Shlukování

Zpracováno s využitím skvělého tutoriálu autorů Eamonn Keogh, Ziv Bar-Joseph a Andrew Moore

Osnova přednášky

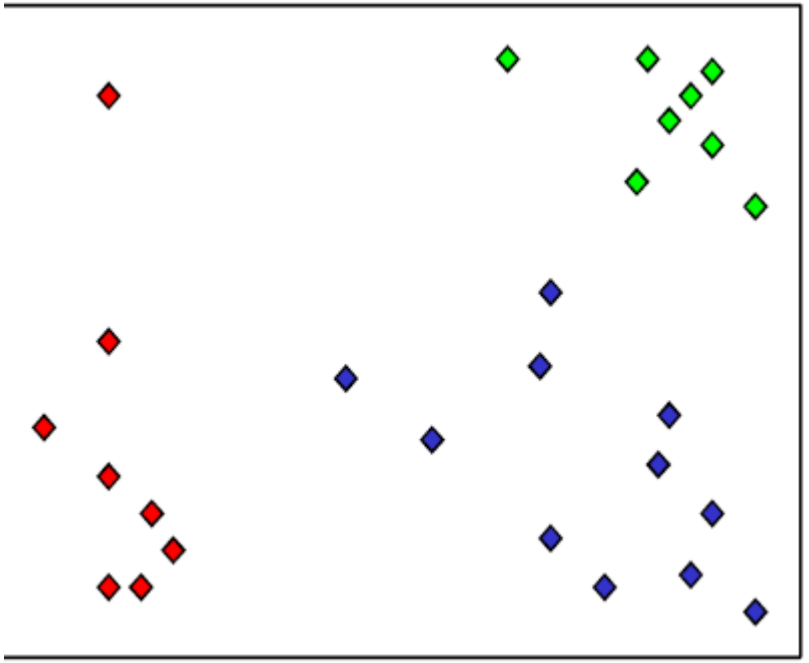
- **Motivace**
- Míra vzdálenosti
- Hierarchické shlukování
- Shlukování rozkladem
  - k-means (k-středů)
  - EM (expectation maximization) algoritmus, Gaussovská směs
  - Odhad počtu shluků

Co je to shlukování?

Seskupení dat do takových ***shluků***, že podobnost dat

- uvnitř shluku je vysoká,
- z různých shluků je nízká.

Hledáme „přirozené“ seskupení

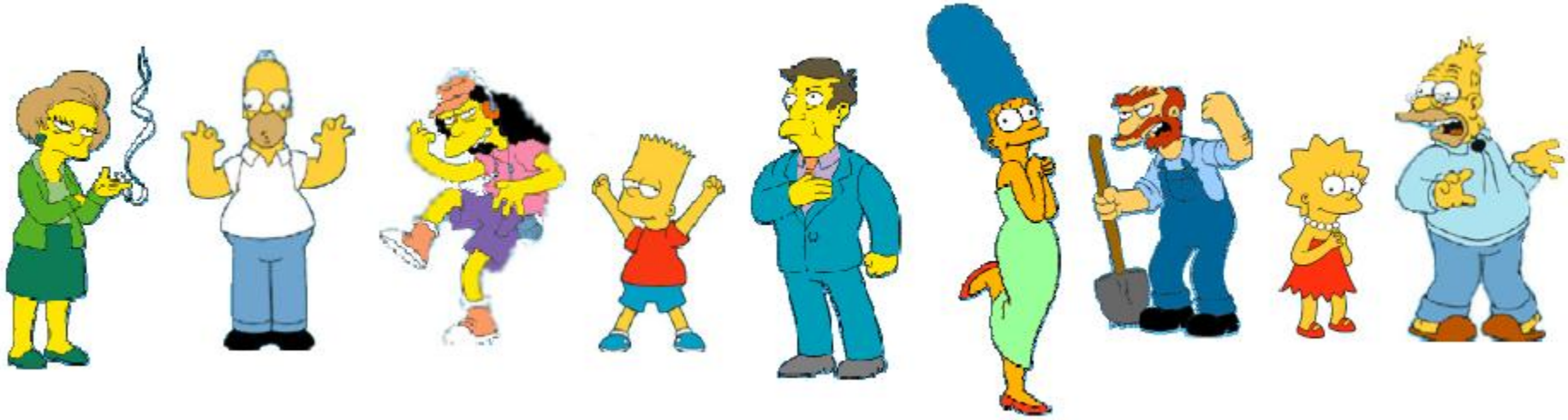


Proč nás to zajímá?

Dá se to použít pro něco konkrétního?

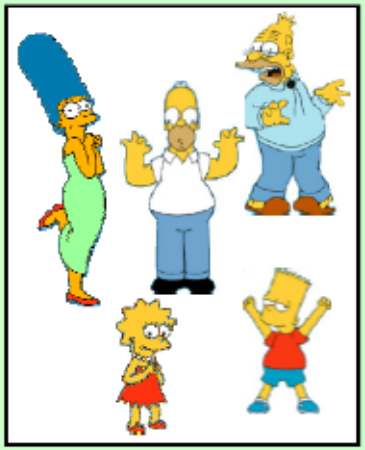
Důležité pro nalezení vnitřní struktury dat:
Mendělejevova tabulka, Clusty/Yippy pro lepší orientaci ve výsledcích webového vyhledávače, segmentace obrazů jako základ pro rozpoznávání objektů či definici hranic, …

Co jsou přirozené shluky z těchto individuí?



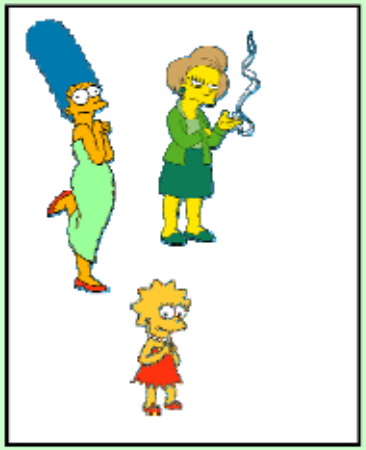
Co jsou „přirozené“ shluky z těchto individuí?

Pojem „přirozenost“ má zde velmi subjektivní charakter!



