GEOINFORMATIKA

Shluková analýza

Algoritmy neřízené klasifikace

Ing. Markéta Potůčková, Ph.D.
Katedra aplikované geoinformatiky a kartografie, Přf UK

Obsah přednášky

- Shlukování
- ***Opakování - Metody neřízené klasifikace v DPZ***
- Používané přístupy ke shlukování
  - Přehled
  - Míry podobnosti
- Příklady algoritmů
  - Sekvenční algoritmy
  - Hierarchické shlukování (dendrogram)
  - k-means
  - DBSCAN

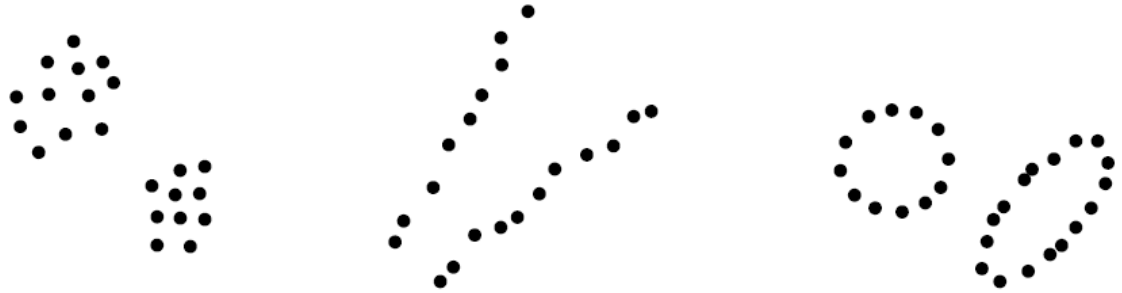
Literatura:

Koutroumbas, K., Theodoridis, S. (2008): *Pattern Recognition*, 4th Edition, Academic Press, 984 p. (online přístup, knihovna UK)
Brunton, S. L., Kutz, J. N. (2019). Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press.

Shlukování

- Metoda identifikace podobných skupin dat v souboru dat
- Při daném počtu shluků, které je třeba vytvořit, se hledá řešení, při kterém jsou všechny prvky v každém shluku co nejpodobnější a všechny shluky se od sebe co nejvíce liší
- Hledání struktur v datech, aniž by tyto byly vázány na konkrétní výsledek

Shlukování



Množina pozorování X = {$\boldsymbol{x}_1$, $\boldsymbol{x}_2$, …, $\boldsymbol{x}_N$}

- $\boldsymbol{x}_i$ … vektor příznaků, i = 1, …, L
- N … počet pozorování (např. bodů, pixelů, polygonů, textových řetězců, …)

Shlukování = rozdělení X do m podmnožin C_1, … C_m (shluků) tak, že platí

$$C_i \neq \emptyset, i = 1, \ldots, m$$

$$\cup_{i=1}^{m} C_i = X$$

$$C_i \cap C_j = \emptyset, i \neq j, i, j = 1, \ldots, m$$

fuzzy přístup definicí funkce příslušnosti u_j

$$u_j : X \rightarrow [0, 1], \quad j = 1, \ldots, m$$

$$\sum_{j=1}^{m} u_j(\boldsymbol{x}_i) = 1, \quad i = 1, 2, \ldots, N, \qquad 0 < \sum_{i=1}^{N} u_j(\boldsymbol{x}_i) < N, \quad j = 1, 2, \ldots, m$$

- vektory obsažené v shluku C_i jsou si „podobnější“ a jsou „méně podobné“ vektorům příznaků ostatních shluků.
- Kvantifikace „podobnosti“ závisí na tvaru shluku (kruhové, liniové, …)

Koutroumbas, K., Theodoridis, S. (2008)

Dílčí kroky shlukové analýzy

- Jsou data vhodná ke shlukové analýze, mají-li tendenci vytvářet shluky v daném příznakovém prostoru – shluková analýza nebude fungovat na data zcela náhodného charakteru
- Výběr příznaků (minimalizace redundance/korelace mezi příznaky)
- Míry „blízkosti“/„podobnosti“/„rozdílnosti“
- Rozhodovací kritérium (např. nákladová funkce)
- Algoritmus shlukování
- Validace
- Interpretace výsledku

Míra podobnosti/rozdílnosti (similarity/dissimilarity measures)

