České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra kybernetiky
Katedra počítačů

Vytěžování dat – přednáška 7

Shluková analýza

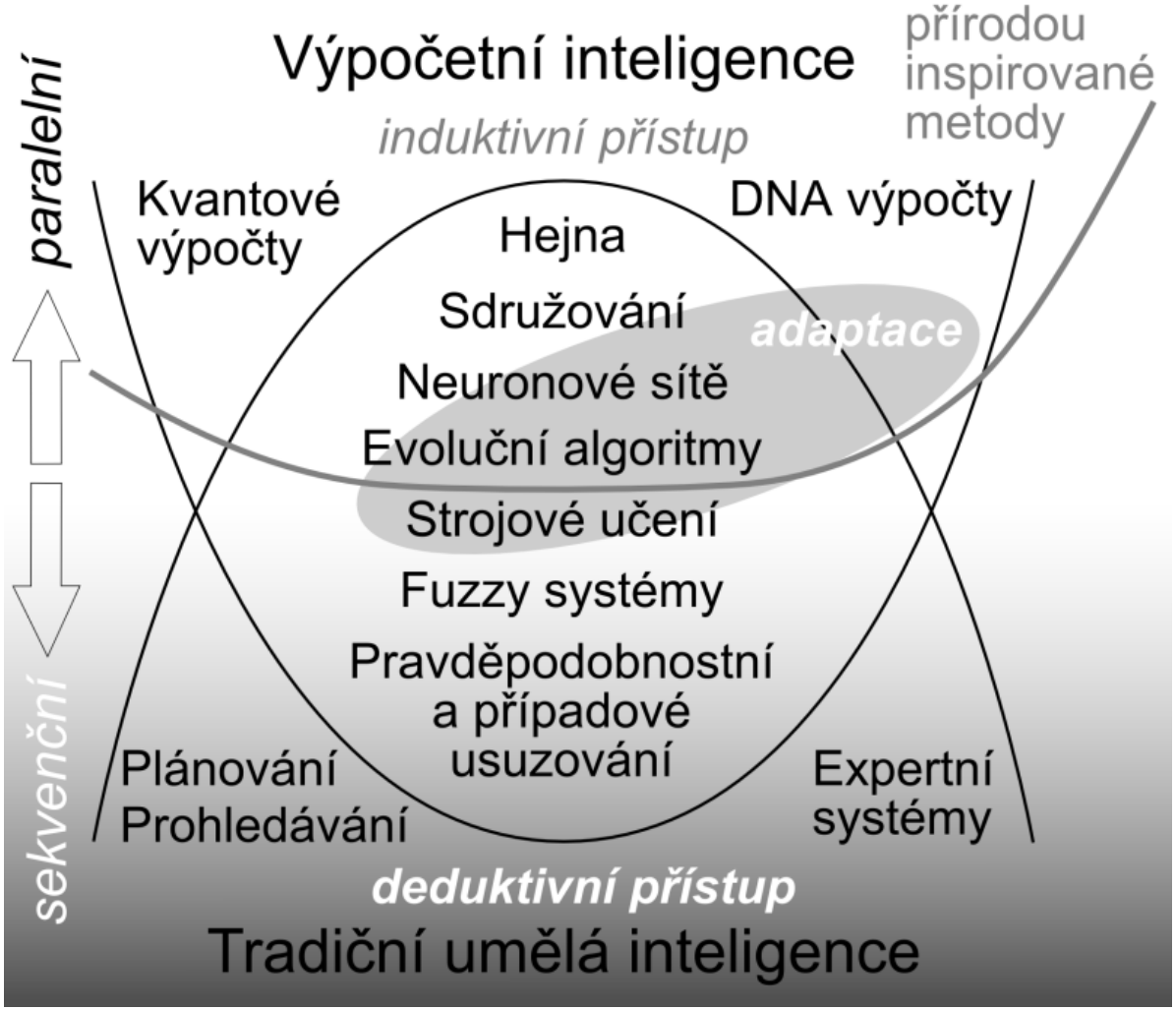
Organizační záležitosti

- Pavel Kordík … kordikp@fit, K428

- Výzkumná skupina Výpočetní inteligence (dříve Neuronové sítě)
  - http://cig.felk.cvut.cz
  - Lidé, projekty
  - FAKE GAME – někdy příště
    - http://sourceforge.net/projects/fakegame

Co to je ta „výpočetní …“

Algoritmy, které zatím neznáte, ale použijete je (nejen) pro vytěžování dat …



Organizace přednášek

- ANN v data miningu přesuneme na přespříští přednášku
- Dnes shluková analýza, kvůli druhé semestrální úloze
- Příště: Samoorganizující se mapa SOM

Osnova dnešní přednášky

- Metody shlukové analýzy
  - Metriky
  - Hiearchické shlukování
    - Algoritmy
    - Dendrogramy
  - K-means
  - Gaussovská směs

Shluková analýza

- Máme data, neznáme kategorie (třídy)
- Chceme najít množiny podobných vzorů, které jsou zároveň nepodobné vzorům z ostatních množin.
- Řešíme optimalizační problém!
- Co jsou naše neznámé?
  - počet shluků
  - přiřazení dat (vzorů) do shluků

Metrika, Euklidovská vzdálenost

- Je třeba nějak určit podobnost vzorů – jejich vzdálenost
- Vzdálenost musí splňovat určité podmínky:
  1. $d(x,y) \geq 0$.
  2. $d(x,y) = 0$ iff $x = y$.
  3. $d(x,y) = d(y,x)$.
  4. $d(x,y) \leq d(x,z) + d(z,y)$ (*trojúhelníková nerovnost*).

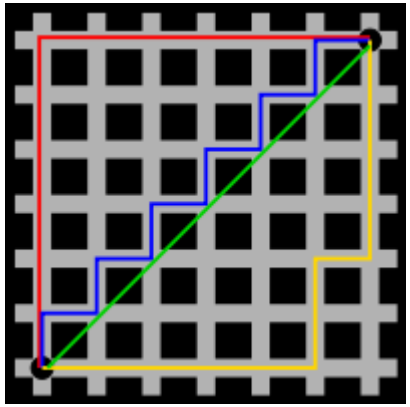
- Je Euklidovská vzdálenost metrika?

Dva body v n-rozměrném prostoru: $P = (p_1, p_2, \ldots, p_n) \quad Q = (q_1, q_2, \ldots, q_n)$

Euklidovská vzdálenost P a Q = $\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$.

