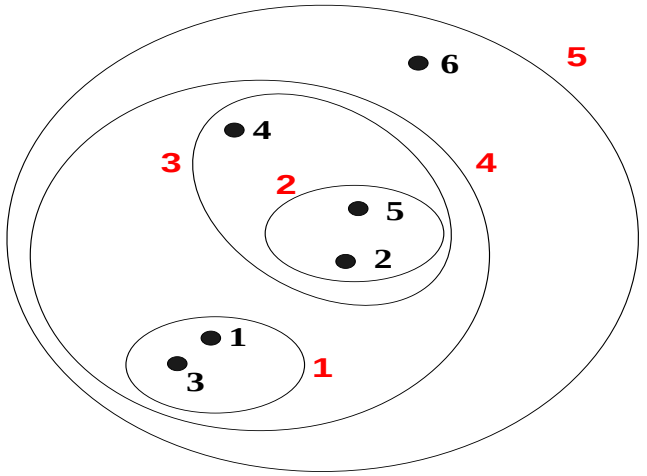




Hijerarhijski clustering

- 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

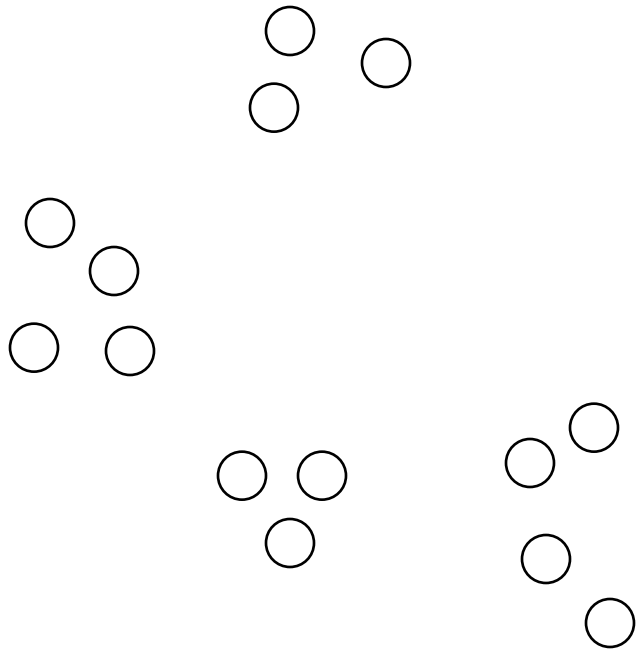
Aglomerativni HC algoritam

- **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

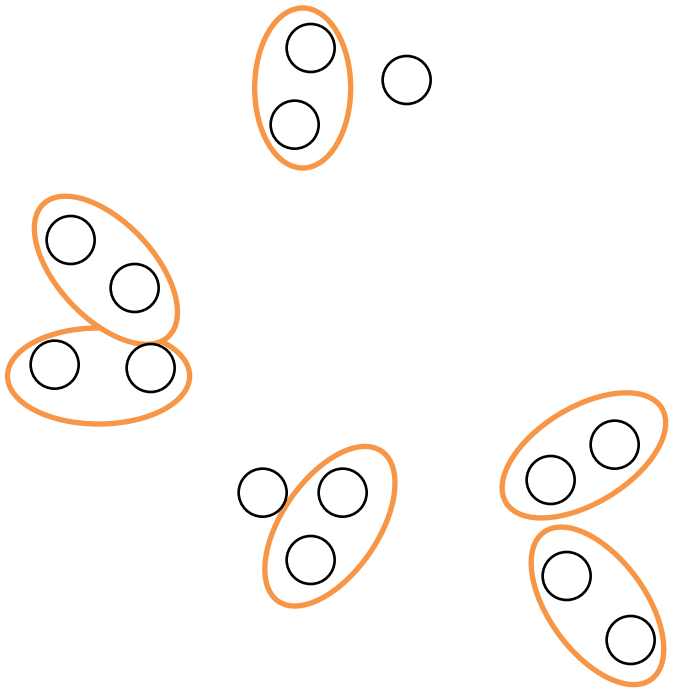


	p1	p2	p3	-...	pn
p1	0	1.1	2.1	...	
P2	1.1	0	3.2	...	
P3	2.1	3.2	0	...	
...	...	...			