A *similarity measure* (SM) s on X is defined as

$$s : X \times X \to \mathcal{R}$$

such that

$$\exists s_0 \in \mathcal{R} : -\infty < s(\boldsymbol{x}, \boldsymbol{y}) \leq s_0 < +\infty, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in X$$

$$s(\boldsymbol{x}, \boldsymbol{x}) = s_0, \quad \forall \boldsymbol{x} \in X$$

and

$$s(\boldsymbol{x}, \boldsymbol{y}) = s(\boldsymbol{y}, \boldsymbol{x}), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in X$$

If in addition

$$s(\boldsymbol{x}, \boldsymbol{y}) = s_0 \quad \text{if and only if} \quad \boldsymbol{x} = \boldsymbol{y}$$

and

$$s(\boldsymbol{x}, \boldsymbol{y}) s(\boldsymbol{y}, \boldsymbol{z}) \leq [s(\boldsymbol{x}, \boldsymbol{y}) + s(\boldsymbol{y}, \boldsymbol{z})] s(\boldsymbol{x}, \boldsymbol{z}), \quad \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in X$$

s is called a *metric SM*.

A *dissimilarity measure* (DM) d on X is a function.

$$d : X \times X \to \mathcal{R}$$

where $\mathcal{R}$ is the set of real numbers, such that

$$\exists d_0 \in \mathcal{R} : -\infty < d_0 \leq d(\boldsymbol{x}, \boldsymbol{y}) < +\infty, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in X$$

$$d(\boldsymbol{x}, \boldsymbol{x}) = d_0, \quad \forall \boldsymbol{x} \in X$$

and

$$d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x}), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in X$$

If in addition

$$d(\boldsymbol{x}, \boldsymbol{y}) = d_0 \quad \text{if and only if} \quad \boldsymbol{x} = \boldsymbol{y}$$

and

$$d(\boldsymbol{x}, \boldsymbol{z}) \leq d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z}), \quad \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in X$$

d is called a *metric DM*.

- Příklad DM: Eukleidovská vzdálenost
- DM – hledáme minimum, SM – hledáme maximum

Koutroumbas, K., Theodoridis, S. (2008)

Míry rozdílnosti/podobnosti mezi dvěma body

DM metriky

- Lp norma $$d_p(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{l} w_i|x_i - y_i|^p\right)^{1/p}$$

- L_1 $$d_1(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{l} w_i|x_i - y_i|$$

L_2, $w_i=1$ $$d_2(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{l}(x_i - y_i)^2}$$

L_2, vážená $$d(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^T B(\boldsymbol{x}-\boldsymbol{y})}$$

L_∞ $$d_\infty(\boldsymbol{x},\boldsymbol{y}) = \max_{1\le i\le l} w_i|x_i - y_i|$$

SM metriky

- Skalární součin a kosinus úhlu příznakových vektorů

$$s_{\text{inner}}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^T\boldsymbol{y} = \sum_{i=1}^{l} x_i y_i \qquad s_{\text{cosine}}(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\|\boldsymbol{x}\|\,\|\boldsymbol{y}\|}$$

- Pearsonův korelační koeficient $$r_{\text{Pearson}}(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}_d{}^T\boldsymbol{y}_d}{\|\boldsymbol{x}_d\|\,\|\boldsymbol{y}_d\|}$$

V případě konečného počtu celočíselných hodnot vektorů ***x***, ***y*** lze vytvořit kontingenční tabulku a z ní spočítat jako DM tzv. Hammingovu vzdálenost (suma nediagonálních prvků)

$$d_H(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=0}^{k-1}\sum_{j=0, j\neq i}^{k-1} a_{ij}$$

Míry rozdílnosti/podobnosti mezi bodem a množinou bodů

- Používá se v případě, že je třeba ohodnotit míru shody mezi vektorem ***x*** a shlukem *C*

A) Hodnotí se vztah ***x*** vůči všem prvkům *C*

The *max proximity function:*

$$\wp^{ps}_{\max}(\boldsymbol{x}, C) = \max_{\boldsymbol{y}\in C} \wp(\boldsymbol{x}, \boldsymbol{y})$$