Rodina Simpsonů

Zaměstnanci školy



Ženy

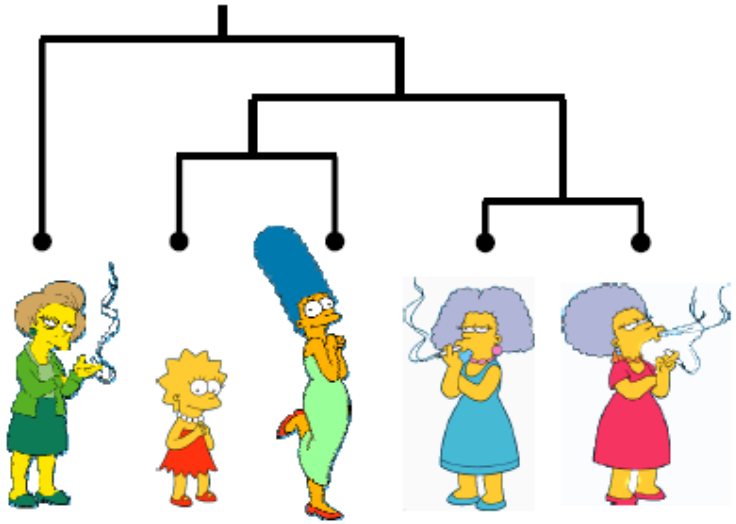
Muži

Dva přístupy ke tvorbě shluků

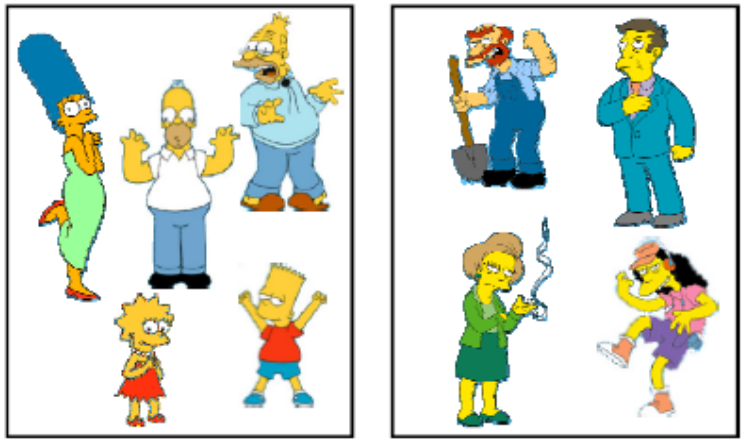
Shlukování rozkladem konstruuje různé rozklady množiny uvažovaných objektů a z těch vybírá nejvhodnější vzhledem k nějakému kritériu.

Hierarchické shlukování postupně sdružuje uvažované objekty podle zvoleného kritéria

Hierarchické shlukování



Shlukování rozkladem



Co je to podobnost?

Podobnost je obtížné definovat, ale lehce ji rozpoznáme, když ji vidíme!

- Jedná se o filosofickou otázku.
- Pragmatická charakteristika staví na definici **vzdálenosti**

Osnova přednášky

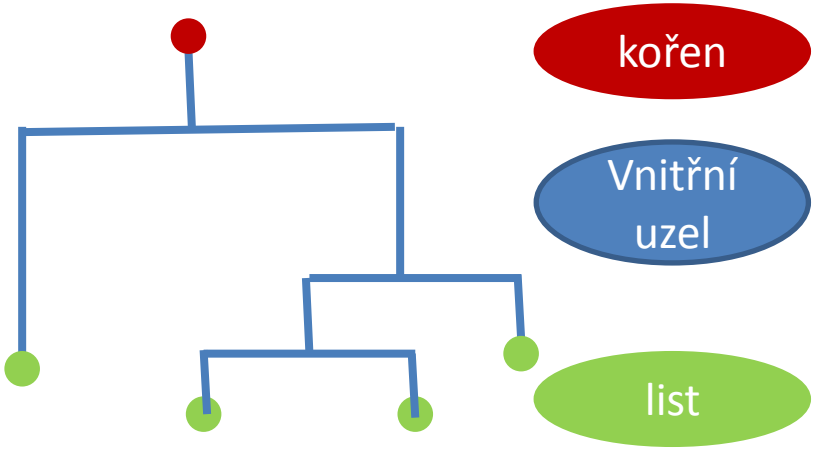
- Motivace
- **Míra vzdálenosti**
- Hierarchické shlukování
- Shlukování rozkladem
  - k-means (k-středů)
  - EM (expectation maximization) algoritmus, Gaussovská směs
  - Odhad počtu shluků

Jak definovat míru vzdálenosti?

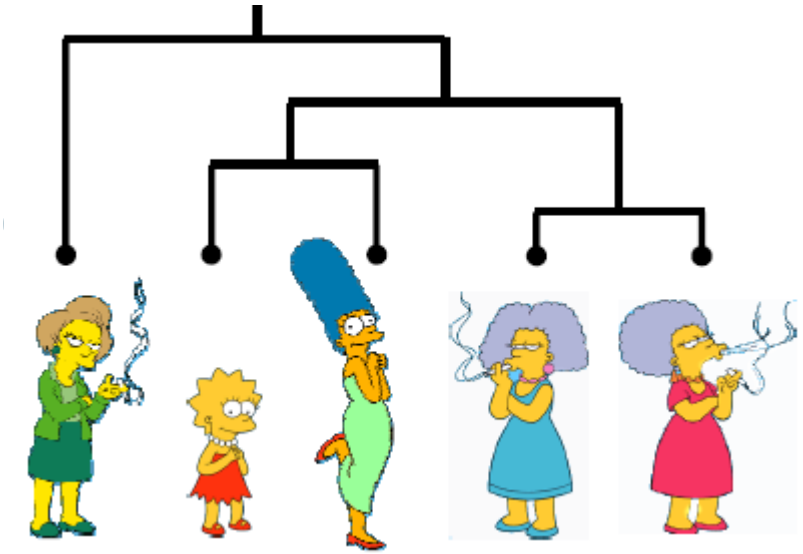
Funkce D, která každým 2 uvažovaným objektům o_1 a o_2 přiřazuje reálné číslo $D(o_1,o_2)$ tak, aby pro libovolné objekty **A**,**B** a **C** měla funkce D tyto vlastnosti

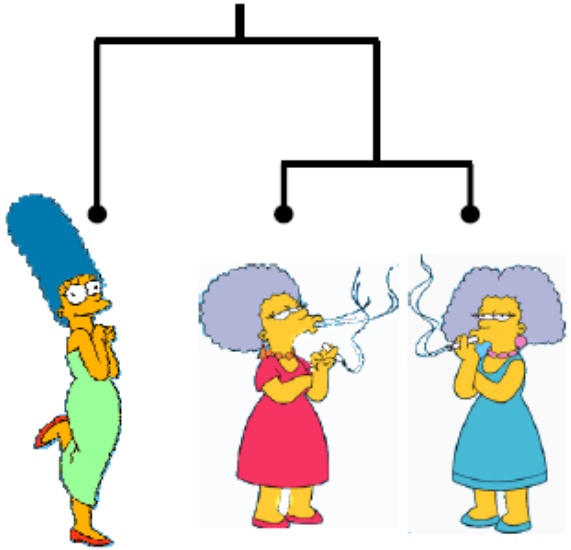
- $D(\mathbf{A},\mathbf{B}) = D(\mathbf{B},\mathbf{A})$ symetrie
- $D(\mathbf{c}) = 0$ iff $\mathbf{A} = \mathbf{B}$ konzistence samo-podobnosti
- $0 \leq D(\mathbf{A},\mathbf{B})$ pozitivita
- $D(\mathbf{A},\mathbf{B}) \leq D(\mathbf{A},\mathbf{C}) + D(\mathbf{C},\mathbf{B})$ trojúhelníková nerovnost

Dendrogram jako užitečný nástroj pro kompaktní znázornění vztahů podobnosti ve skupině



Podobnost 2 objektů ***a*** a ***b*** v dendrogramu je vyjádřena výškou (= vzdáleností od listu) nejnižšího uzlu, který leží jak na cestě od kořene k ***a*** i k ***b.***





Jaké vlastnosti by měl mít algoritmus pro tvorbu shluků?

