Vytěžování Dat
Přednáška 4 – Shluková analýza

Miroslav Čepek

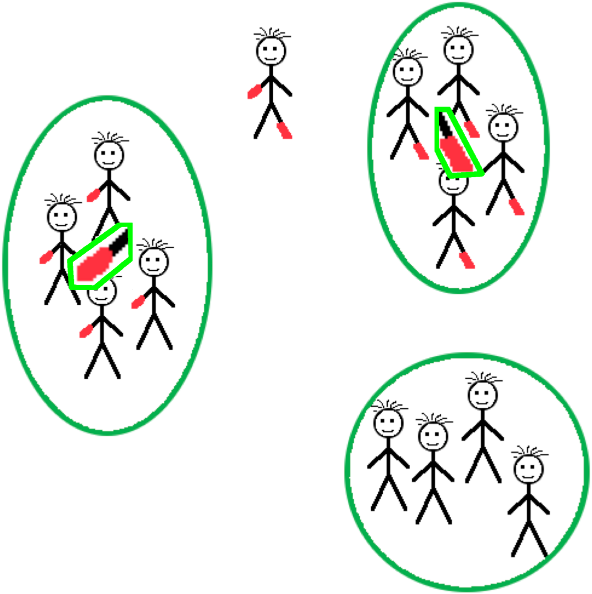
Katedra počítačů, Computational Intelligence Group

Evropský sociální fond Praha & EU:
Investujeme do vaší budoucnosti

14.10.2014

Co to je shluková analýza

- Je jednou ze základních úloh vytěžování dat.
- Jde o *třídění objektů do skupin* podle jejich vlastností.
    - Tak aby si objekty ve skupinách byly "nějak" podobné.
    - A zároveň nebyly podobné objektům v jiných skupinách.



Co to je shluková analýza (II)

- V principu jde o optimalizační problém.
- Co se musí optimalizovat?
  - Počet shluků (skupin).
  - Přiřazení instancí do shluků.

Co to je shluková analýza (II)

- V principu jde o optimalizační problém.
- Co se musí optimalizovat?
  - Počet shluků (skupin).
  - Přiřazení instancí do shluků.

Jak zjistit, že jsou si dva vzory podobné?

- To je obecně velmi složitá otázka.
- Shlukovou analýzu provádí hlavně počítače, obvyklé je zavedení číselné vzdálenostní funkce:
  - $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,
  - kde $\mathcal{X}$ je prostor instancí,
  - vzdálenost je nepřímo úměrná podobnosti.
- Při splnění několika podmínek mluvíme o (vzdálenostní) *metrice*:
  - $d(x, y) \geq 0$
  - $d(x, y) = d(y, x)$
  - $d(x, y) = 0 \Leftrightarrow x = y$
  - $d(x, y) + d(y, z) \geq d(x, y)$

Jak zjistit, že jsou si dva vzory podobné?

- To je obecně velmi složitá otázka.
- Shlukovou analýzu provádí hlavně počítače, obvyklé je zavedení číselné vzdálenostní funkce:
  - $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,
  - kde $\mathcal{X}$ je prostor instancí,
  - vzdálenost je nepřímo úměrná podobnosti.
- Při splnění několika podmínek mluvíme o (vzdálenostní) *metrice*:
  - $d(x, y) \geq 0$
  - $d(x, y) = d(y, x)$
  - $d(x, y) = 0 \Leftrightarrow x = y$
  - $d(x, y) + d(y, z) \geq d(x, y)$

Metriky

- Jaké znáte metriky?
  - Eukleidovská metrika
  - Manhattanská metrika
  - Kosinová metrika
- Příklady dalších metrik
  - Editační vzdálenost (vzdálenost dvou slov = počet změn, kterými můžu změnit jedno slovo na druhé)
  - Grafová metrika (počet hran, které musím v grafu projít, abych se dostal z jednoho uzlu do druhého)
  - `http://en.wikipedia.org/wiki/Metric_space`

Metriky

- Jaké znáte metriky?
  - Eukleidovská metrika
  - Manhattanská metrika
  - Kosinová metrika
- Příklady dalších metrik
  - Editační vzdálenost (vzdálenost dvou slov = počet změn, kterými můžu změnit jedno slovo na druhé)
  - Grafová metrika (počet hran, které musím v grafu projít, abych se dostal z jednoho uzlu do druhého)
  - `http://en.wikipedia.org/wiki/Metric_space`