The *min proximity function:*

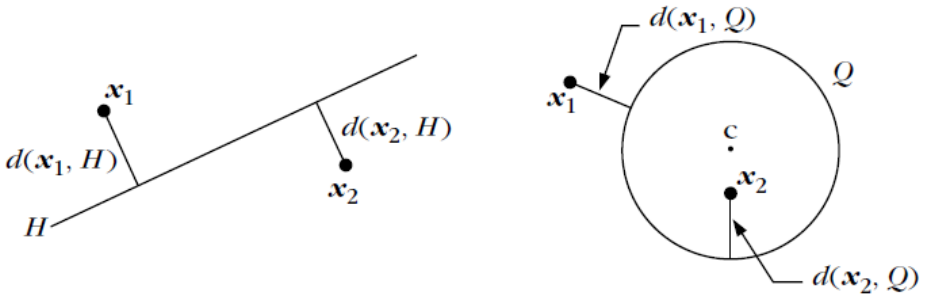
$$\wp^{ps}_{\min}(\boldsymbol{x}, C) = \min_{\boldsymbol{y}\in C} \wp(\boldsymbol{x}, \boldsymbol{y})$$

The *average proximity function:*

$$\wp^{ps}_{\text{avg}}(\boldsymbol{x}, C) = \frac{1}{n_C} \sum_{\boldsymbol{y}\in C} \wp(\boldsymbol{x}, \boldsymbol{y})$$

B) Hodnotí se vztah ***x*** vůči reprezentativnímu výběru z *C*

- průměr, medián
- vzdálenost k nadrovině, „hypersféře“, …



Koutroumbas, K., Theodoridis, S. (2008)

Kolika způsoby můžeme m shluků vytvořit?

Množina pozorování X = {$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$}

- $\boldsymbol{x}_i$ … vektor příznaků, i = 1, …, L
- N … počet pozorování (např. pixelů v obraze)

S(N, m) … možnosti shlukování N vektorů do m shluků

$$S(N,1) = 1$$

$$S(N,N) = 1$$

$$S(N,m) = 0, \quad \text{for } m > N$$

jinak

$$S(N,m) = \frac{1}{m!}\sum_{i=0}^{m}(-1)^{m-i}\binom{m}{i}i^N$$

Příklad:

$$S(15,3) = 2375101$$

$$S(20,4) = 45232115901$$

$$S(25,8) = 69022372111836858O$$

$$S(100,5) \simeq 10^{68}$$

Přístupy ke shlukové analýze

- Sekvenční přístup
- Hierarchický přístup (spojování či rozdělování datasetu)
- Optimalizace nákladové funkce

- Další přístupy:
  - Valley-seeking clustering algorithms
  - Density-based algorithms
  - Kernel-based methods

(dle Koutroumbas, K., Theodoridis, S., 2008)

Základní schéma sekvenčního algoritmu shlukování

- Počet tříd není znám, příznakové vektory vstupují do algoritmu pouze jednou
- Parametry definované uživatelem: prahová hodnota DM - Θ, maximální počet shluků - q
- Volba DM, tj. d($\boldsymbol{x}$, C) vede k různým algoritmům
- Co ovlivní výsledek?
  - Pořadí vstupujících vektorů
  - Volba Θ (čím větší, tím menší počet shluků)
- Je-li C reprezentován bodem, algoritmus zvýhodňuje kompaktní (kruhové) shluky
- Protože se očekává, že konečný počet shluků m bude mnohem menší než N, je časová složitost BSAS O(N)
- Možné úpravy:
  - Příznakové vektory vstupují do algoritmu 2x – prvně pro vytvoření shluků, podruhé pro konečné přiřazení ke shlukům
  - Volba více prahových hodnot Θ
  - Spojování shluků
  - Změna pořadí vstupních vektorů

Basic Sequential Algorithmic Scheme (BSAS)