Manhattonská vzdálenost

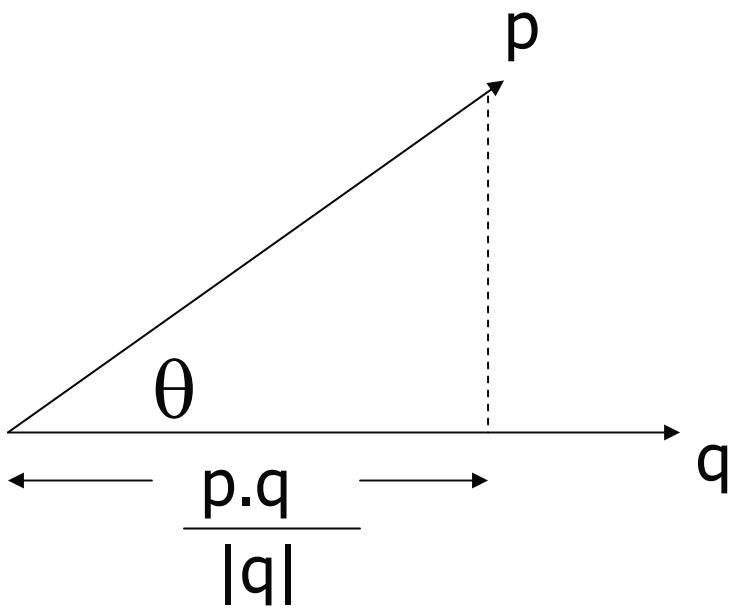
- Jak budeme počítat vzdálenost dvou cyklistů v Manhattonu?



$$M(P,Q) = |p_1 - q_1| + |p_2 - q_2| + \ldots + |p_n - q_n|$$

Kosinová vzdálenost

- Je invariantní vůči natočení



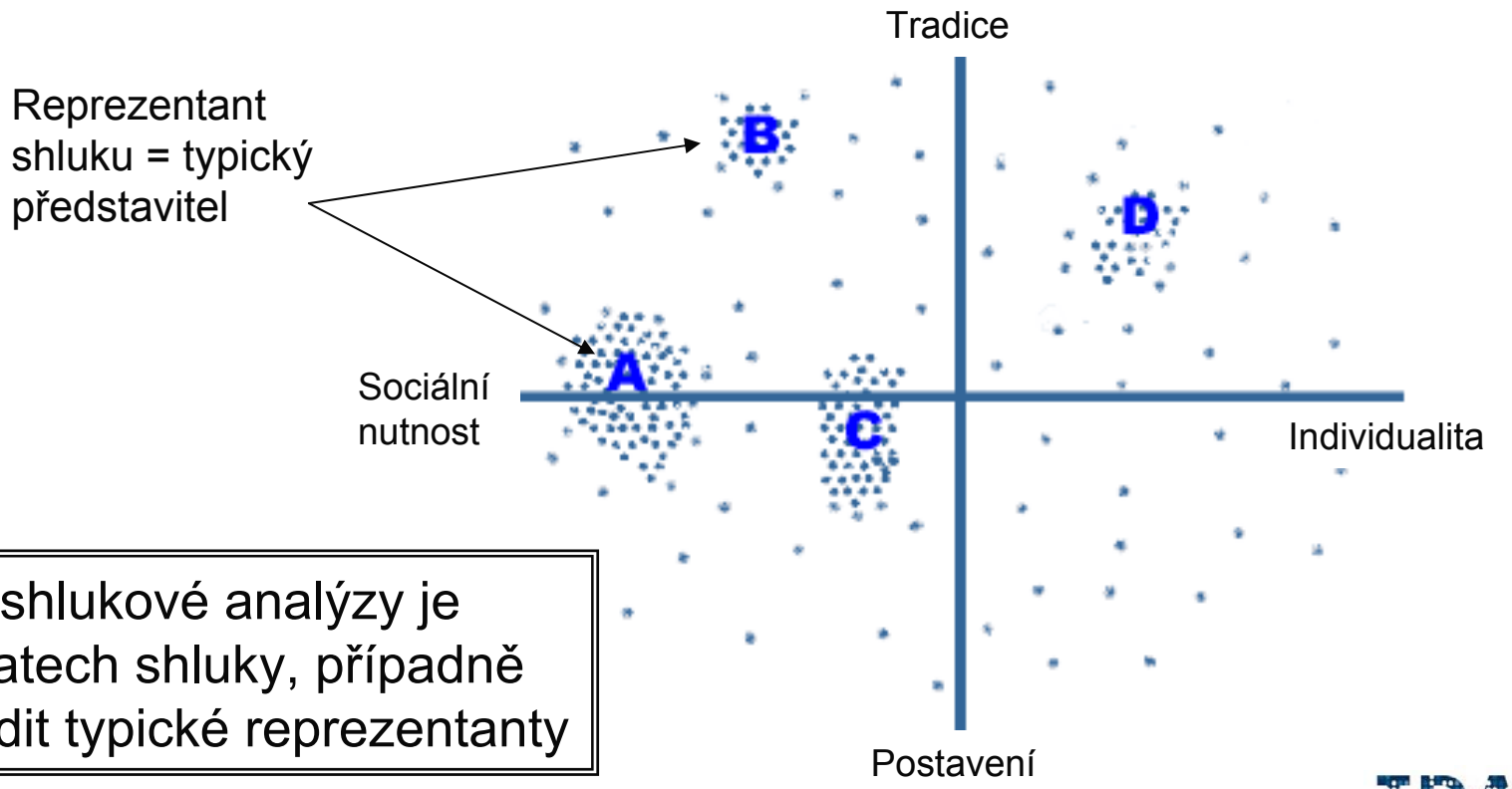
dist(p, q) = θ = arccos(p.q/|q||p|)

Editační vzdálenost

- Pro určení vzdálenosti např. dvou slov
- Počítá se jako počet smazání (vložení) písmene, potřebný k transformaci jednoho slova na druhé.