Metriky

- Jaké znáte metriky?
  - Eukleidovská metrika
  - Manhattanská metrika
  - Kosinová metrika
- Příklady dalších metrik
  - Editační vzdálenost (vzdálenost dvou slov = počet změn, kterými můžu změnit jedno slovo na druhé)
  - Grafová metrika (počet hran, které musím v grafu projít, abych se dostal z jednoho uzlu do druhého)
  - `http://en.wikipedia.org/wiki/Metric_space`

Eukleidovská metrika

- Nejpřirozenější metrika, protože se s ní běžně setkáváme.
- Jak změříme vzdálenost dvou bodů na tabuli?
- Pravítkem :)!
- A když známe souřadnice, můžeme ji spočítat. Jak?

Eukleidovská metrika

- Nejpřirozenější metrika, protože se s ní běžně setkáváme.
- Jak změříme vzdálenost dvou bodů na tabuli?
- Pravítkem :)!
- A když známe souřadnice, můžeme ji spočítat. Jak?

Eukleidovská metrika (II)

- Pythagorova věta! $c = \sqrt{a^2 + b^2}$
- A Pythagorovu větu můžeme zobecnit pro $\mathbb{R}^n$

$$\mathbf{x} = (x_1, x_2, \ldots, x_n), \mathbf{y} = (y_1, y_2, \ldots, y_n)$$

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Eukleidovská metrika (II)

- Pythagorova věta! $c = \sqrt{a^2 + b^2}$
- A Pythagorovu větu můžeme zobecnit pro $\mathbb{R}^n$

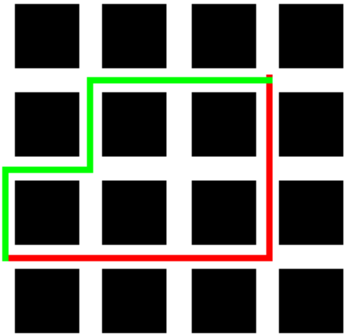
$$\mathbf{x} = (x_1, x_2, \ldots, x_n), \mathbf{y} = (y_1, y_2, \ldots, y_n)$$

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Manhattanská metrika (City-block distance)

- Základní myšlenka: Kolik bloků ve městě musím obejít, abych se dostal z jednoho místa na druhé?
- Nebo také – kolik tahů králem musím udělat abych se dostal z jednoho místa šachovnice na druhé?

Manhattanská metrika (City-block distance) (II)



- Pokud znám souřadnice, vzdálenost spočítam takto:

$$d(\mathbf{x},\mathbf{y}) = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_n - y_n|$$

Kosinová vzdálenost

- Vzdálenost dvou vektorů délky n odpovídá úhlu, který svírají:

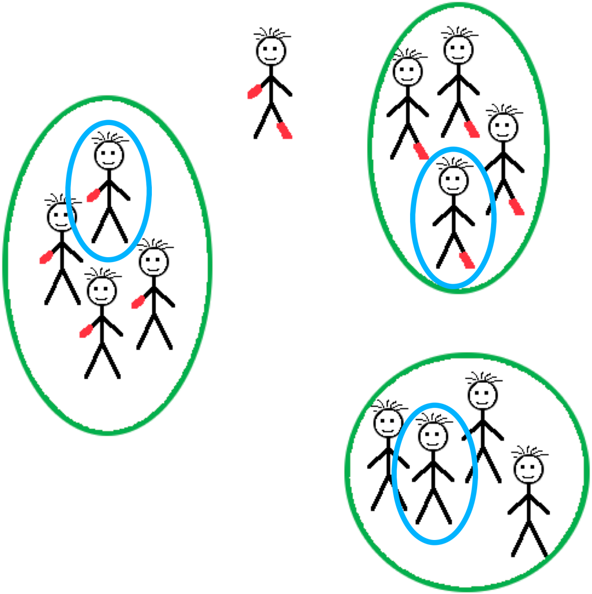
$$similarity(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n}(x_i y_i)}{\sqrt{\sum_{i=1}^{n}(x_i^2) \sum_{i=1}^{n}(y_i^2)}}$$