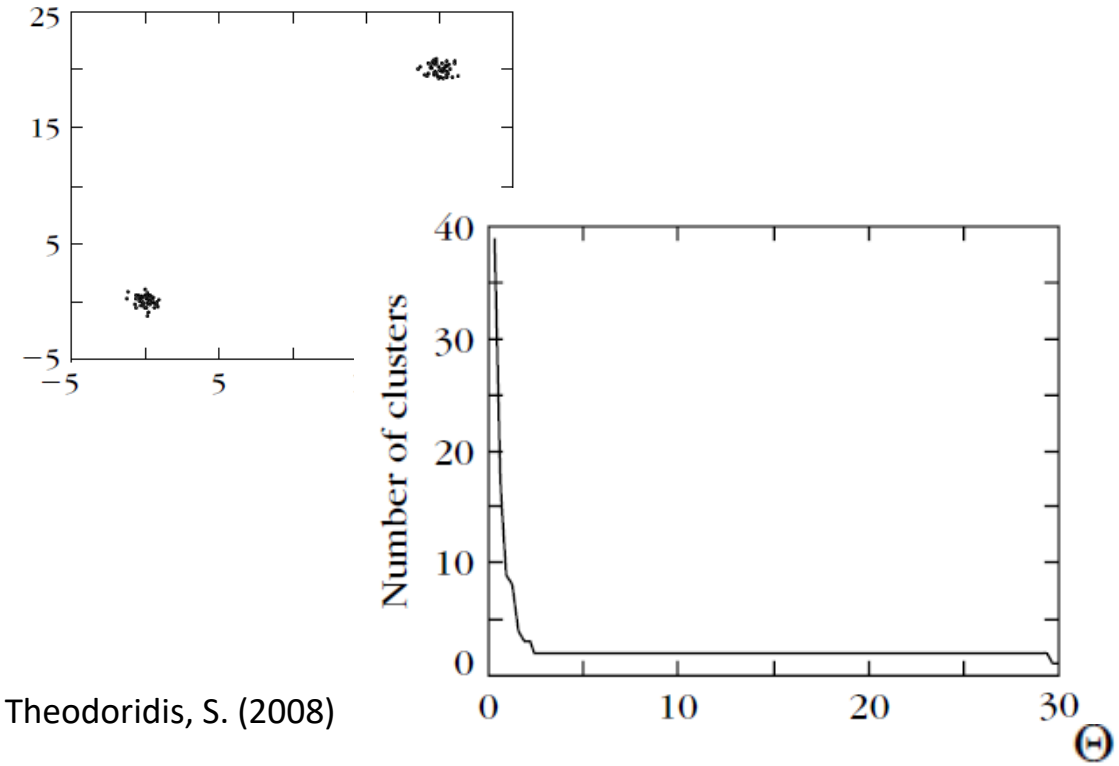
- $m = 1$
- $C_m = \{\boldsymbol{x}_1\}$
- For $i = 2$ to N
  - Find C_k: $d(\boldsymbol{x}_i, C_k) = \min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$.
  - If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$ then
    - $m = m + 1$
    - $C_m = \{\boldsymbol{x}_i\}$
  - Else
    - $C_k = C_k \cup \{\boldsymbol{x}_i\}$
    - Where necessary, update representatives[2]
  - End {if}
- End {For}

Koutroumbas, K., Theodoridis, S. (2008)

Optimální počet shluků

- S využitím sekvenčního shlukování
- Hodnoty *a*, *b* jsou minimální a maximální hodnoty DM mezi všemi páry vektorů v *X*, tj. a = $\min_{i,j=1,\ldots,N} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$, b = $\max_{i,j=1,\ldots,N} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$.
  Volba c je přímo ovlivněna typem $d(\boldsymbol{x}, C)$.
- Čím větší s, tím větší je statistický vzorek a tím vyšší přesnost výsledků

- For $\Theta = a$ to b step c
  - Run s times the algorithm BSAS(Θ), each time presenting the data in a different order.
  - Estimate the number of clusters, m_Θ, as the most frequent number resulting from the s runs of BSAS(Θ).
- Next Θ



Koutroumbas, K., Theodoridis, S. (2008)

Hierarchické shlukování

Algoritmy zahrnují N kroků (počet vstupních vektorů). V každém kroku t se získá nové shlukování na základě shlukování vytvořeného v předchozím kroku t-1.

Dvě hlavní kategorie

- Aglomerativní hierarchické algoritmy
  - Vychází z N shluků (= každý vstupní vektor reprezentuje jeden shluk) a ty postupně spojuje, výsledný shluk obsahuje celý dataset *X*
- Dělící hierarchické algoritmy
  - Výchozí shluk obsahuje celý dataset *X* a postupně je dělen až do úrovně dílčích vektorů

Obecný postup aglomerativního hierarchického shlukování

- Pokud se dva vektory spojí do jednoho shluku na úrovni t, zůstanou ve stejném shluku pro všechny následující shlukování → neexistuje způsob, jak napravit „špatné“ shlukování, ke kterému mohlo dojít na nižší úrovni hierarchie
- Celkový počet dvojic, které je třeba prozkoumat během celého procesu shlukování, je

$$\frac{(N-1)N(N+1)}{6}$$

  tj. výpočetní náročnost algoritmu obecně odpovídá N^3, ale existují varianty $O(N^2 \log N)$ či $O(N^2)$
- $g(C_i, C_j)$ je řešena na základě matice rozdílnosti (disimilarity matrix) či grafových algoritmů

- Initialization:
  - Choose $\Re_0 = \{C_i = \{\boldsymbol{x_i}\},\ i = 1, \ldots, N\}$ as the initial clustering.
  - $t = 0$.
- Repeat:
  - $t = t + 1$
  - Among all possible pairs of clusters (C_r, C_s) in $\Re_{t-1}$ find the one, say (C_i, C_j), such that

$$g(C_i, C_j) = \begin{cases} \min_{r,s} g(C_r, C_s), & \text{if } g \text{ is a dissimilarity function} \\ \max_{r,s} g(C_r, C_s), & \text{if } g \text{ is a similarity function} \end{cases} \quad (13.1)$$

  - Define $C_q = C_i \cup C_j$ and produce the new clustering $\Re_t = (\Re_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$.
- Until all vectors lie in a single cluster.