Matrica udaljenosti

Aglomerativni HC algoritam

- i. korak

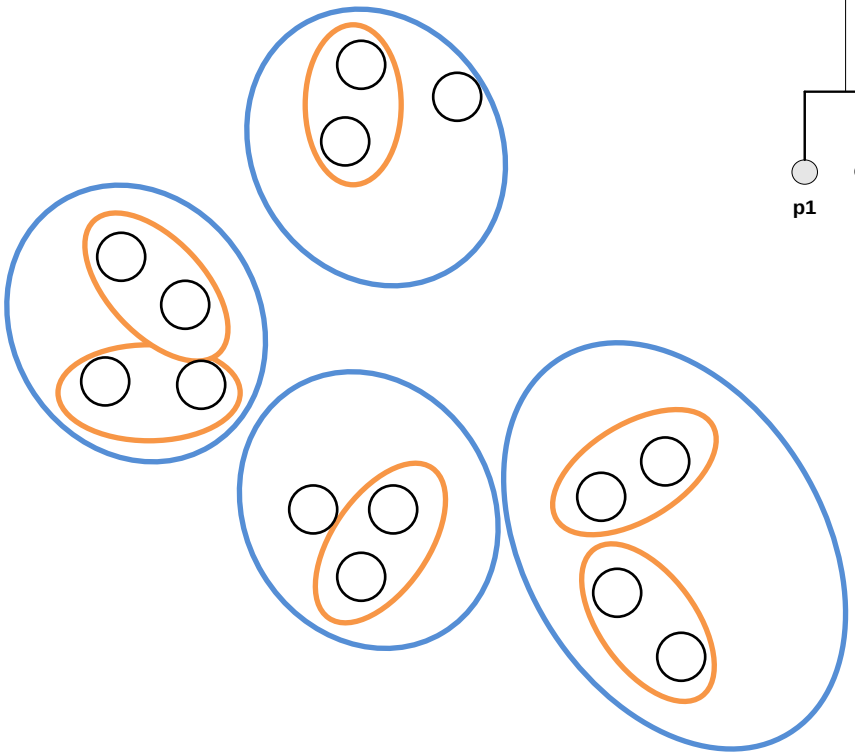


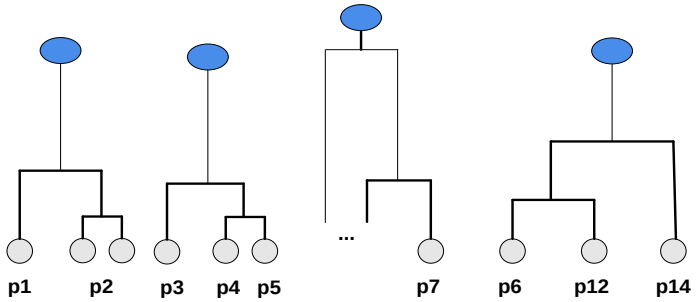
	c1	c2		-...	cj
C1	0	1.1		...	
c3	2.1	3.2		...	
...	...	...			

Matrica udaljenosti

Aglomerativni HC algoritam

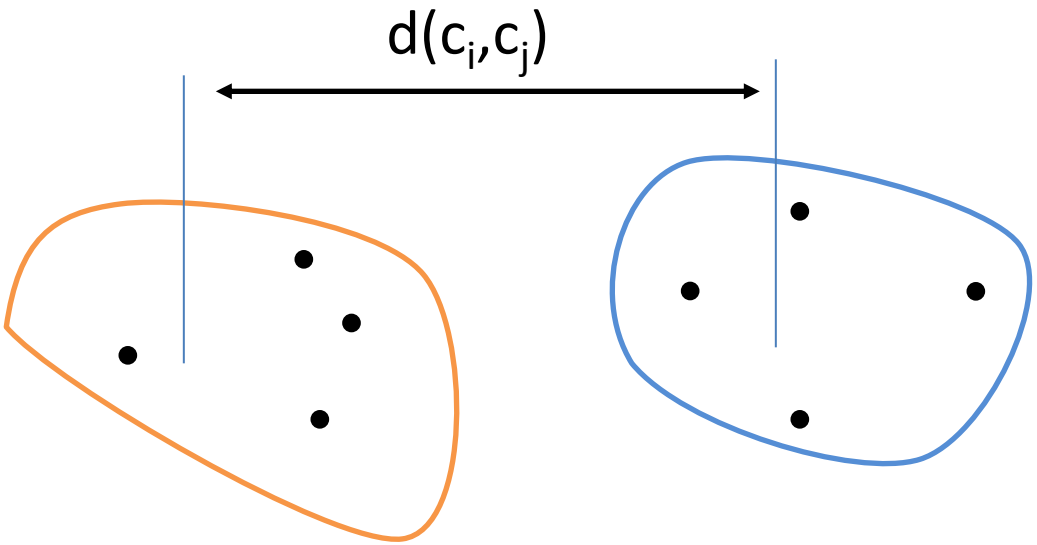
- Zadnji korak (K=4)