- Oborem hodnot této funkce je interval $\langle -1, +1 \rangle$.
- -1 znamená úplný opak, 0 nezávislost a $+1$ naprostou shodu.
- Převod na vzdálenost:

$$d(\mathbf{x}, \mathbf{y}) = 1 - similarity(\mathbf{x}, \mathbf{y})$$

Shlukování pomocí KMeans

- Jednotlivé shluky budou zastoupeny jedním reprezentantem.
  - Ten ponese vlastnosti typické pro danou skupinu/shluk.
- I každá instance (vzor) bude zastoupena jedním reprezentantem.
  - Tím, který je jí nejpodobnější.
  - Jinými slovy tím, který jí bude nejblíž (v dané metrice).



Algoritmus KMeans – značení

- Máme množinu N vstupních vzorů/instancí (vektorů) $\mathbf{x_i}$, $i \in 1 \ldots N$. Jednotlivé složky vektoru budeme označovat $x_i(s)$, , $s \in 1 \ldots n$.
- A máme množinu K reprezentantů, vektorů $\mathbf{c_k}^t$. Kde $k \in 1 \ldots K$ je index reprezentanta, t je číslo iteračního kroku, $c_k^t(s)$ je jednou z jeho složek.

Shlukování pomocí KMeans

- Jak určit, kde je správné místo pro reprezentanty shluků?
  - Vzdálenost mezi instancemi a jejich reprezentanty musí být co nejmenší.
  - Lze formulovat jako optimalizační problém:

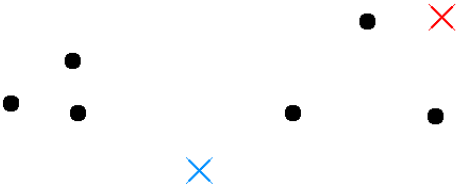
$$\arg\min_{\mathbf{c_1}\ldots\mathbf{c_K}} \sum_{i=1}^{N} min_k d^2(\mathbf{x_i}, \mathbf{c_k})$$

  - Jako d volíme euklidovskou vzdálenost.
  - Jeden z nejjednodušších postupů je iterační.

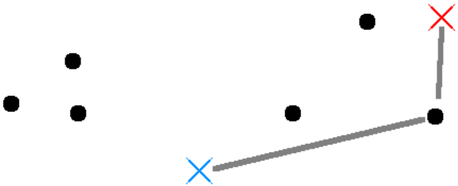
Algoritmus KMeans

1. Nastav reprezentanty ${\mathbf{c_k}}^0$ do náhodných počátečních bodů.
2. Najdi a přiřaď každé instanci jejího nejbližšího reprezentanta.
   - $\forall \mathbf{x_i}$ najdi k tak, aby $\forall j\ d(\mathbf{x_i}, {\mathbf{c_k}}^t) \leq d(\mathbf{x_i}, {\mathbf{c_j}}^t)$,
   - pro každé ${\mathbf{c_k}}^t$ vytvoř množinu $nearest_k^t$ instancí, ke kterým je nejblíž.
3. Přesuň reprezentanta tak, aby ležel "uprostřed" své množiny nejbližších instancí.
   - $c_k^{t+1}(s) = \frac{1}{\|nearest_k^t\|} \sum_{x_i \in nearest_k^t} x_i(s)$
4. Pokud se změnila poloha alespoň jednoho středu, vrať se na bod 2. Jinak skonči.

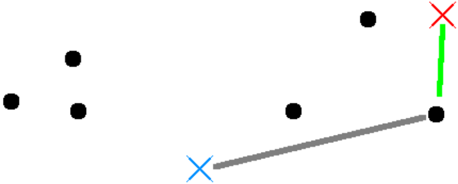
Ilustrace KMeans



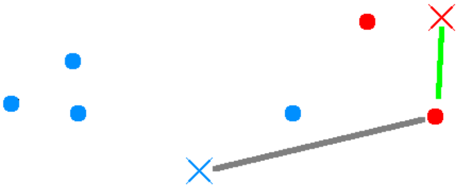
Ilustrace KMeans (II)



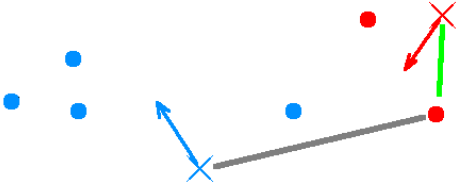
Ilustrace KMeans (III)



Ilustrace KMeans (IV)



Ilustrace KMeans (V)



Pohádka o Algoritmu KMeans :)

- 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...