Koutroumbas, K., Theodoridis, S. (2008)

Aglomerativní hierarchické shlukování - dendrogram

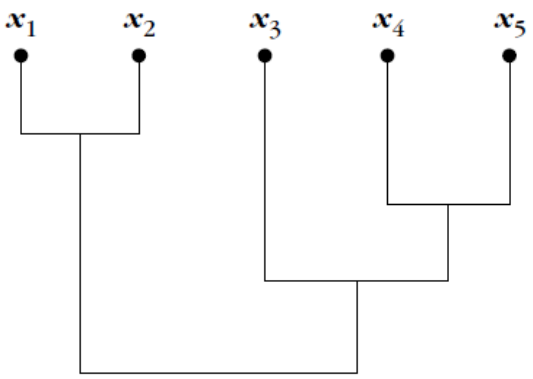
$\{\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

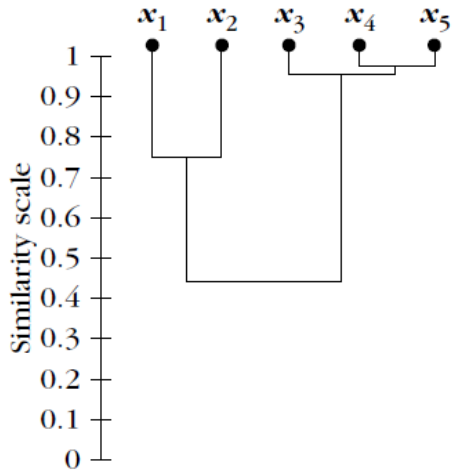
$\{\{x_1, x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

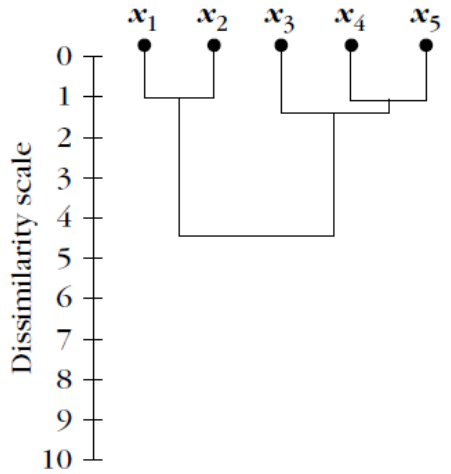
$\{\{x_1, x_2\},\{x_3\},\{x_4, x_5\}\}$

$\{\{x_1, x_2\},\{x_3, x_4, x_5\}\}$

$\{\{x_1, x_2, x_3, x_4, x_5\}\}$







Která úroveň je dostačující?

Příklad kritéria:
Definujeme DM uvnitř shluku C např. $h_1(C) = \max\{d(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x}, \boldsymbol{y} \in C\}$ $h_2(C) = \text{med}\{d(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x}, \boldsymbol{y} \in C\}$
a prahovou hodnotu Θ
Algoritmus bude ukončen na úrovni t, platí-li $\exists C_j \in \Re_{t+1} : h(C_j) > \theta$

Koutroumbas, K., Theodoridis, S. (2008)

Shlukování s optimalizační funkcí

- Nejvyužívanější postup shlukové analýzy
- Založen na optimalizaci nákladové funkce *J* (funkce vektorů datové sady X) parametrizované neznámým vektorem Θ
- Většina algoritmů vyžaduje znalost počtu shluků m
- Využíváno i pro geometrické úlohy (např. robotika a rozeznávání objektů splňujících dané geometrické parametry – rovina, sféra atd.)
- Různé přístupy:
  - Mixture decomposition – nákladová funkce je konstruována na základě náhodných vektorů a přiřazení do shluků se řídí rozdělením pravděpodobnosti (paralela s Bayesovskou klasifikací); podmíněné pravděpodobnosti jsou výsledkem procesu optimalizace
  - Fuzzy – pracuje s DM/SM jako funkcí „blízkosti“ hodnoceného vektoru a shluku a zavádí funkce příslušnosti
  - Hard - speciální případ přístupu fuzzy shlukování, kde každý vektor patří výhradně do jednoho shluku