- Škálovatelnost vzhledem k rozsahu dat (složitost v nárocích na čas i paměť)
- Nezávislost na pořadí vstupu dat
- Interpretovatelnost výsledků
- Schopnost vyrovnat se šumem a s „outliers“
- Schopnost pracovat s různými typy dat
- Schopnost využít omezující podmínky uživatele
- …

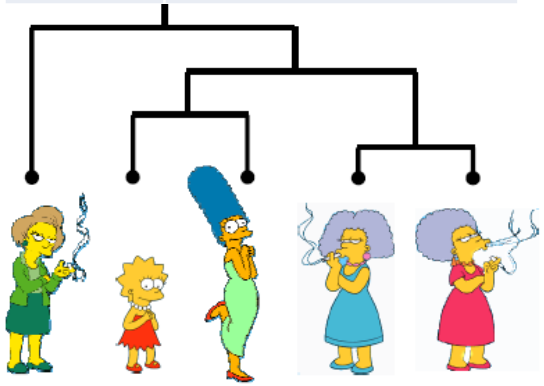
Osnova přednášky

- Motivace
- Míra vzdálenosti
- **Hierarchické shlukování**
- Shlukování rozkladem
  - k-means (k-středů)
  - EM (expectation maximization) algoritmus, Gaussovská směs
  - Odhad počtu shluků

Postupy pro hierarchické shlukování

Počet dendrogramů s ***n*** listy ? NP úloha → k řešení je nutná heuristika

# listů	# Dendrogramů
2	1
3	3
4	15
5	105
…	…
10	34 459 425



Postup zdola-nahoru (aglomerativní):

Začínáme od jednotlivých objektů, z nichž každý tvoří vlastní shluk. Najdeme 2 nejbližší shluky, které sloučíme. Proces opakujeme až do okamžiku, kdy všechny objekty jsou ve stejném shluku.

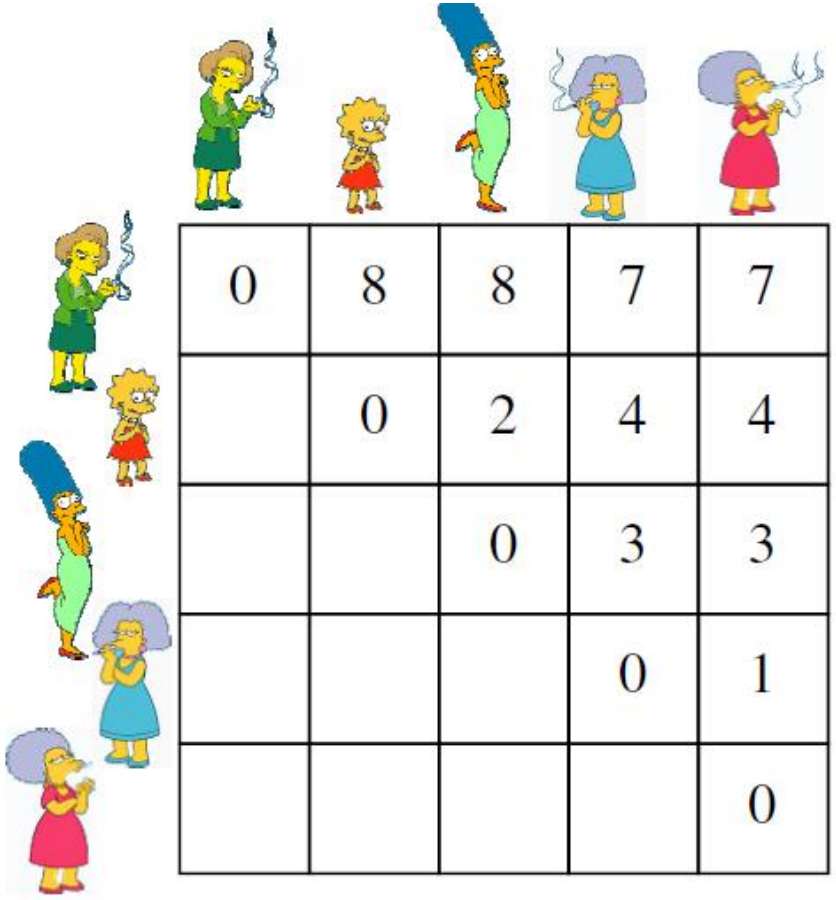
Postup shora-dolů (postupné dělení):

Začínáme od jediného shluku složeného ze všech data. Otestujeme všechny možnosti, jak shluk rozdělit na 2 disjunktní části a vybereme nejlepší variantu. Rekurzivně pokračujeme na obou vzniklých podmnožinách.

Předpokládejme, že máme k dispozici míru pro vzdálenost a údaje pro všechny páry, viz tabulka

$D($  $) = 8$

$D($  $) = 1$

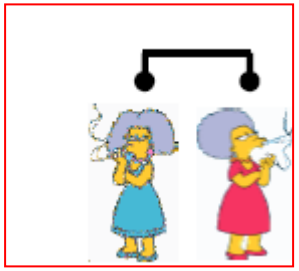


0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

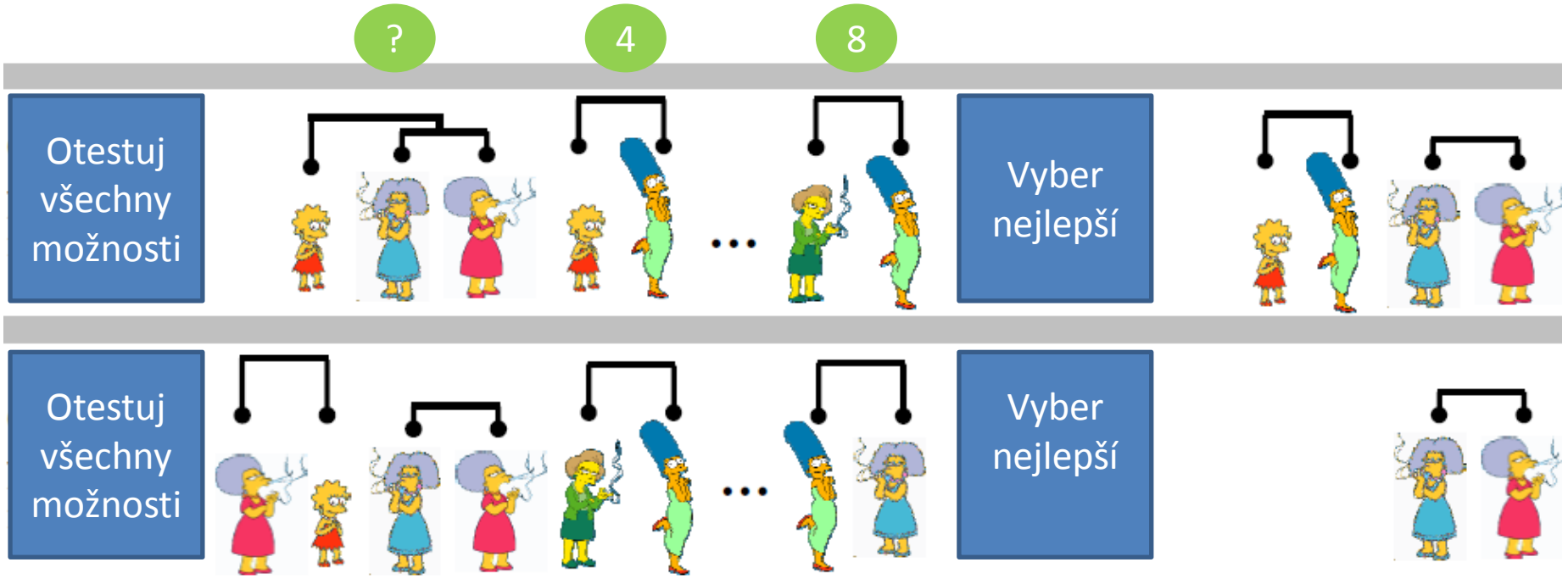
Aglomerativní postup zdola-nahoru: Simpsonovi-1