Shluky, reprezentanti

- Výsledky ankety, proč lidé pijí alkohol



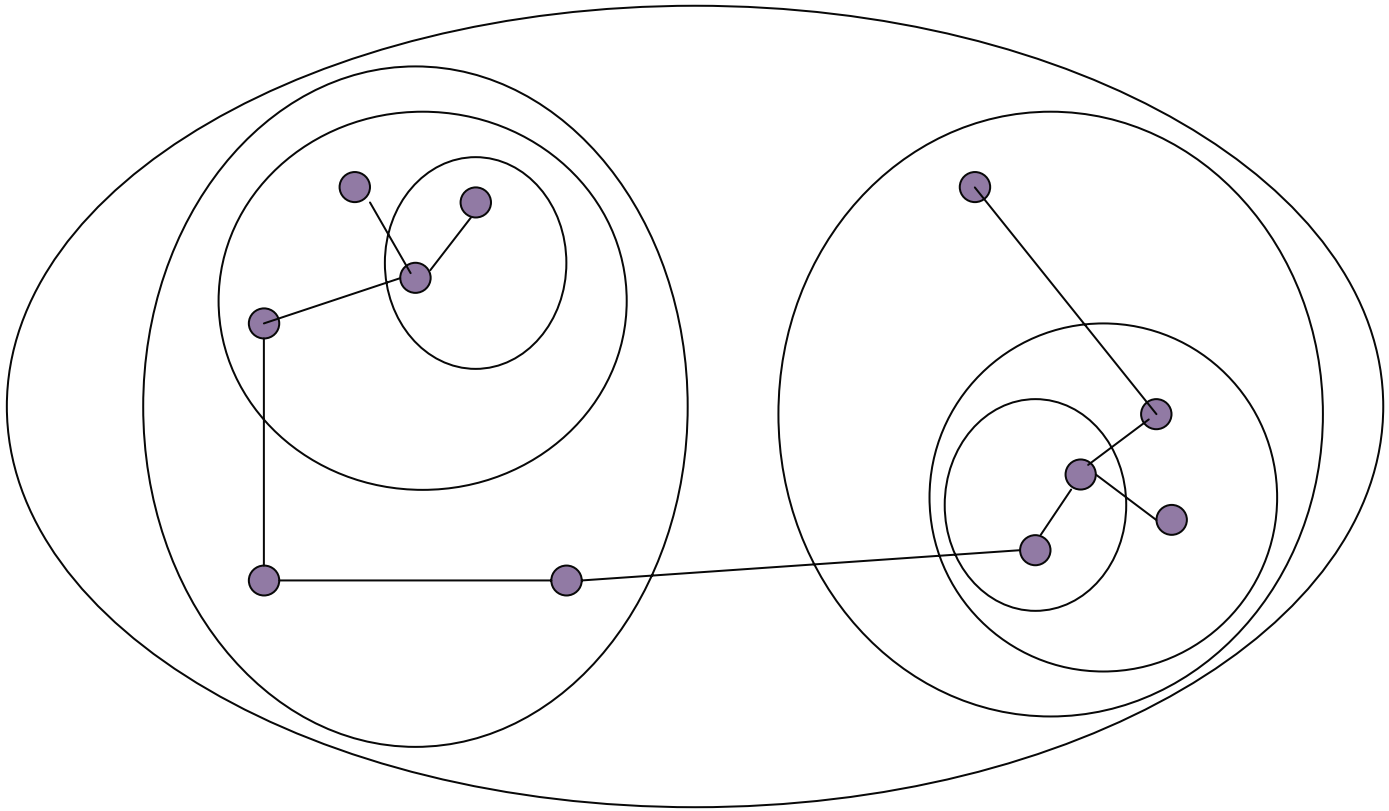
Úkolem shlukové analýzy je najít v datech shluky, případně jim přiřadit typické reprezentanty

Shluková analýza III

- Klasická **shluková analýza** (Cluster Analysis) je nástroj pro disjunktní rozklad množiny vzorů ze vstupního prostoru $\mathbf{R}^n$ do $H > 1$ tříd (shluků).
- Shluková analýza požaduje maximální podobnost vzorů v rámci jedné třídy a současně maximální nepodobnost vzorů různých tříd.

Jak byste problém řešili vy?

- Jak najít shluky?
- spojujeme vždy 2 nejpodobnější vektory

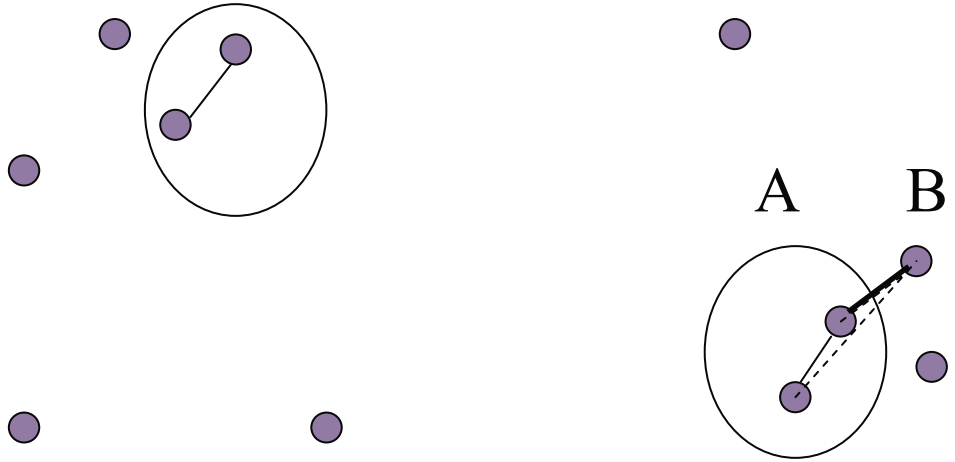


Metody vyhodnocení vzdálenosti shluků

- Metoda nejbližšího souseda – **single linkage** – vzdálenost shluků je určována vzdáleností dvou nejbližších objektů z různých shluků
- Metoda nejvzdálenějšího souseda – **complete linkage** – zdálenost shluků je určována naopak vzdáleností dvou nejvzdálenějších objektů z různých shluků
- Centroidní metoda – **centroid linkage** – vzdálenost shluků je určována vzdáleností jejich center
- Metoda průměrné vazby – **average linkage** – vzdálenost shluků je určována jako průměr vzdáleností všech párů objektů z různých shluků
- Wardova metoda – **Ward's linkage** – vzdálenost shluků se určí jako suma čtverců vzdáleností jejich center