Obecné schéma tvrdého klasifikátoru s optimalizační funkcí

- Choose $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j, j = 1, \ldots, m$.
- $t = 0$
- Repeat
  - For $i = 1$ to N
    - For $j = 1$ to m
      - *Determination of the partition:*[10]

$$u_{ij}(t) = \begin{cases} 1, & \text{if } d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t)) = \min_{k=1,\ldots,m} d(\boldsymbol{x}_i, \boldsymbol{\theta}_k(t)) \\ 0, & \text{otherwise,} \end{cases}$$

    - End {For-j}
  - End {For-i}
  - $t = t + 1$
  - For $j = 1$ to m
    - *Parameter updating*: Solve

$$\sum_{i=1}^{N} u_{ij}(t-1) \frac{\partial d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_j} = \mathbf{0}$$

    with respect to $\boldsymbol{\theta}_j$ and set $\boldsymbol{\theta}_j(t)$ equal to the computed solution.
  - End {For-j}.
- Until a termination criterion is met.

Optimalizační nákladová funkce

$$J(\boldsymbol{\theta}, U) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij} d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$$

$$u_{ij} \in \{0, 1\}, \qquad j = 1, \ldots, m$$

$$\sum_{j=1}^{m} u_{ij} = 1$$

Ukončení výpočtu

$$\|\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)\| < \varepsilon$$

Koutroumbas, K., Theodoridis, S. (2008)

ISODATA, k-means

- Hledáme minimum nákladové funkce *J(Θ, U)*

$$J(\boldsymbol{\theta}, U) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij} \|\boldsymbol{x}_i - \boldsymbol{\theta}_j\|^2$$

- Algoritmus generuje co nejkompaktnější shluky
- Nízká výpočetní náročnost *O(Nmq)*
  - q … počet nutných iterací
  - vhodný pro velké datové sady
- Mnoho variant algoritmu
- Algoritmus nemusí vést k nalezení globálního minima funkce *J(Θ, U)*, různé inicializace algoritmu mohou vést k různým řešením
- Počet shluků je vstupním parametrem
- Otázka optimálního počtu shluků („NP-úplný problém“)
- Citlivý k šumu a odlehlým hodnotám
- Optimalizace středu shluků k-means++

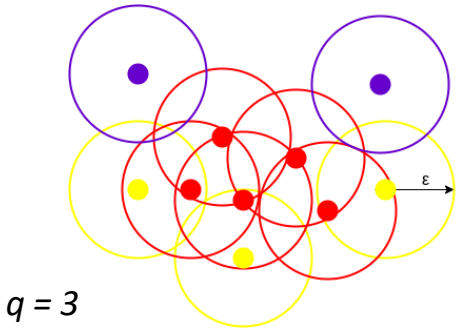
- Choose arbitrary initial estimates $\boldsymbol{\theta}_j(0)$ for the $\boldsymbol{\theta}_j$'s, $j = 1, \ldots, m$.
- Repeat
  - For $i = 1$ to N
    - Determine the closest representative, say $\boldsymbol{\theta}_j$, for $\boldsymbol{x}_i$.
    - Set $b(i) = j$.
  - End {For}
  - For $j = 1$ to m
    - Parameter updating: Determine $\boldsymbol{\theta}_j$ as the mean of the vectors $\boldsymbol{x}_i \in X$ with $b(i) = j$.
  - End {For}.
- Until no change in $\boldsymbol{\theta}_j$'s occurs between two successive iterations.

Koutroumbas, K., Theodoridis, S. (2008)