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

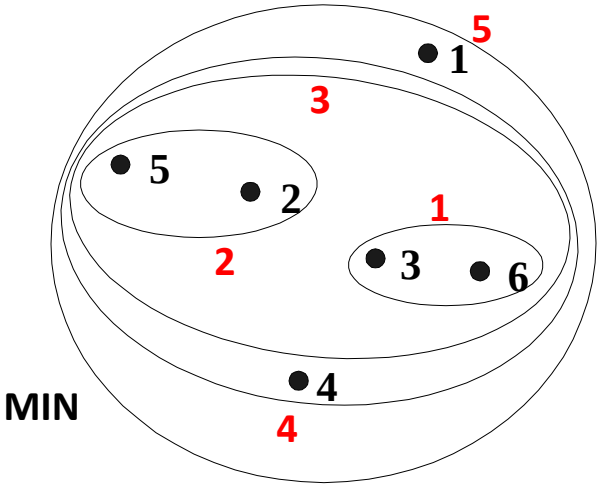
- Zavisno o odabranoj metodi dobit ćemo različiti rezultat !

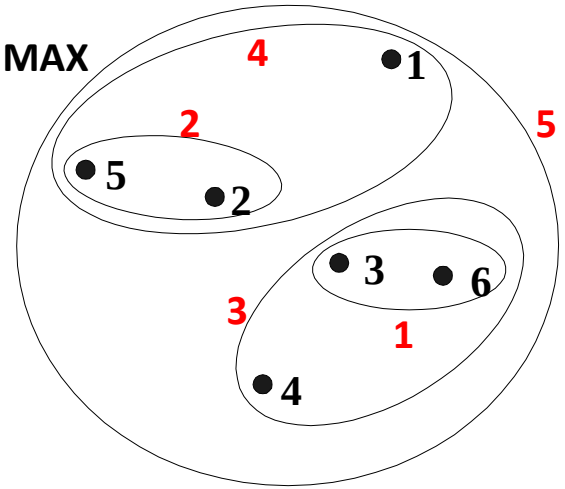
- **MIN $d(x_i,x_j)$**
  - dobro: dobro aproksimira eliptične oblike cluster-a
  - loše: osjetljiva na šum i "outlier" točke

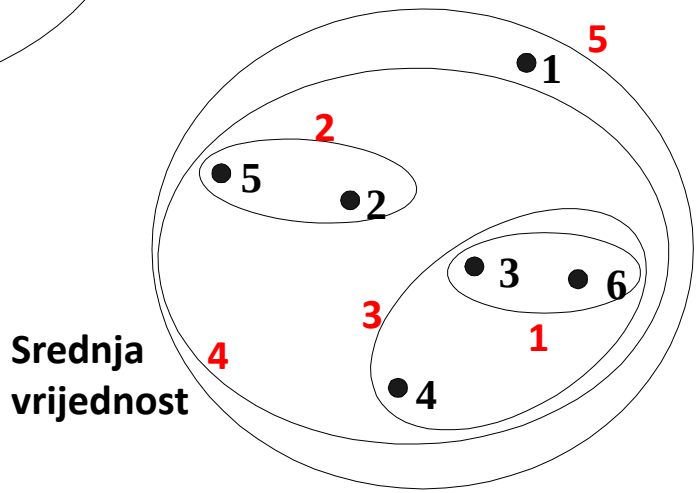
- **MAX $d(x_i,x_j)$**
  - dobro: manje osjetljiva na šum i outlier-e
  - loše: - "sklona" mrvljenju većih cluster-a
    - "sklona" stvaranju globularnih cluster-a

- **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

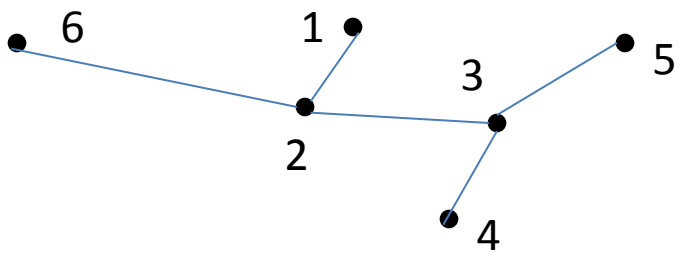
- (N= br primjera – N^2 matrica udaljenosti/sličnosti)