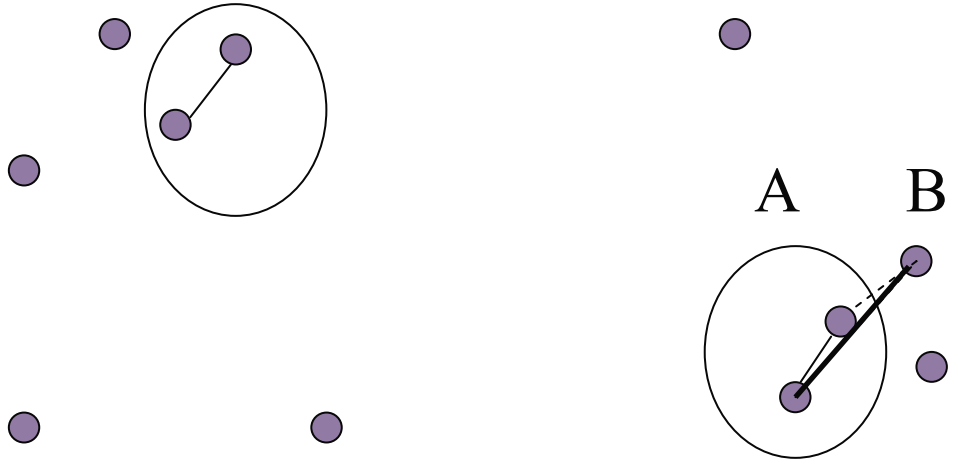
Single linkage

- Ze shluku vždy vybírám toho nejbližšího



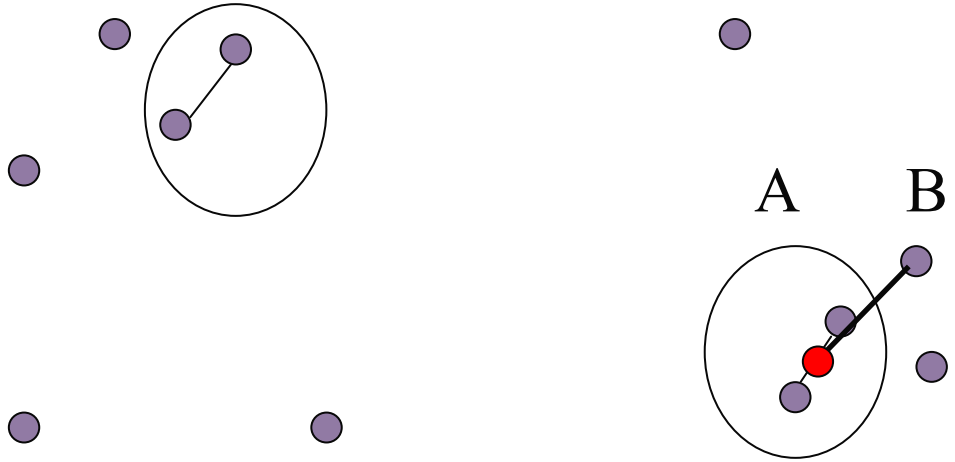
Complete linkage

- Ze shluku vždy vybírám toho nejvzdálenějšího



Centroid linkage

- Reprezentantem shluku je centroid



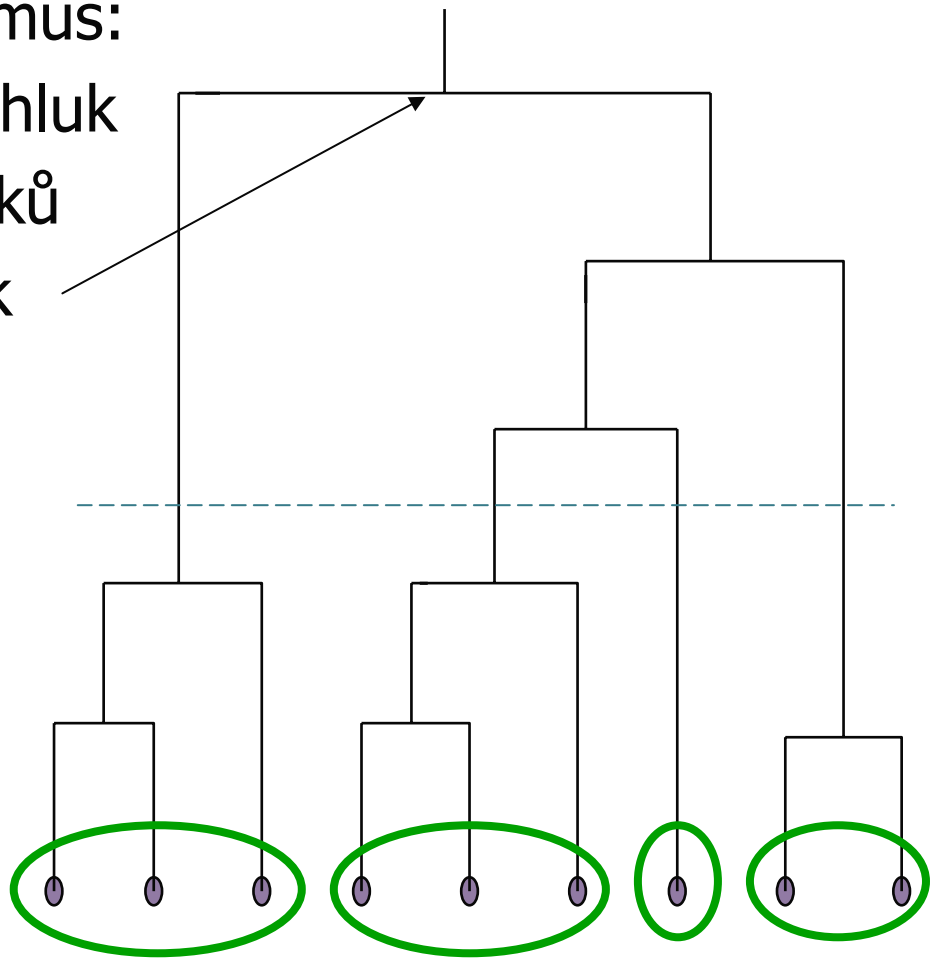
Kolik jsme našli shluků?

- Jiný pohled na náš algoritmus:
- Na začátku každý vektor shluk
- Spojování vektorů do shluků
- Na konci jeden velký shluk
- Počet shluků ?

- Dendrogram =>

- Algoritmus se jmenuje:

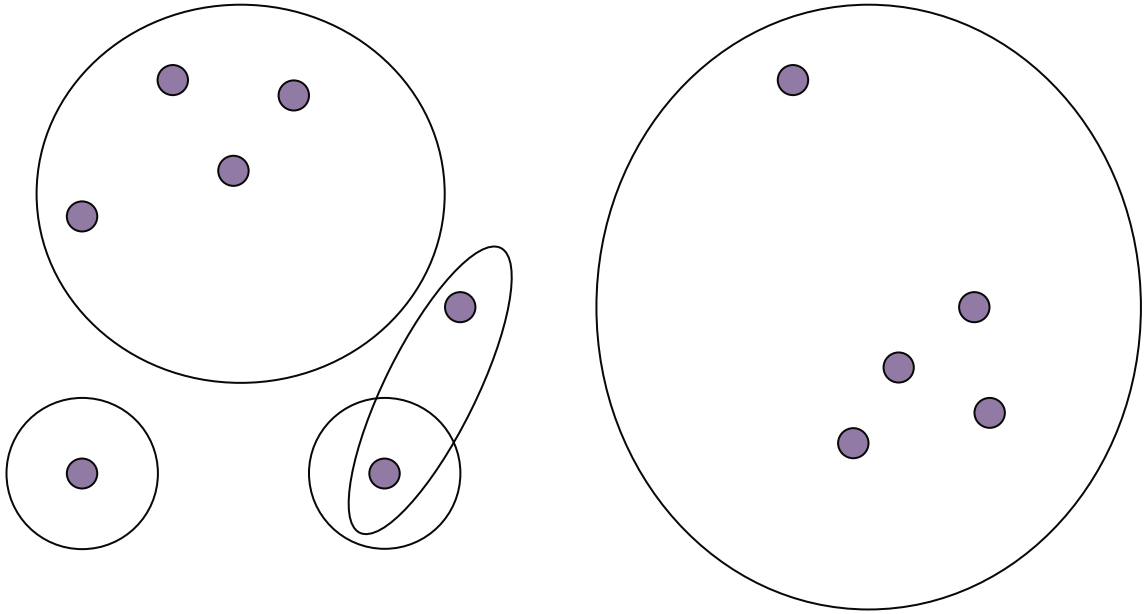
Hierarchické shlukování



Obrázek je ilustrační, neodpovídá přesně datům z minulého slajdu

Záleží na tom, kde „řízneme“ dendrogram

- Co se děje, když řežeme dendrogram na nižší/vyšší úrovni?
- Kam patří nový vektor?



- Problém? Musím počítat vzdálenosti ke všem vektorům!

Obsahují data opravdu shluky?

- Vypočteme CPCC (Cophenetic Correlation Coeffitient)
- CPCC je normovaná kovariance vzdáleností v původním prostoru a v dendrogramu
- Pokud je hodnota CPCC menší než cca 0.8, všechny instance patří do jediného velkého shluku
- Obecně platí, že čím vyšší je kofenetický koeficient korelace, tím nižší je ztráta informací, vznikající v procesu slučování objektů do shluků

Hierarchické shlukování

- Pseudokód algoritmu hierarchického shlukování
  - c je požadovaný počet shluků

```
1. begin initialize c, c'←n, Di←{xi} i=1,...,n
2.   do c'←c'+1
3.         vypočteme matici vzdáleností
4.         najdeme nejbližší shluky Di a Dj
5.         sloučíme shluky Di a Dj
6.   until c=c'
7.   return c shluků
8. end
```

- procedura skončí, když je dosaženo požadovaného počtu shluků
  - když c=1, dostaneme dendogram
- složitost
  - $O(cn^2d)$ a typicky n>>c

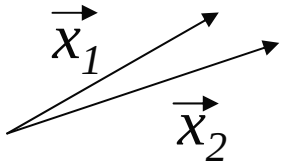
Algoritmus K-středů (K-means)

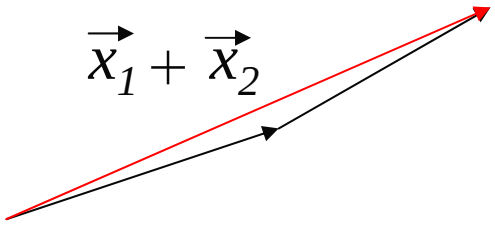
- Jak se vyhnout výpočtu všech vzájemných vzdáleností?
- Budu počítat vzdálenosti od reprezentantů shluků.
- Počet reprezentantů je výrazně menší než počet instancí.
- Nevýhoda: musím dopředu určit počet reprezentantů (K).

K-means

- Reprezentanti - zde se jmenují středy (centroidy)
- Střed shluku c vypočteme: $\vec{\mu}(c) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$

Co to znamená? Jak se sčítají vektory?

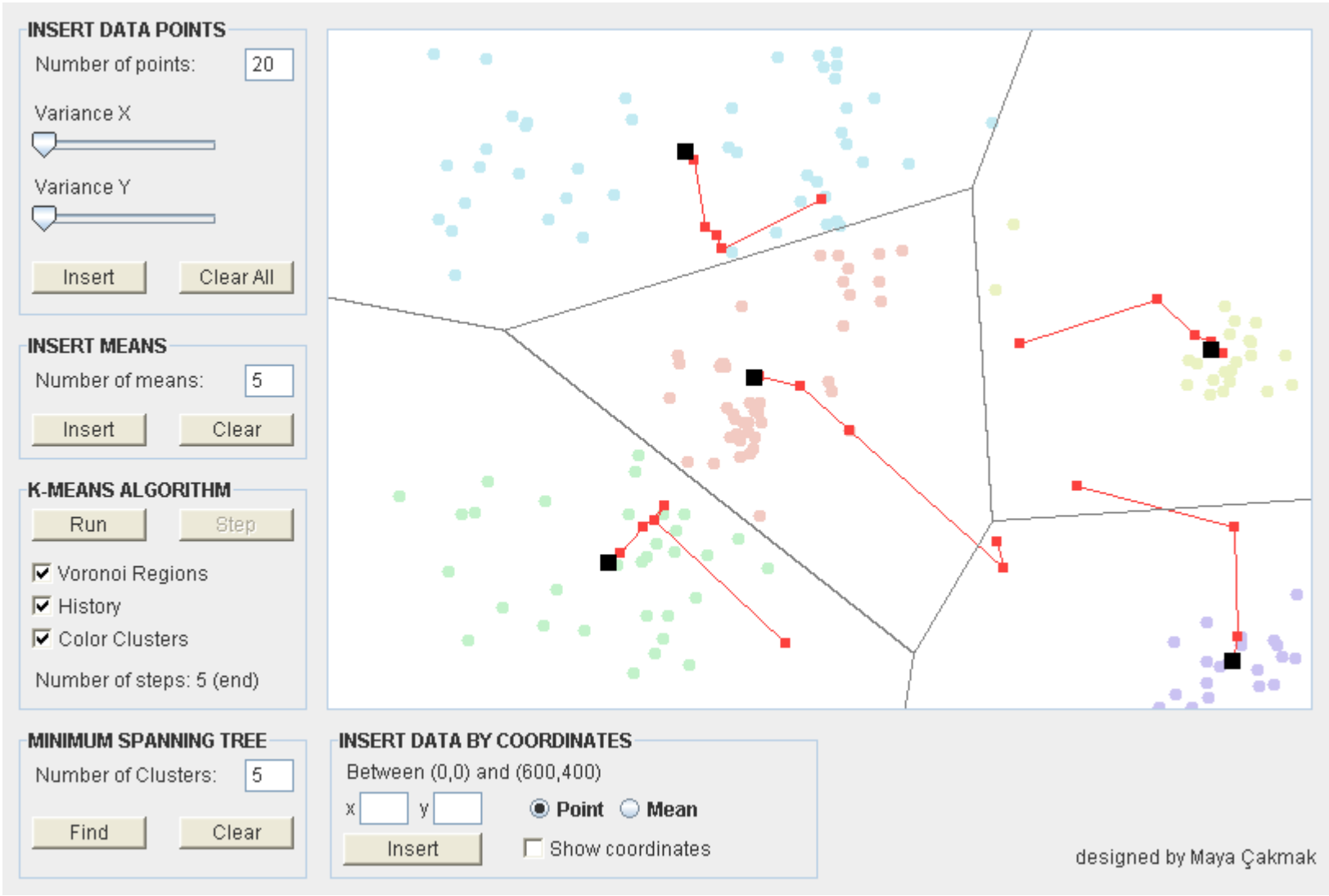




přeškálování

- Dejme tomu, že známe počet shluků (centroidů), jen hledáme jejich pozici.