Otestujeme všech 5*4/2 párů a vybereme ten, jehož objekty jsou si nejblíž!

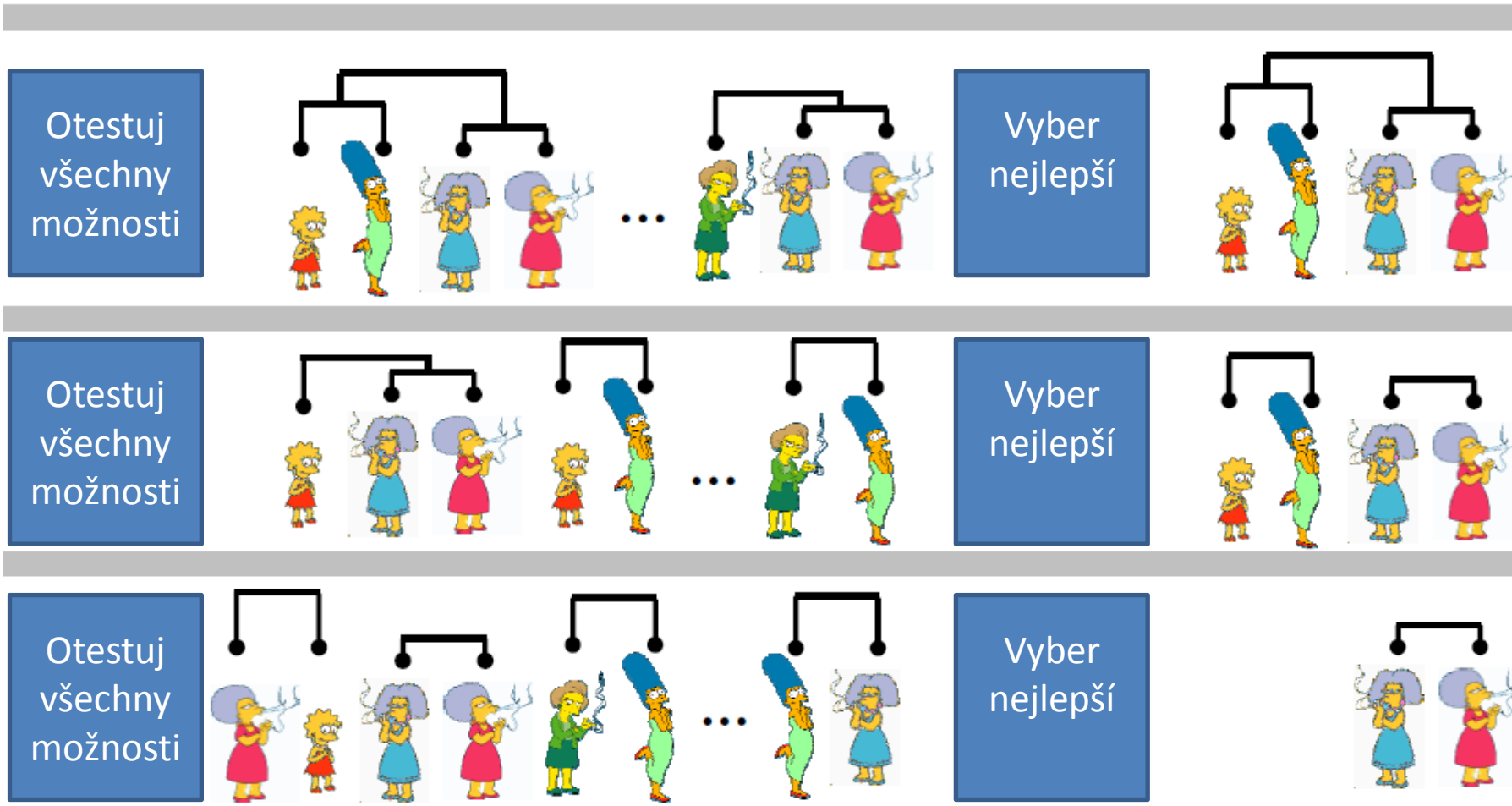
4 1 8 3



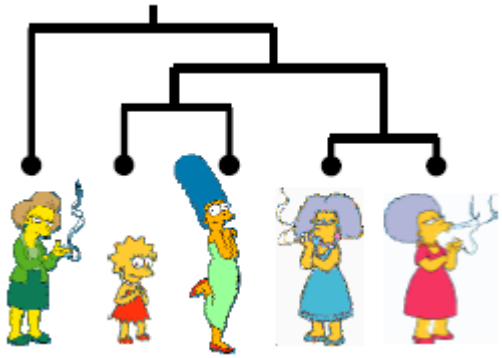
Simpsonovi-2



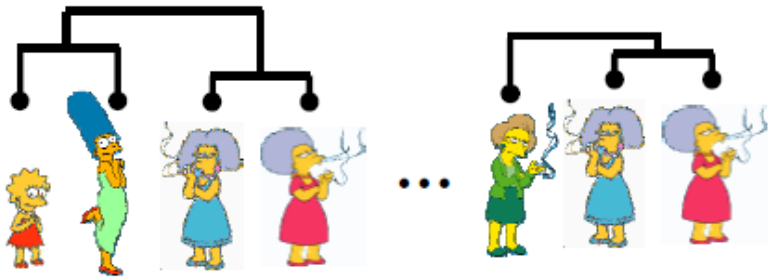
Simpsonovi-3



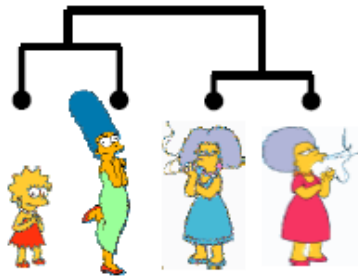
Simpsonovi-4



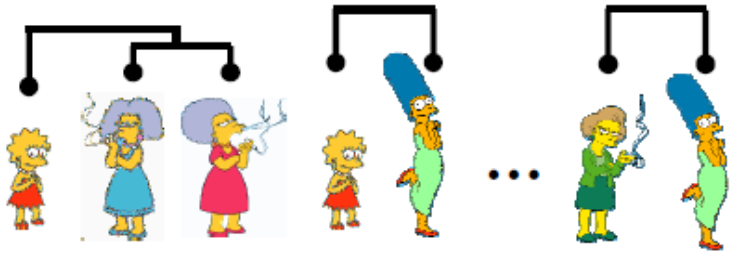
Otestuj všechny možnosti



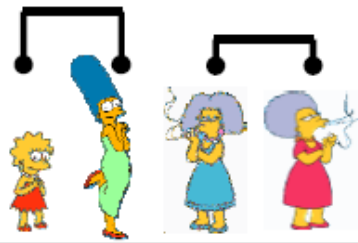
Vyber nejlepší



Otestuj všechny možnosti



Vyber nejlepší