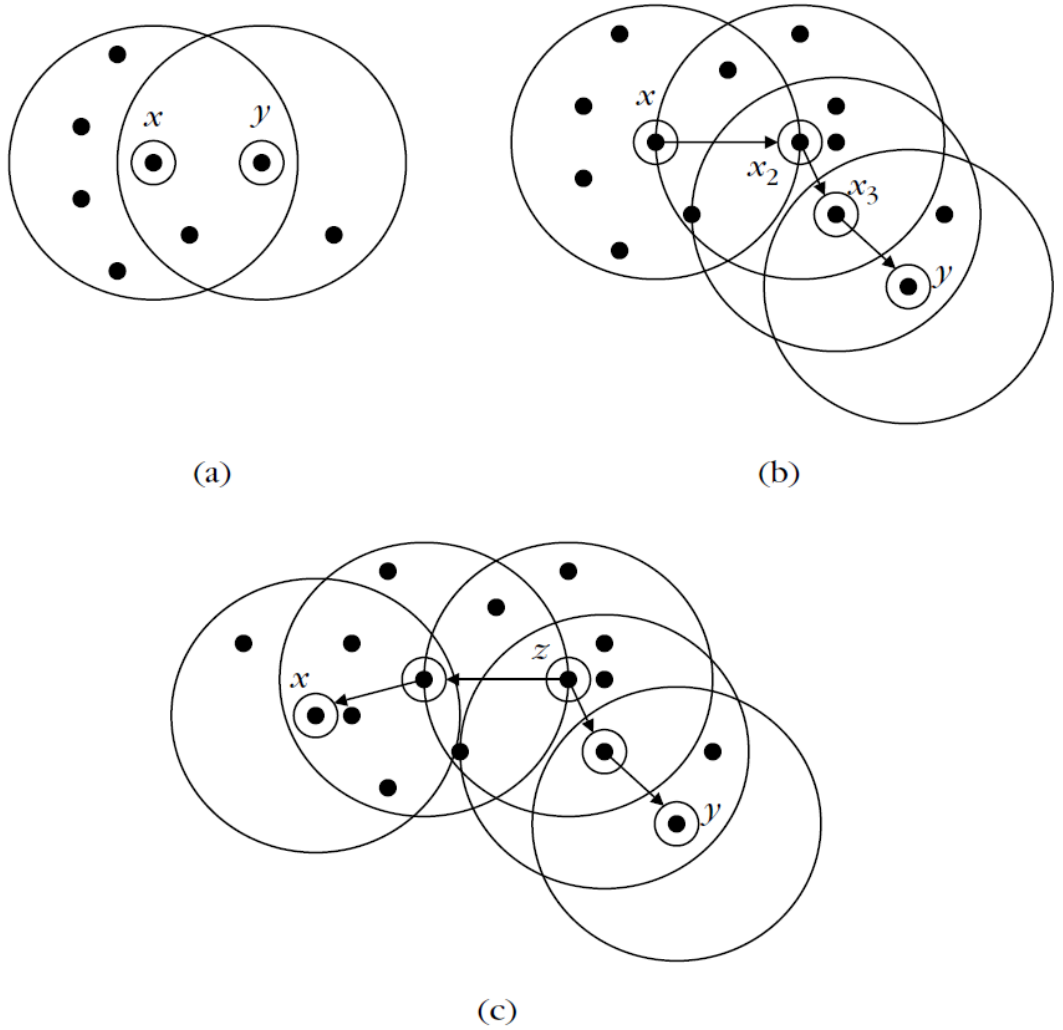
Rozdíl ISODATA, k-means?

DBSCAN

- Density-Based Spatial Clustering of Applications with Noise
- „hustota“ bodu ***x*** = počet bodů *X*, které se nacházejí v definovaném okolí bodu ***x***
- Obvykle se uvažuje sférické okolí $V_\varepsilon(\boldsymbol{x})$ o uživatelem definovaném poloměru ε
- Bod x označíme jako vnitřní (jádrový, „core“ bod shluku, pokud počet bodů v jeho okolí $N_\varepsilon(\boldsymbol{x}) > q$ (parametr q volí uživatel)
- Hraniční bod – dosažitelný z vnitřního bodu
- Šum



q = 3



Koutroumbas, K., Theodoridis, S. (2008)

FIGURE 15.22

Assuming that $q = 5$, (a) $\boldsymbol{y}$ is directly density reachable from $\boldsymbol{x}$, but not vice versa, (b) $\boldsymbol{y}$ is density reachable from $\boldsymbol{x}$, but not vice versa, and (c) $\boldsymbol{x}$ and $\boldsymbol{y}$ are density connected (in addition, $\boldsymbol{y}$ is density reachable from $\boldsymbol{x}$, but not vice versa).