K-means applet



http://www.kovan.ceng.metu.edu.tr/~maya/kmeans/index.html

Jak K-means pracuje?

- Náhodně inicializuj *k* centroidů. Opakuj dokud algoritmus nezkonverguje*:*
  - **fáze přiřazení vektorů**: každý vektor *x* přiřaď shluku X_i, pro který vzdálenost *x* od $\vec{\mu}_i$ (centroid X_i) je minimální
  - **fáze pohybu centroidů**: oprav pozici centroidů podle aktuálních vektorů ve shlucích

$$\vec{\mu}_i(X_i) = \frac{1}{|X_i|} \sum_{\vec{x}_j \in X_i} \vec{x}_j$$

K-means vlastnosti

- Lokálně minimalizujeme energii

$$\sum_{l=1}^{K} \sum_{x_i \in X_l} \| x_i - \mu_l \|^2$$

- Co to znamená?
  - Pro K shluků sečti vzdálenost všech vektorů daného shluku od jeho centroidu
- Konverguje vždy do globálního minima energie?
- Ne, často do lokálních minim. Závislost na inicializaci centroidů.

Algoritmus k-středů

- Pseudokód algoritmu K-středů
- vstup:
  - n vzorů a počet výsledných středů c
- výstup:
  - výsledné středy $\boldsymbol{\mu}_1,\ldots, \boldsymbol{\mu}_c$
- algoritmus:

```
1. begin initialize n, c, μ1,..., μc
2.        do klasifikuj n vzorů k jejich nejbližšímu μi
3.           přepočti μi
4.   until žádný μi se nezměnil
6.   return μ1,..., μc
7. end
```

složitost: O(ndcT)

- kde d je dimenze vzorů a T je počet iterací

Dětský k-means pseudokód

- Once there was a land with N houses...
- One day K kings arrived to this land..
- Each house was taken by the nearest king..
- But the community wanted their king to be at the center of the village, so the throne was moved there.
- Then the kings realized that some houses were closer to them now, so they took those houses, but they lost some.. This went on and on..(2-3-4)
- Until one day they couldn't move anymore, so they settled down and lived happily ever after in their village...

Počet středů (shluků)

- Pro K-means je třeba K určit předem – to je těžké když o datech nic nevíme
- Vymyslete algoritmus, který bude počet shluků odvozovat automaticky z dat.

- Např. Leader-follower strategie
- Problém – určit univerzální hodnotu prahu

Jaké použít kritérium pro volbu K?

- Minimum energie?

$$W(K) = \sum_{l=1}^{K} \sum_{x_i \in X_l} \| x_i - \mu_l \|^2$$

- Nevhodné, klesá k nule pro K=počet instancí.
- Lépe najít maximum funkce:

$$H(K) = \frac{W(K) - W(K+1)}{W(K+1)}$$

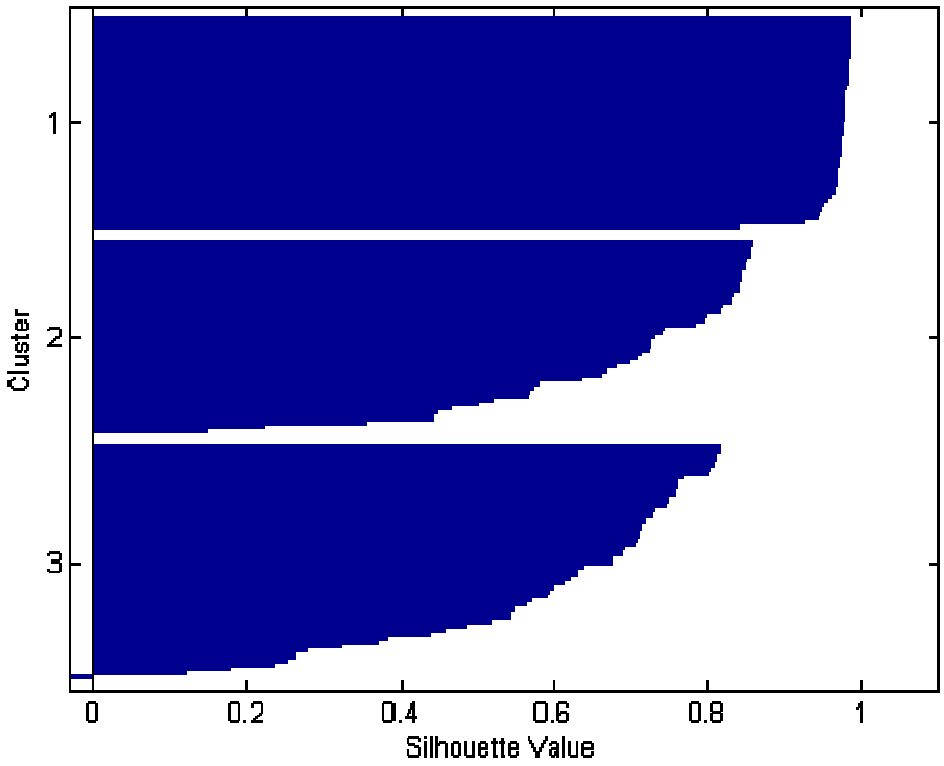
Silhouette – graf obrysů shluků

- Iris data, pro každou instanci vypočti jistotu zařazení do shluku *s(i)* ϵ <-1,1>