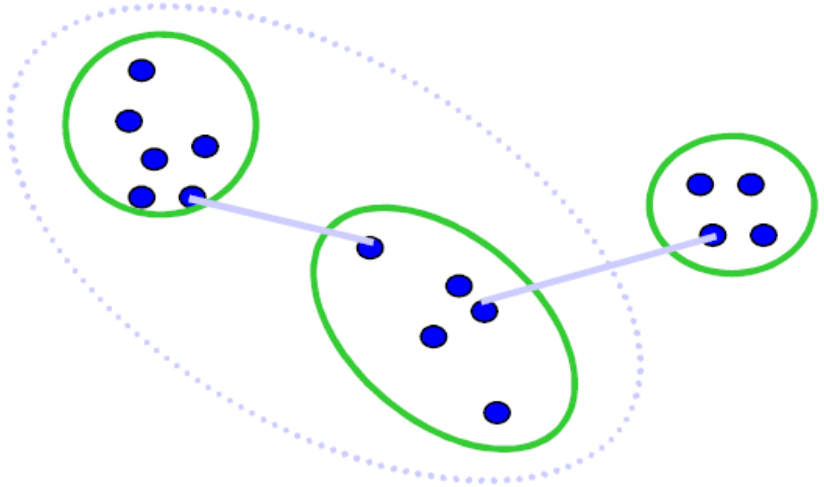


Otestuj všechny možnosti

Vyber nejlepší

Určování vzdálenosti 2 shluků -1

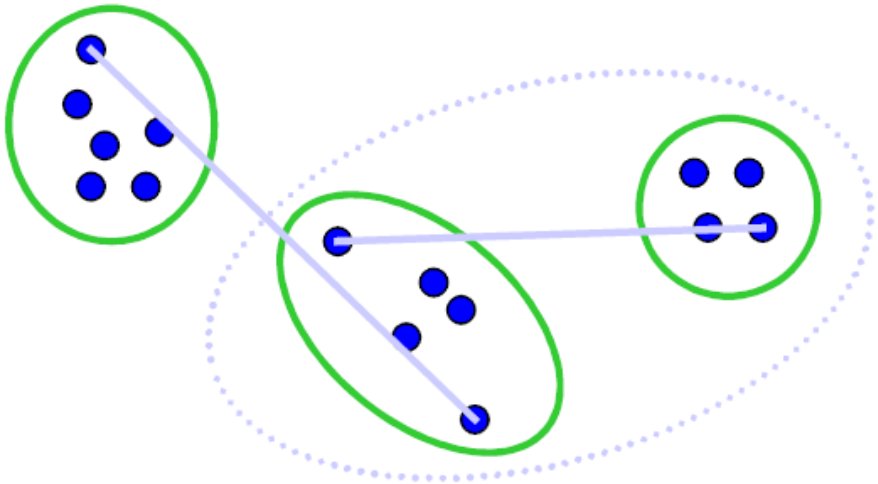
Vzdálenost shluků (*single link*) = vzdálenost jejich 2 **nejbližších prvků**



- Tato míra pro vzdálenost má tendenci k tvorbě řetízků menších shluků

Určování vzdálenosti 2 shluků - 2

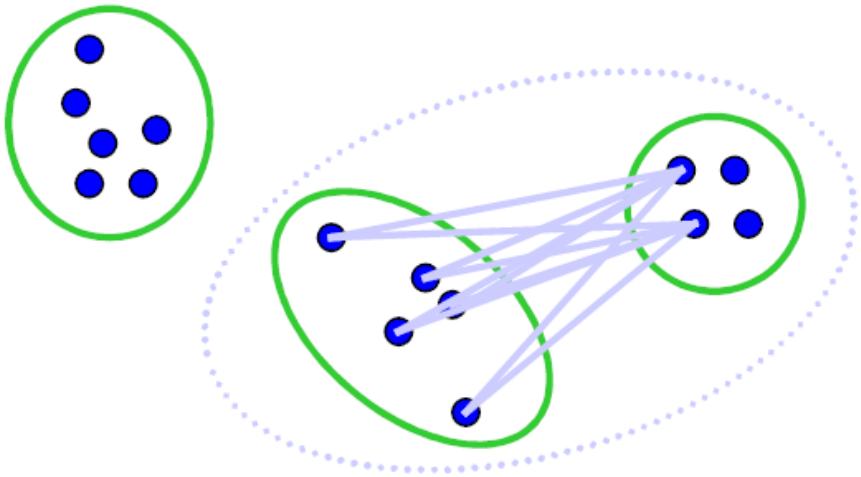
Vzdálenost shluků (complete link) = vzdálenost jejich 2 **nejvzdálenějších prvků**