KMeans jako varianta EM algoritmu

- Lze najít analogii mezi KMeans a dřívější látkou kurzu?
- Ano, řada společných znaků s EM algoritmem:
  - Skrytá proměnná = přiřazení instancí ke shlukům.
  - E-krok = pro dané středy shluků urči rozdělení instancí do shluků.
  - M-krok = pro dané rozdělení instancí odhadni nové středy shluků.
- Odlišnosti od EM:
  - Nepracuje s věrohodností (lze ale ukázat, že minimalizace vzdálenosti ke středům je analogií).
  - Uvažuje ostré (hard) přiřazení ke shlukům, nikoli pravděpodobnostní.
- Závěr: KMeans lze považovat za příklad hard EM algoritmu.
- Existuje i klasická aplikace EM pro shlukování, model je směsí gaussovských rozdělení.

KMeans jako varianta EM algoritmu

- Lze najít analogii mezi KMeans a dřívější látkou kurzu?
- Ano, řada společných znaků s EM algoritmem:
  - Skrytá proměnná = přiřazení instancí ke shlukům.
  - E-krok = pro dané středy shluků urči rozdělení instancí do shluků.
  - M-krok = pro dané rozdělení instancí odhadni nové středy shluků.
- Odlišnosti od EM:
  - Nepracuje s věrohodností (lze ale ukázat, že minimalizace vzdálenosti ke středům je analogií).
  - Uvažuje ostré (hard) přiřazení ke shlukům, nikoli pravděpodobnostní.
- Závěr: KMeans lze považovat za příklad hard EM algoritmu.
- Existuje i klasická aplikace EM pro shlukování, model je směsí gaussovských rozdělení.

Vlastnosti KMeans algoritmu

- Může se KMeans zacyklit? NE
- Dopadne shlukování pomocí KMeans pokaždé stejně? NE
- Jak určit správný počet středů (shluků)? heuristicky
- Jak vyhodnotit jestli shlukování dopadlo dobře a jestli jsme zvolili přiměřené K? viz dále

Vlastnosti KMeans algoritmu

- Může se KMeans zacyklit? NE
- Dopadne shlukování pomocí KMeans pokaždé stejně? NE
- Jak určit správný počet středů (shluků)? heuristicky
- Jak vyhodnotit jestli shlukování dopadlo dobře a jestli jsme zvolili přiměřené K? viz dále

Vlastnosti KMeans algoritmu

- Může se KMeans zacyklit? NE
- Dopadne shlukování pomocí KMeans pokaždé stejně? NE
- Jak určit správný počet středů (shluků)? heuristicky
- Jak vyhodnotit jestli shlukování dopadlo dobře a jestli jsme zvolili přiměřené K? viz dále

Vlastnosti KMeans algoritmu

- Může se KMeans zacyklit? NE
- Dopadne shlukování pomocí KMeans pokaždé stejně? NE
- Jak určit správný počet středů (shluků)? heuristicky
- Jak vyhodnotit jestli shlukování dopadlo dobře a jestli jsme zvolili přiměřené K? viz dále

Vlastnosti KMeans algoritmu

- Může se KMeans zacyklit? NE
- Dopadne shlukování pomocí KMeans pokaždé stejně? NE
- Jak určit správný počet středů (shluků)? heuristicky
- Jak vyhodnotit jestli shlukování dopadlo dobře a jestli jsme zvolili přiměřené K? viz dále

Vlastnosti KMeans algoritmu

- Může se KMeans zacyklit? NE
- Dopadne shlukování pomocí KMeans pokaždé stejně? NE
- Jak určit správný počet středů (shluků)? heuristicky
- Jak vyhodnotit jestli shlukování dopadlo dobře a jestli jsme zvolili přiměřené K? viz dále

Vyhodnocení rozkladu z KMeans algoritmu

- Jednou z možných metod je tzv. silueta.
- Silueta pro každou vstupní instanci spočítá jistotu zařazení instance do daného shluku.