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

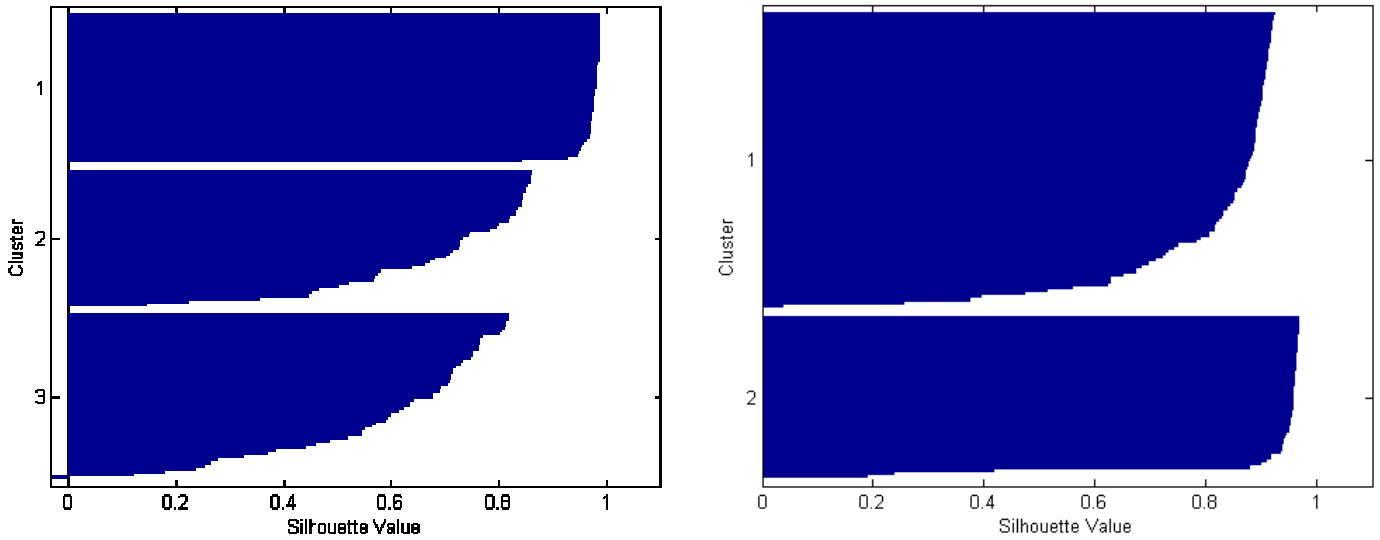
kde *a(i)* je průměrná vzdálenost instance i od instancí shluku, do kterého je zařazena

b(i) je průměrná vzdálenost instance i od instancí nejbližšího shluku



Hodnocení shluknutí pomocí Silhouette

- Který výstup K-means je lepší?



- Ten, který má lepší průměr hodnoty s(i) pro všechny instance.
- Ideálně na testovacích datech.

Hodnocení stability shluknutí

- Jak na to?
- Náhodným smazáním např. 10% různých instancí vygenerovat M podmožin dat
- Spustit shlukování pro všechny podmnožiny
- Spočítat průměr shody zařazení do shluků pro všechny kombinace podmnožin
- Čím vyšší, tím lepší …

Predikovatelnost shluků

- Použiji křížovou validaci
- Na každé trénovací množině shluknu data a natrénuji klasifikátor
- Spočítám podobnost klasifikace na testovacích datech a shlukování testovacích dat.

- Průměr pro všechny foldy by měl být co největší.

Shlukování založené na modelech

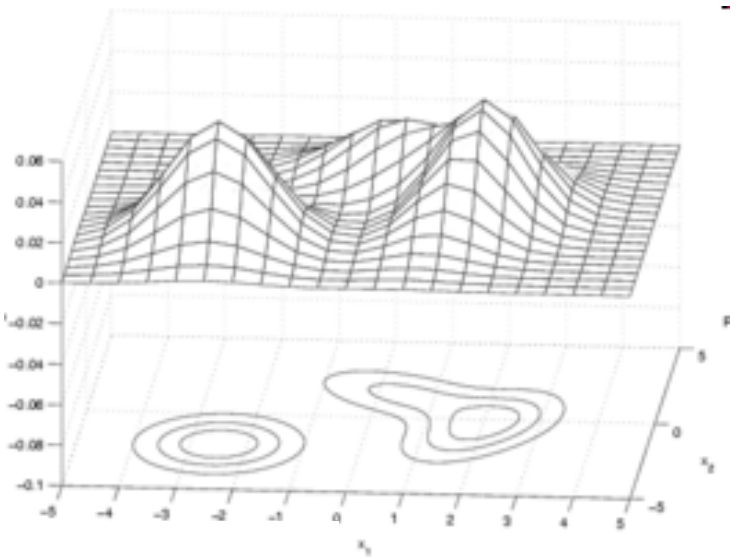
Např:

- Modelování hustoty pravděpodobnosti gaussovskou směsí (ukážeme si nyní)
- Model založený na samoorganizující se mapě (příští přednáška)

Gaussovská směs

- Mnoharozměrná gaussovská hustota pravděpodobnosti
- 1D gaussovská funkce

$$g(x) = ae^{-\frac{(x-b)^2}{2c^2}}$$



- Vícedimenzionální směs

$$\mathrm{g}_{\mathbf{M},\mathbf{C}}(\mathbf{x}) = \frac{1}{\sqrt{2\pi}^d \sqrt{\det(\mathbf{c})}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{M})^T \mathbf{C}^{-1}(\mathbf{x}-\mathbf{M})}$$

M a C ?