- Tato míra pro vzdálenost obvykle tvoří poměrně kompaktní shluky

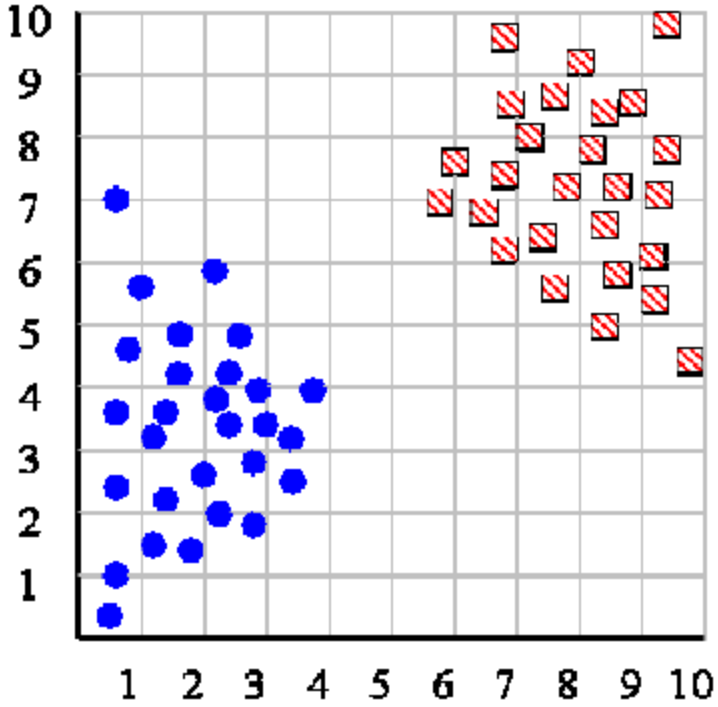
Určování vzdálenosti 2 shluků - 3

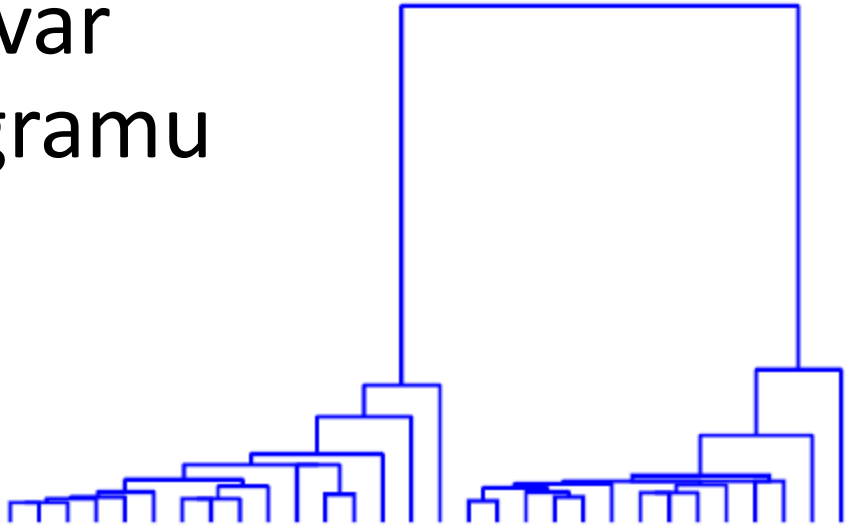
Vzdálenost shluků = průměrná vzdálenost mezi **všemi prvky obou shluků**



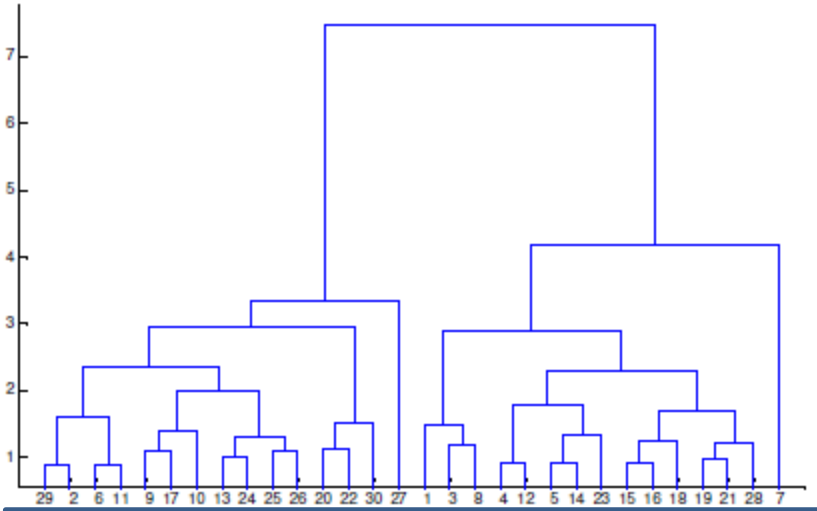
- Nejčastěji používaná míra pro vzdálenost.
- Robustní vůči šumu!

Vliv volby míry na tvar výsledný tvar dendrogramu





Single linkage



Average linkage

Shrnutí: typické vlastnosti hierarchických metod shlukování

- **Výhoda**: není třeba předem specifikovat počet shluků
- Hierarchickou strukturu dokáže uživatel často dobře interpretovat – odpovídá „intuici“, ovšem jde pak o subjektivní pohled!

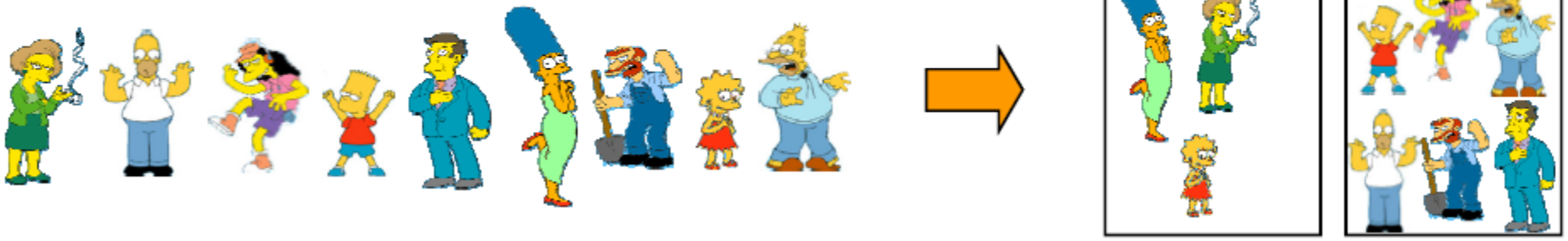
- Problém s rozsáhlými daty, neboť dolní odhad pro složitost shlukování je $\mathcal{O}(n^2)$, kde n je mohutnost shlukované množiny
- Nebezpečí uvíznutí v lokálním optimu

Osnova přednášky

- Motivace
- Míra vzdálenosti
- Hierarchické shlukování
- **Shlukování rozkladem**
  - k-means (k-středů)
  - EM (expectation maximization) algoritmus, Gaussovská směs
  - Odhad počtu shluků