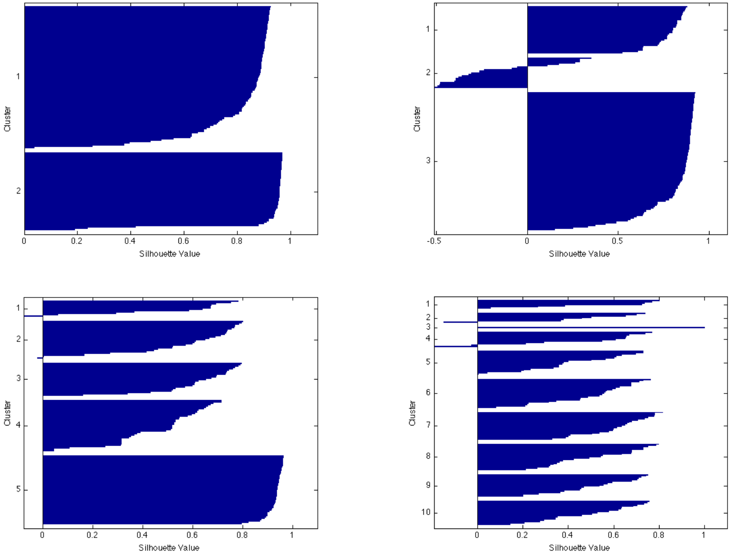
$$s(x_k) = \frac{b(x_k) - a(x_k)}{\max(a(x_k), b(x_k))}$$

- $a(x_k)$ je průměrná vzdálenost x_k od ostatních instancí shluku, ke kterému je přiřazena.
- $b(x_k)$ je průměrná vzdálenost x_k od instancí v nejbližším dalším shluku.
- Výsledné hodnoty jsou mezi -1 (x_k do shluku úplně nepatří) a $+1$ (úplně patří)
- `ftp://ftp.win.ua.ac.be/pub/preprints/87/Silgra87.pdf`

Vyhodnocení rozkladu z KMeans algoritmu (II)

- Pokud vypočítáte siluetu pro všechny instance a vykreslíte ji do grafu, můžete si udělat představu, jak shlukování dopadlo.

Ukázka Siluety – shluky Kosatců



Vyhodnocení rozkladu z KMeans algoritmu (III)

- Které shlukování dopadlo lépe?
- Co třeba průměrná silueta přes všechny instance (ideálně přes testovací data)?

Vyhodnocení rozkladu z KMeans algoritmu (III)

- Které shlukování dopadlo lépe?
- Co třeba průměrná silueta přes všechny instance (ideálně přes testovací data)?

Stabilita shluků

- Jak ověřit, že shluky nejsou nahodilé a odpovídají přirozeným třídám?
- Náhodným smazáním např. 10% různých instancí vygenerovat M podmnožin dat a spustit shlukování na každé podmnožině.

- Existuje několik ukázkových apletů/aplikací, kde si můžete zkusit, jak algoritmus funguje.
- `http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html`

Interpretace shluků

- Mám shluky, co s nimi dál?
  - Jaké skupiny objektů jednotlivé shluky reprezentují?
  - Jaké jsou jejich typické vlastnosti? Můžeme je stručně anotovat?
  - Co říkají pozice centroidů?
  - Generalizují shluky vzhledem k dalším příznakům nepoužitým pro shlukování?

Interpretace shluků

- Mám shluky, co s nimi dál?
  - Jaké skupiny objektů jednotlivé shluky reprezentují?
  - Jaké jsou jejich typické vlastnosti? Můžeme je stručně anotovat?
  - Co říkají pozice centroidů?
  - Generalizují shluky vzhledem k dalším příznakům nepoužitým pro shlukování?

Hierarchické shlukování – úvod

- KMeans, jak jsme viděli, má některé mouchy.
  - Kolik je v datech shluků?
  - Závislost výsledků na počátečních podmínkách.
  - Upřednostňuje kulovitý tvar shluků.
- Šlo by shlukování dělat i jinak?
- Šlo :). Jednou z možností je *Hierarchické shlukování*.
  - Základní myslenka je, že vytvoříme hierarchii shluků.
  - Lze postupovat zdola nahoru nebo shora dolů.
  - V prvním případě vždy spojíme dva nejpodobnější shluky do jednoho většího.
  - Pokračujeme dokud nevytvoříme jediný shluk se všemi objekty.