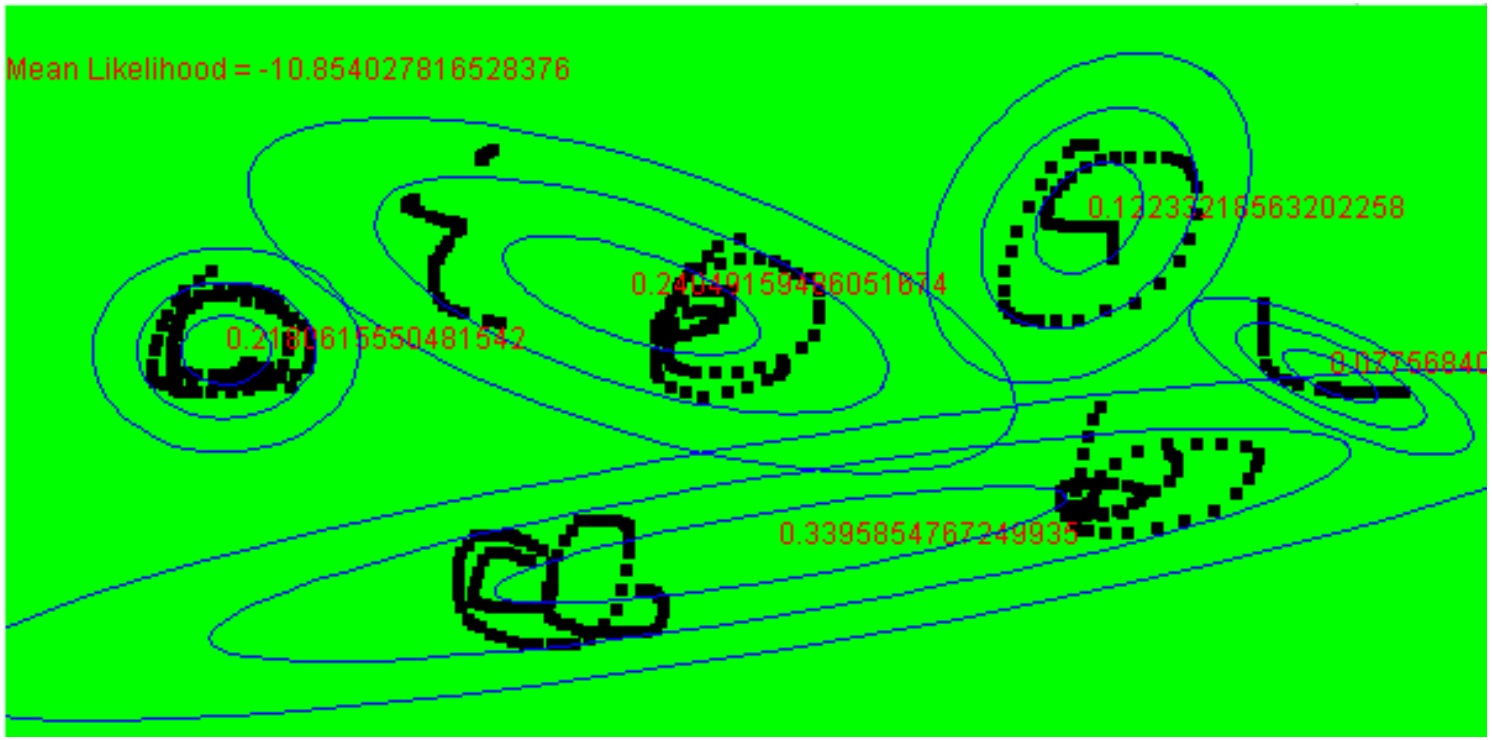
normalizace - pravděpodobnost

$$\mathrm{gm}(\mathbf{x}) = \sum_{k=1}^{K} w_k \cdot g_{\mathbf{M}_k,\mathbf{C}_k}(\mathbf{x})$$

vážený průměr K gaussovek

Směs gausovských rozdělení

- M_k a C_k počítáme např. pomocí EM algoritmu
- http://www.neurosci.aist.go.jp/~akaho/MixtureEM.html



- Každá gaussovka jeden shluk – předpokládá normální rozdělení dat ve shluku

K-Means a EM algoritmus

- *K*-means je speciální případ obecnější procedury nazývané *Expectation Maximization (EM)* algoritmus.
- **Expectation:** Použij aktuální parametry (a data) k rekonstrukci „černé skříňky"
- **Maximization:** Použij „černou skříňku" a data ke zpřesnění parametrů

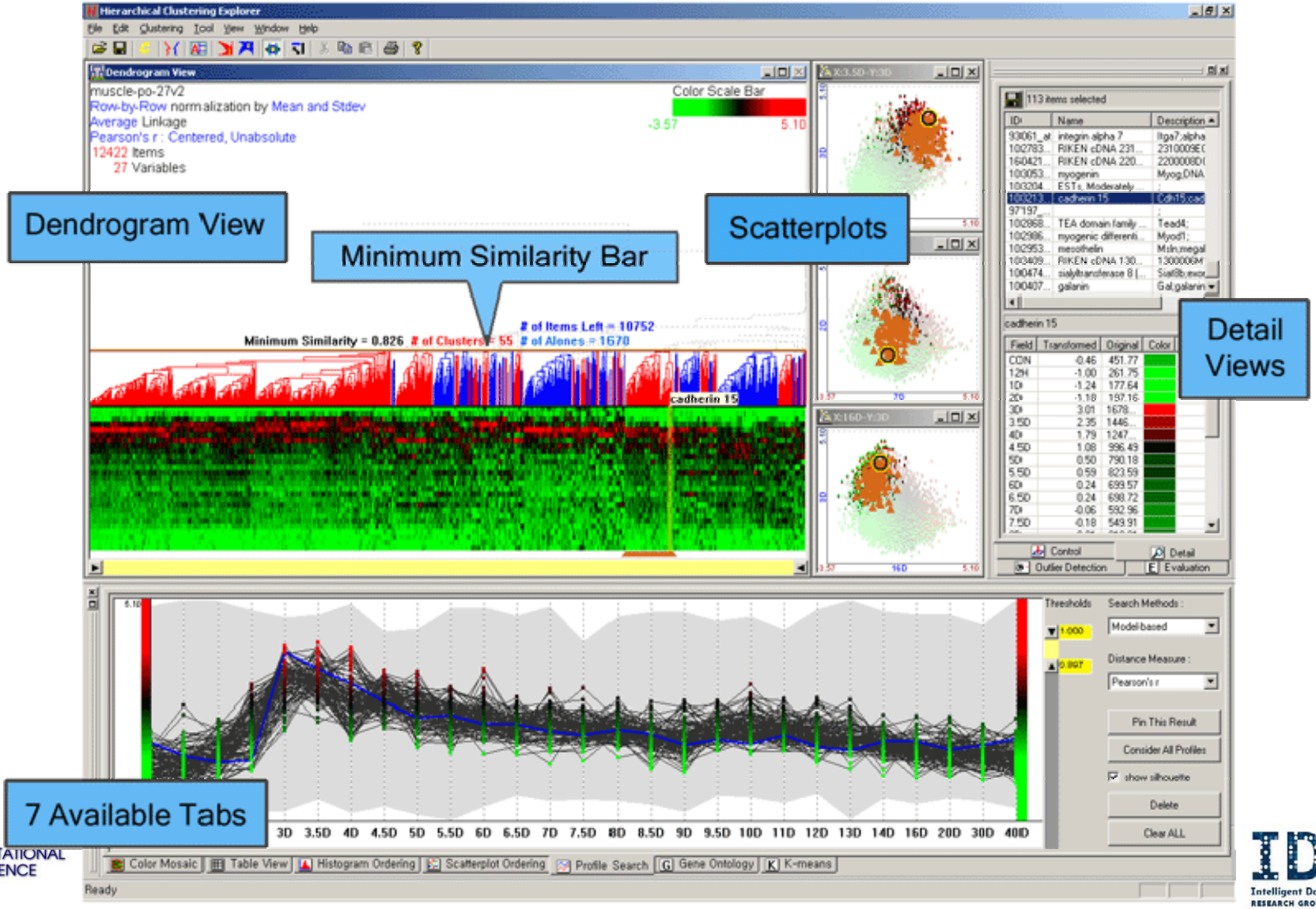
http://en.wikipedia.org/wiki/Expectation-maximization_algorithm

Aplikační oblasti shlukové analýzy

- Hledání podobností v datech
- Určování významnosti proměnných
- Detekce odlehlých instancí
- Redukce dat

Open source shlukovací nástroje

http://www.cs.umd.edu/hcil/hce/



COMPUTATIONAL INTELLIGENCE GROUP

IDA Intelligent Data Analysis RESEARCH GROUP