$O(N^3)$ – 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)

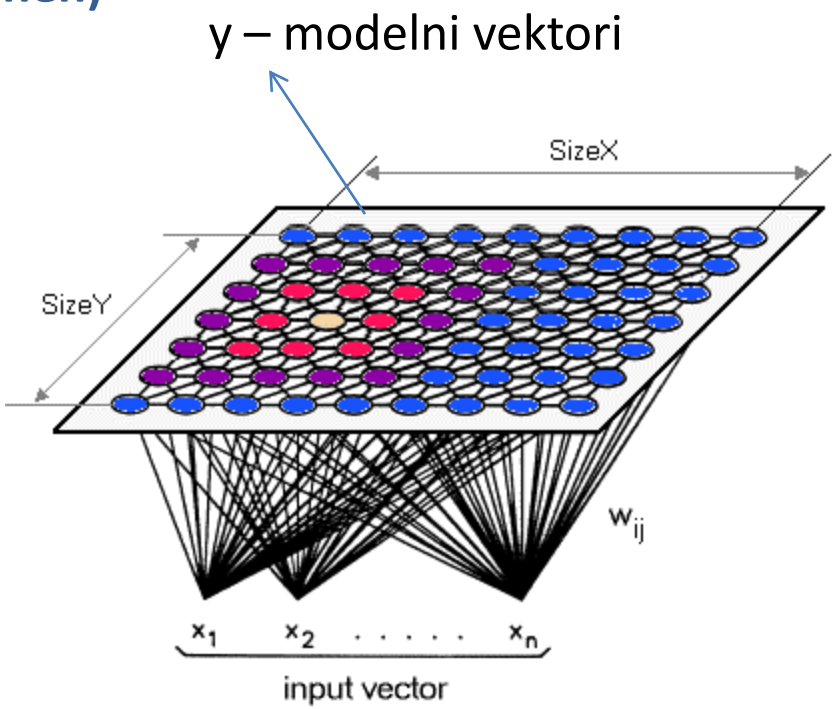
- Cilj: topologija primjera => mapiranje primjera u niže-dimenzionalni prostor, uz uvjet da udaljenosti između primjera budu što je više moguće sačuvane

- 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 $0 < \eta(t) \leq \eta(t-1) \leq 1$ 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

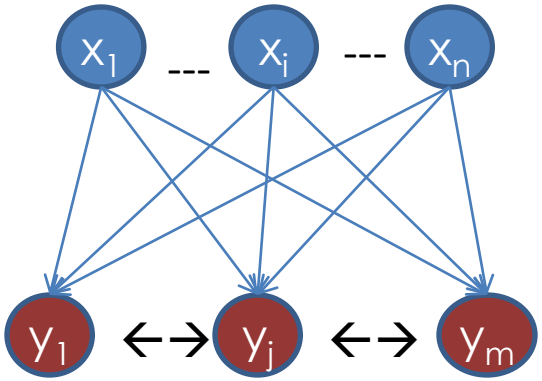
$$\sum_{k=1}^{n} (x_{i,k} - y_{j,k})^2$$

  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:

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

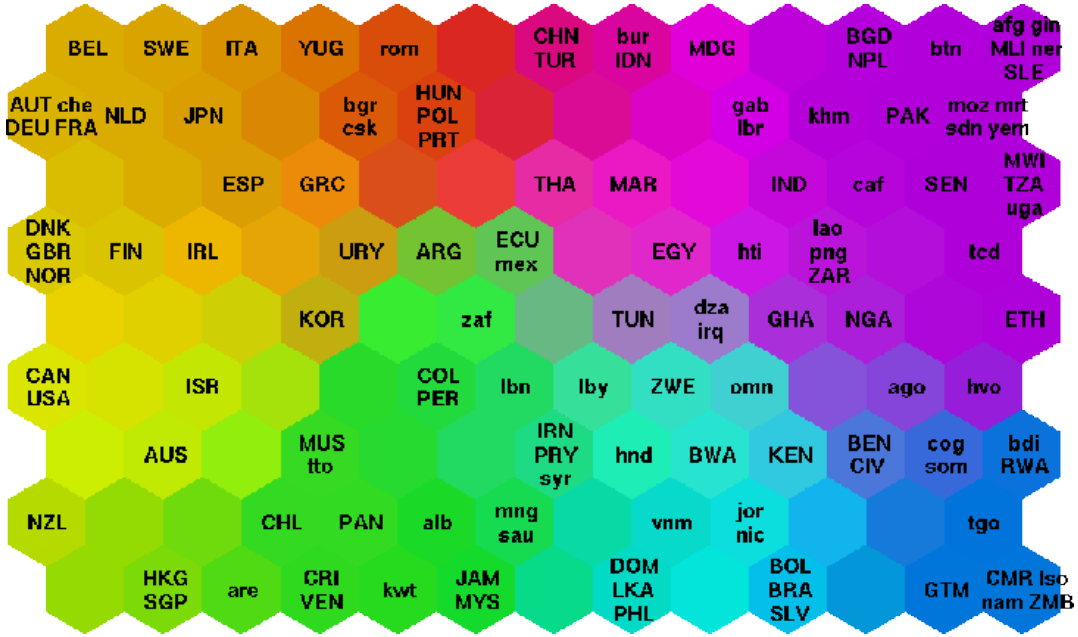
- **povećaj *t***

Arhitektura



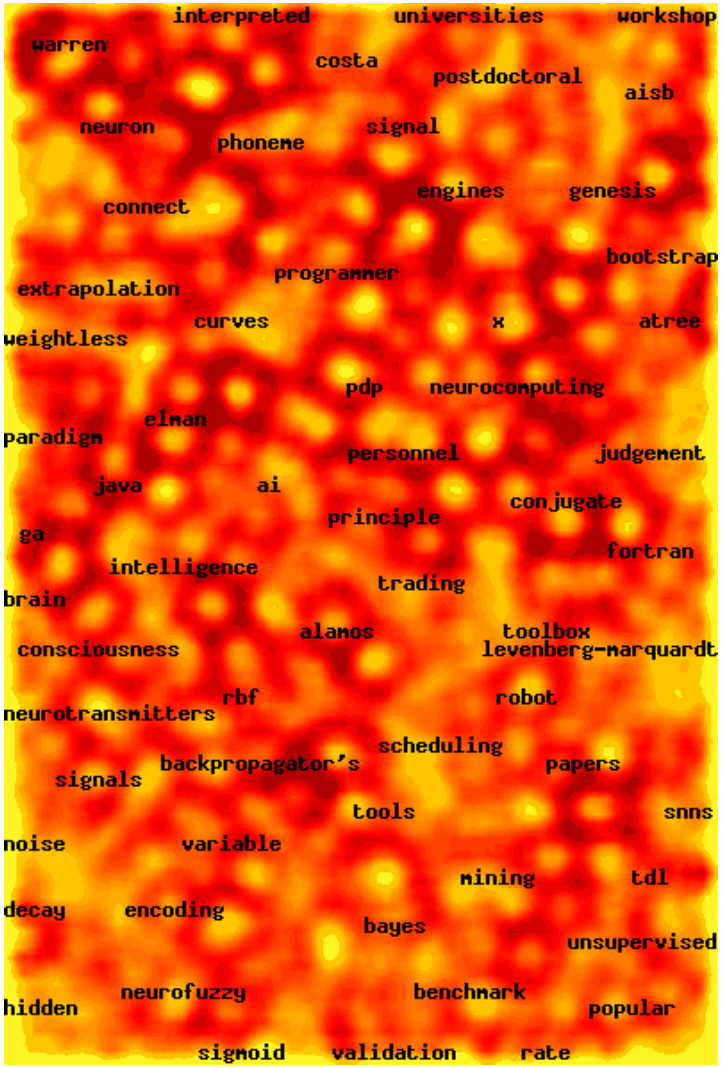
SOM – značaj i primjene

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



SOM:
World poverty map

SOM (Self-organizing-maps)