Hierarchické shlukování – úvod

- KMeans, jak jsme viděli, má některé mouchy.
  - Kolik je v datech shluků?
  - Závislost výsledků na počátečních podmínkách.
  - Upřednostňuje kulovitý tvar shluků.
- Šlo by shlukování dělat i jinak?
- Šlo :). Jednou z možností je *Hierarchické shlukování*.
  - Základní myslenka je, že vytvoříme hierarchii shluků.
  - Lze postupovat zdola nahoru nebo shora dolů.
  - V prvním případě vždy spojíme dva nejpodobnější shluky do jednoho většího.
  - Pokračujeme dokud nevytvoříme jediný shluk se všemi objekty.

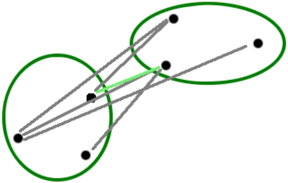
Aglomerativní hierarchické shlukování

1. Začne ze stavu, kdy každá instance je jedním shlukem.
2. Najdi dva nejbližší shluky.
3. Spoj je do jednoho.
4. Zůstávají nějaké shluky, které lze spojit? Pokud ano, vrať se na bod 2.

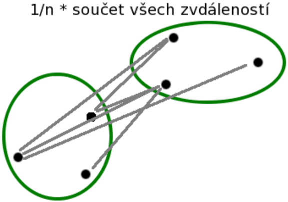
Nejbližší shluky

- Jak zjistím vzdálenost dvou shluků?
- Dokud shluky obsahují jen jednu instanci, je spočítání vzdálenosti jednoduché. Ale pak?
  - $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ zobecníme na $\delta : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \to \mathbb{R}$
- Vzdálenost shluků je určena
  - Nejbližší sousedé – vzdáleností nejbližších instancí ve shluku.
  - Nejvzdálenější sousedé – vzdáleností nejvzdálenějších instancí ve shluku.
  - Vzdálenost středů – vzdáleností center (středů) shluků.
  - Průměrná vzdálenost – průměrná vzdálenost mezi všemi instancemi v obo shlucích

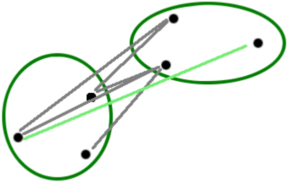
Vzdálenost shluků – ilustrace



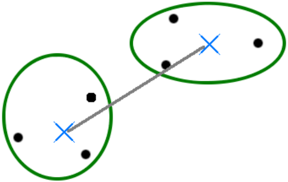
Nejkratší vzdálenost



Průměrná vzdálenost



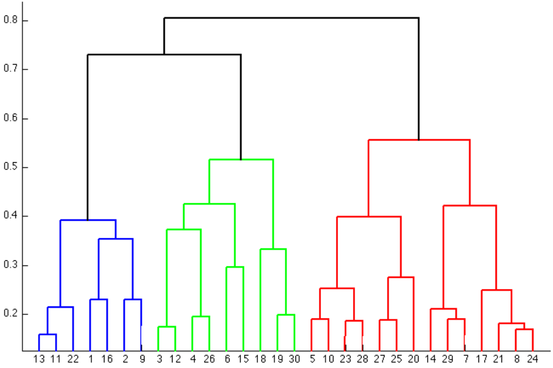
Největší vzdálenost



Vzdálenost mezi reprezentanty

Dendrogram

- Vizualizací postupu shlukování získáme strom – dendrogram.
- Jak nalezneme konkrétní rozklad do shluků?
  - Řezem dendrogramu na dané výšce.
  - Jak volíme počet shluků? Lze přihlédnout k výšce větví.



Vyhodnocení hierarchického shlukování

- Můžeme opět použít siluetu, stejně jako u KMeans.
- Jinou možností je CPCC (Cophenetic Correlation Coefficient).
  - 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.

Další informace a zdroje

- http://cmp.felk.cvut.cz/cmp/courses/recognition/zapis_prednasky/zapis_02/13/shlukovani.pdf
- Jain et al.: **Data Clustering: A Review.** ACM Computing Surveys.