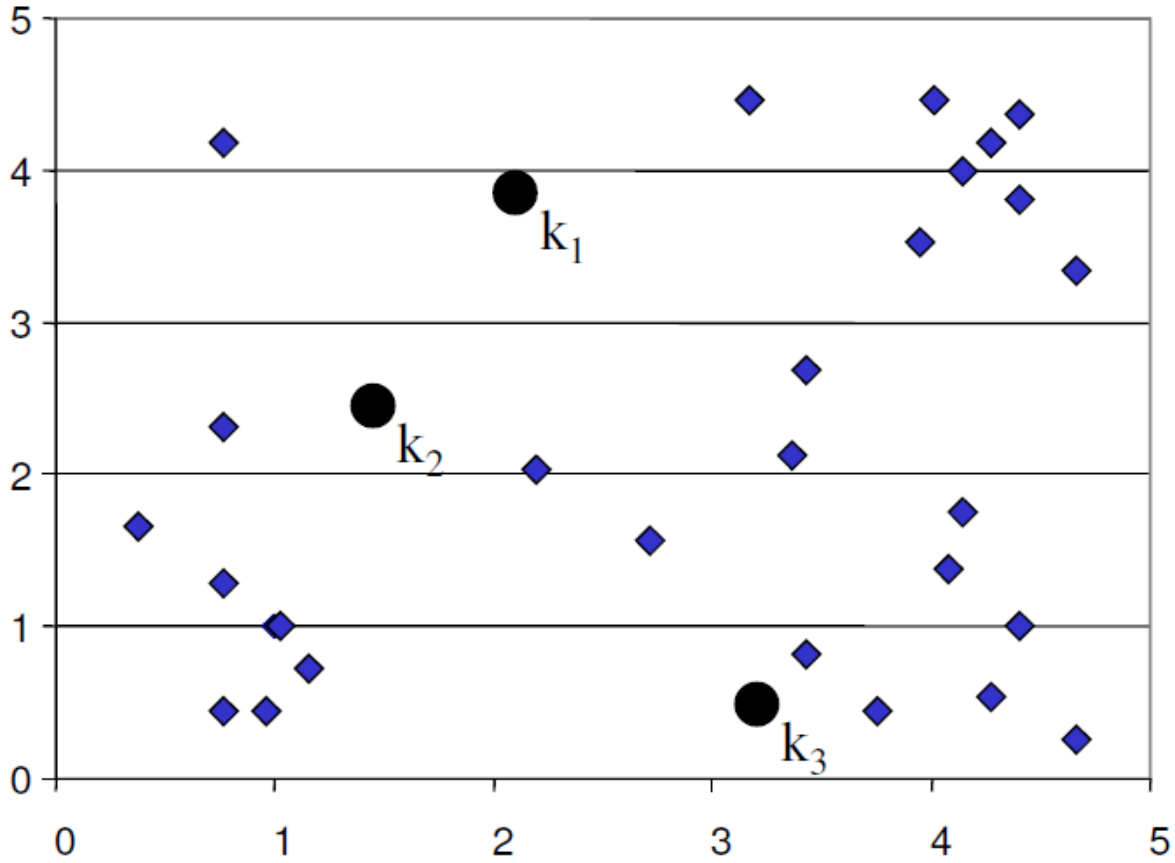
Shlukování rozkladem

- Nehierarchický postup, při němž se každý objekt vloží do jednoho z ***k*** disjunktních shluků.
- Předpokládá se, že uživatel předem stanoví ***k***, tj. požadovaný cílový počet shluků



K-means shlukování: inicializace

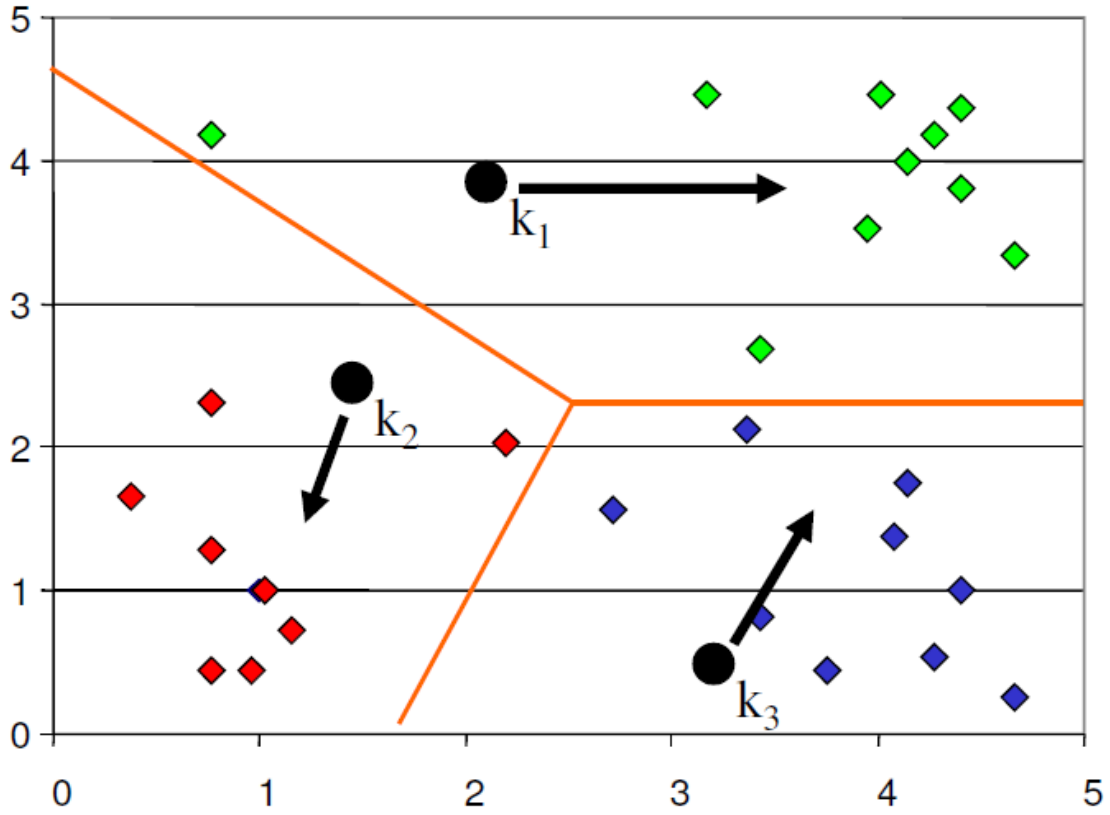
a. Stanov hodnotu **k** a vyber náhodně ***k*** bodů (jader) ve výchozí množině



K-means shlukování: iterační krok 1

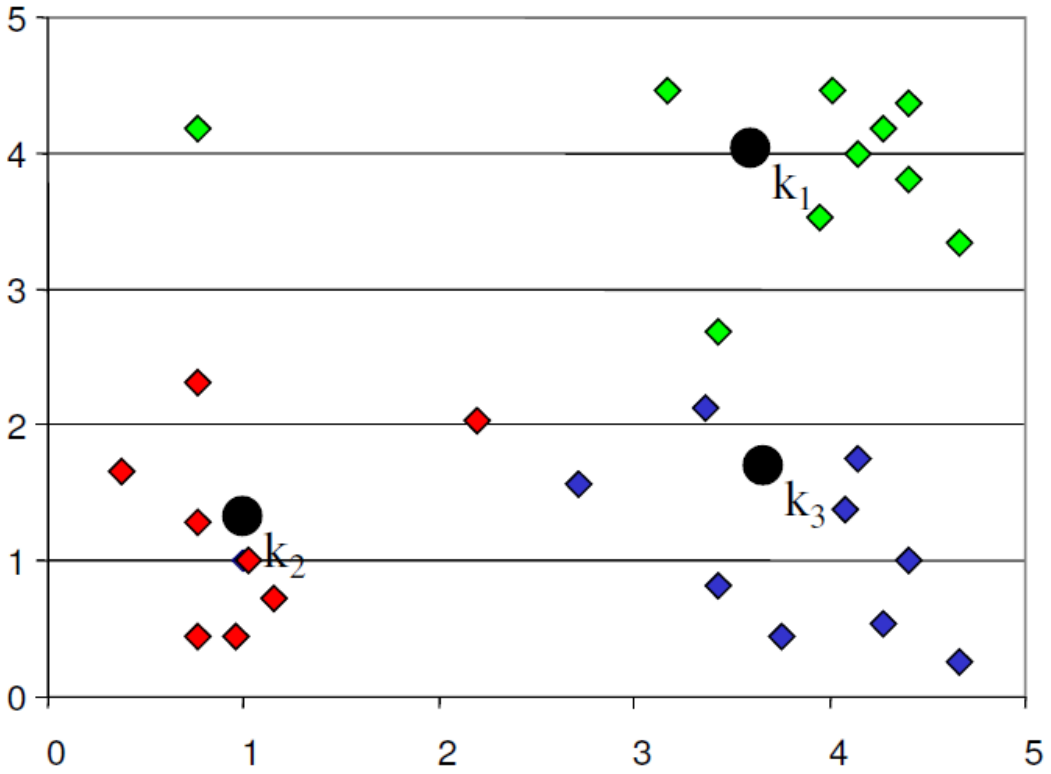
a. Přiřaď každý bod výchozí množiny k **nejbližšímu** z vybraných ***k*** jader.