DBSCAN

- Set $X_{un} = X$
- Set $m = 0$
- While $X_{un} \neq \emptyset$ do
  - Arbitrarily select a $\boldsymbol{x} \in X_{un}$.
  - If $\boldsymbol{x}$ is a noncore point then
    - Mark $\boldsymbol{x}$ as noise point.
    - $X_{un} = X_{un} - \{\boldsymbol{x}\}$
  - If $\boldsymbol{x}$ is a core point then
    - $m = m + 1$
    - Determine all density-reachable points in X from $\boldsymbol{x}$.
    - Assign $\boldsymbol{x}$ and the previous points to the cluster C_m. The border points that may have been marked as noise are also assigned to C_m.
    - $X_{un} = X_{un} - C_m$
  - End { if }
- End { while }

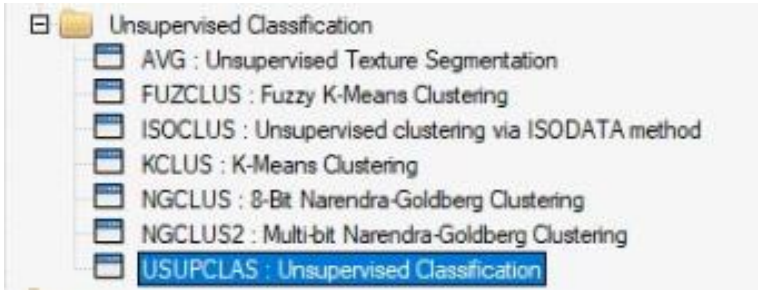
- The results of the algorithm are greatly influenced by the choice of ε and q. Different values of the parameters may lead to totally different results. One should select these parameters so that the algorithm is able to detect the least "dense" cluster. In practice, one has to experiment with several values for ε and q in order to identify their "best" combination for the data set at hand.

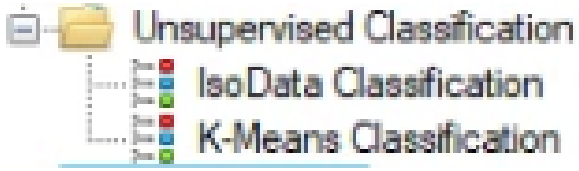
- The DBSCAN is not appropriate for cases where the clusters in X exhibit significant differences in density, and it is not well suited for high-dimensional data ([Xu 05]).

Koutroumbas, K., Theodoridis, S. (2008)

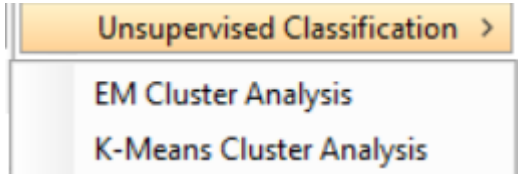
Co nabízejí softwary pro zpracování obrazu DPZ?

PCI GEOMATICS

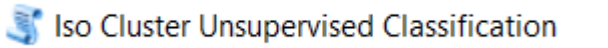




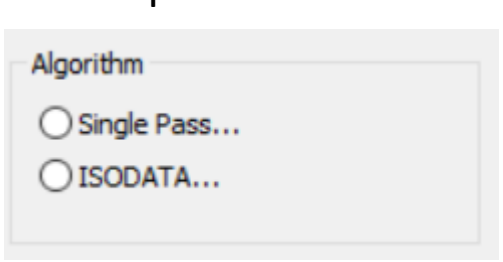
SNAP



esri



Multispec



Shlukové analýzy v Matlabu

MATLAB® supports many popular cluster analysis algorithms:

- **Hierarchical clustering** builds a multilevel hierarchy of clusters by creating a cluster tree.
- **k-Means clustering** partitions data into k distinct clusters based on distance to the centroid of a cluster.
- **Gaussian mixture models** form clusters as a mixture of multivariate normal density components.
- **Spatial clustering** (such as the popular density-based DBSCAN) groups points that are close to each other in areas of high density, keeping track of outliers in low-density regions. Can handle arbitrary non-convex shapes.
- **Self-organizing maps** use neural networks that learn the topology and distribution of the data.
- **Spectral clustering** transforms input data into a graph-based representation where the clusters are better separated than in the original feature space. The number of clusters can be estimated by studying eigenvalues of the graph.
- **Hidden Markov models** can be used to discover patterns in sequences, such as genes and proteins in bioinformatics.

Cvičení - Matlab

1. Vygenerujte alespoň 3 množiny bodů/klastry (např. funkce *randn*) ve 2D.
2. Vytvořte vlastní implementaci algoritmu shlukování k-means.
3. Porovnejte svůj výsledek s výsledkem shlukování nad totožnou množinou s využitím funkce *kmeans.* Případné rozdíly komentujte.

Bonusová úloha: rozšiřte své řešení shlukování k-means do n-dimensionálního prostoru.

Děkuji za pozornost

marketa.potuckova@natur.cuni.cz