SOM: comp.ai.neural-nets newsgroup

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

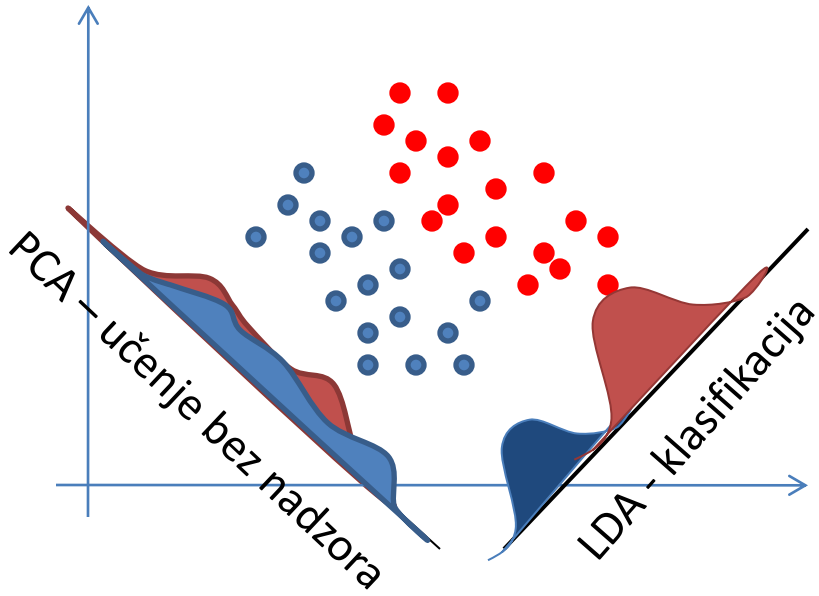
Ekstrakcija manjeg broja novih dimenzija-varijabli - kombiniranjem postojećih

- 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

PCA - Principal Component Analysis (metoda osnovnih komponenti)



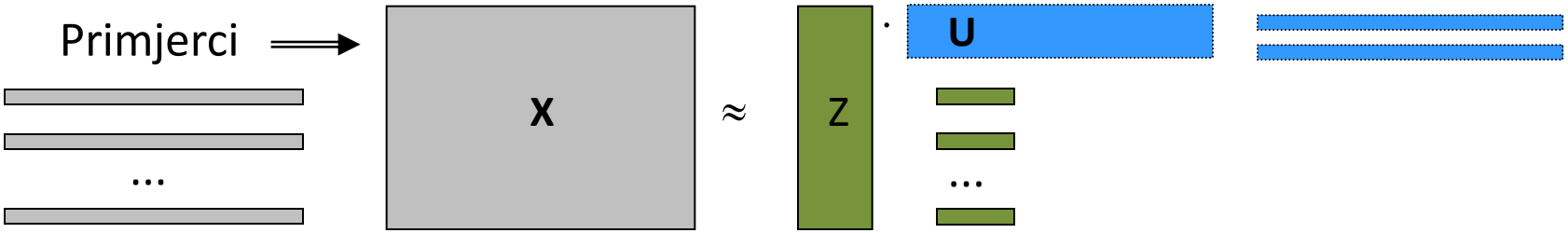
Projekcija zavisi od funkcije cilja koja se optimira

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

Transformacija u niže-dimenzionalnu reprezentaciju



- Određivanje svojstvenih vrijednosti/vektora
  (za $m \times m$ matricu **A**)

$$\mathbf{Av} = \lambda \mathbf{v}$$

$$\mathbf{v} \in \mathbf{R}^m \neq \mathbf{0} \text{ - svojstveni vektor}$$

$$\lambda \in \mathrm{R} \text{ - svojstvena vrijednost}$$

$$\mathbf{Av} = \lambda \mathbf{v} \Leftrightarrow (\mathbf{A} - \lambda \mathbf{I})\mathbf{v} = \mathbf{0}$$

$$\left| \mathbf{A} - \lambda \mathbf{I} \right| = 0$$

- Maksimalno *m* različitih rješenja

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:

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

PCA - Principal Component Analysis (metoda osnovnih komponenti)

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

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

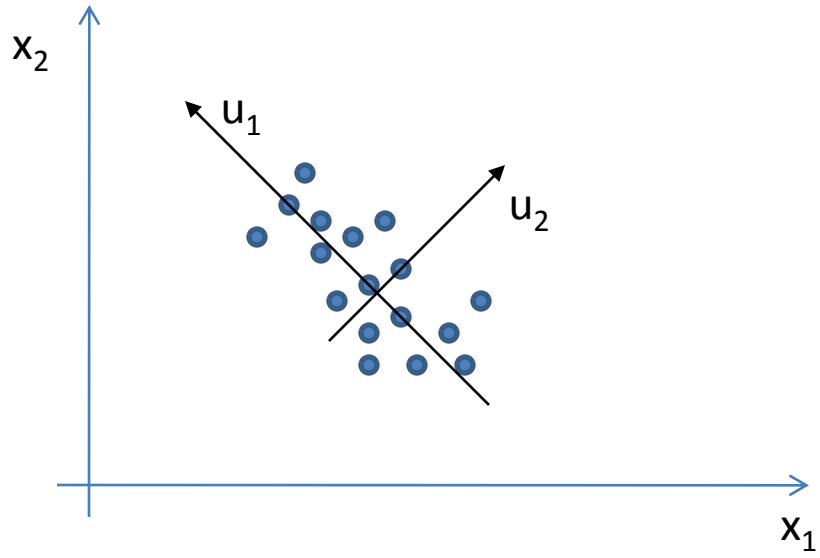
Gdje je **C** matrica kovarijance

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

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:

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

PCA - Principal Component Analysis (metoda osnovnih komponenti)



Projekcija $\mathbf{x}^k$ na u_1 i u_2 daje komponente transformiranog vektora $\mathbf{z}^k$

Što dobijamo ovakvom projekcijom ? Koje su primjene PCA ?

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)...

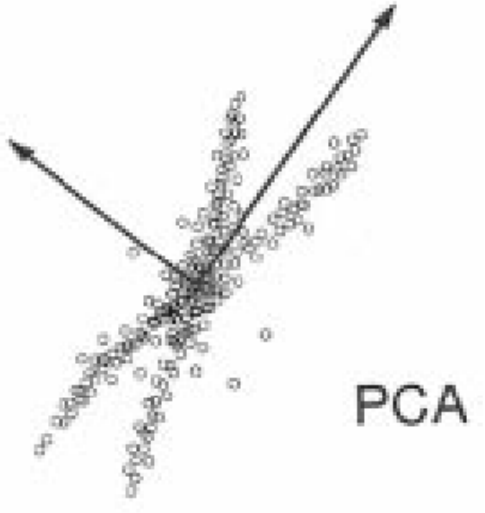
Dekoreliranost osnovnih komponenti nije i nezavisnost

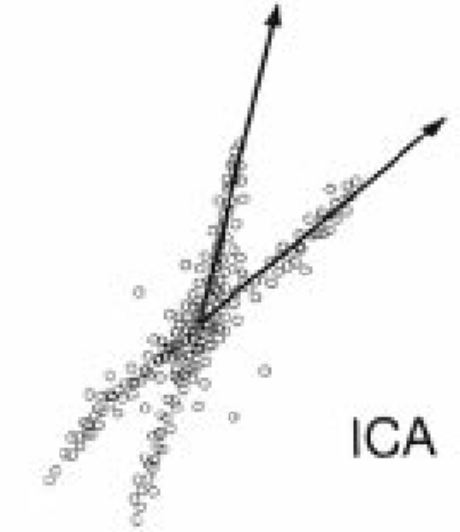
-> metoda nezavisnih komponenti (ICA)

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

-> NMF metoda (nonnegative matrix factorization)

PCA vs ICA





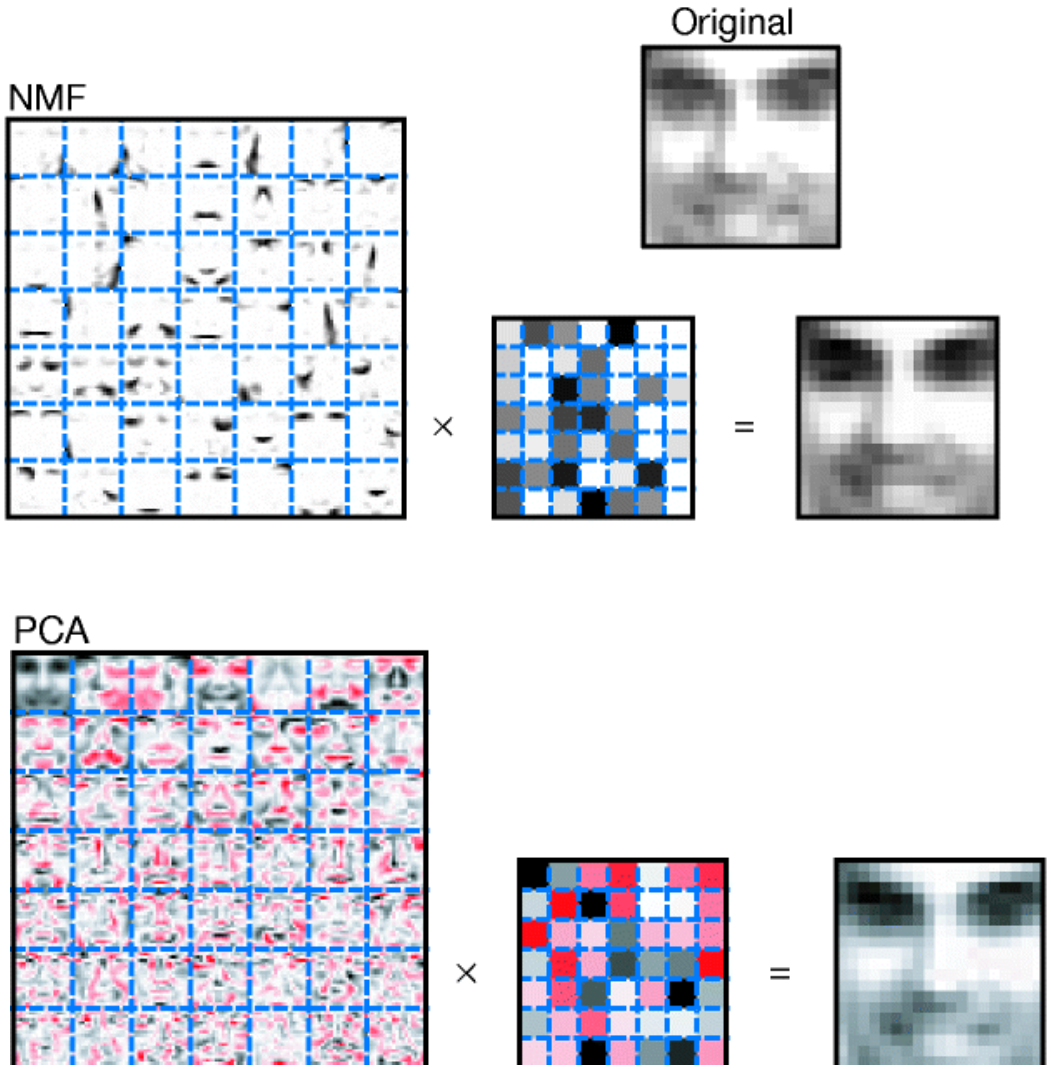
- PCA: dekoreliranje varijabli

  (statistika drugog reda)

- ICA: nezavisnost

  (uključivanje momenata distribucije višeg reda)

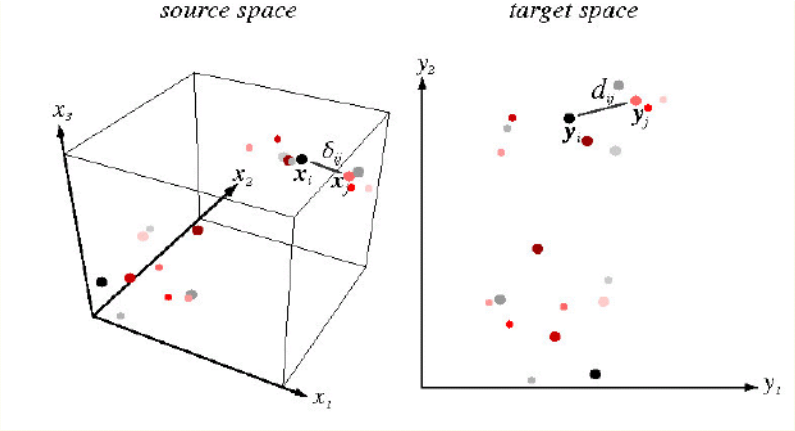
PCA vs NMF



Redukcija dimenzionalnosti: MDS

Višedimenzionalno skaliranje (Multi-dimensional scaling)

Pronalaženje konfiguracije točaka u niže dimenzionalnom prostoru čije međusobne udaljenosti korespondiraju sličnosti u višedimenzionalnom prostoru



Redukcija dimenzionalnosti: MDS

Višedimenzionalno skaliranje (Multi-dimensional scaling)

Zadano:

- $\mathbf{x_1}$,..., $\mathbf{x_n}$ u *p* dimenzija
- udaljenost d_{ij} između $\mathbf{x_i}$ i $\mathbf{x_j}$

Pronađi:

- $\mathbf{y_1}$,..., $\mathbf{y_n}$ u *2-3* dimenzije tako da je udaljenost između $\mathbf{y_i}$ i $\mathbf{y_j}$ **bliska** d_{ij}