b. V každé ze vzniklých ***k*** množin bodů nadefinuj **nové jádro** jako „průměr“ všech prvků této množiny



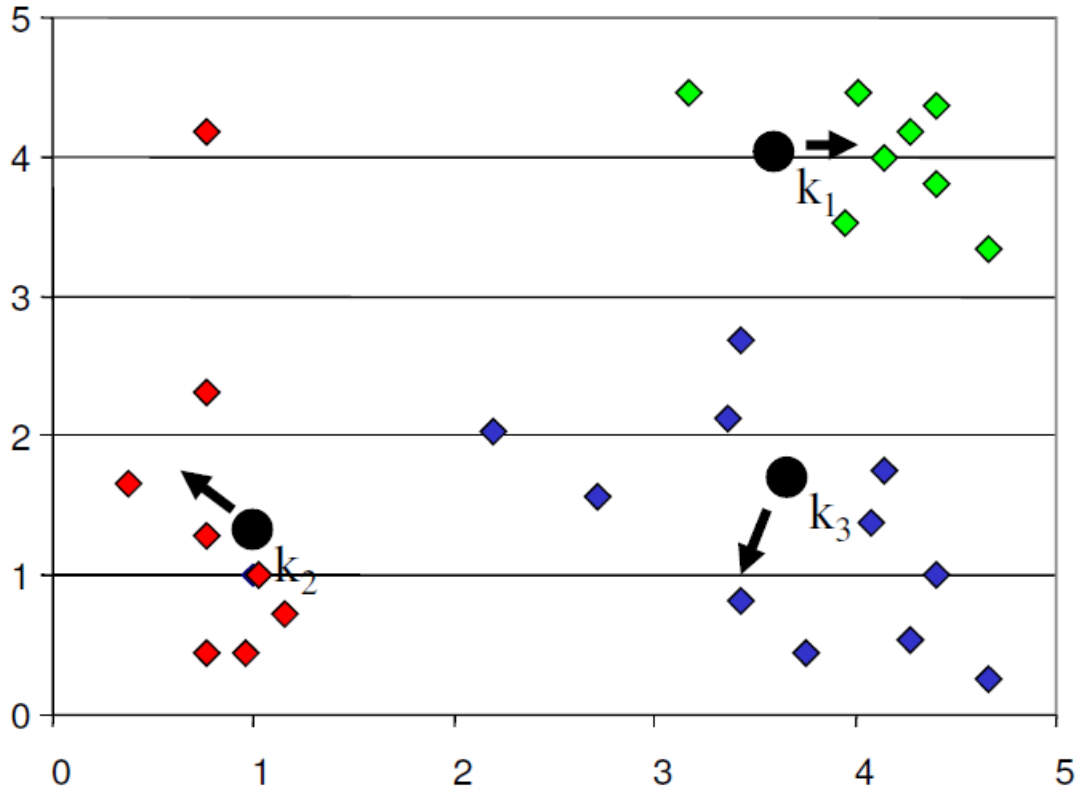
K-means shlukování: iterační krok 2

a. Pro každý bod původní množiny proveď přiřazení k nejbližšímu jádru z těch, která byla nadefinovaná v předchozím kroku.
b. V každé ze vzniklých ***k*** množin bodů nadefinuj **nové jádro** jako „průměr“ všech prvků této množiny



K-means shlukování: iterační krok 2

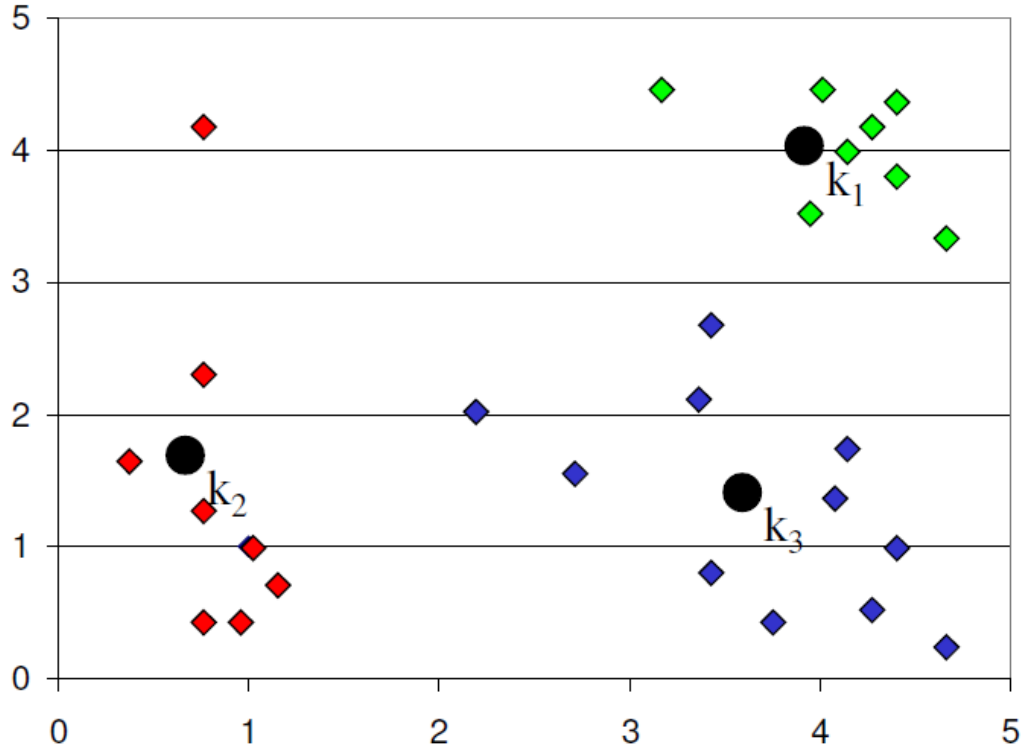
a. Pro každý bod původní množiny proveď přiřazení k nejbližšímu jádru z těch, která byla nadefinovaná v předchozím kroku.
b. V každé ze vzniklých ***k*** množin bodů nadefinuj **nové jádro** jako „průměr“ všech prvků této množiny



K-means shlukování: kriterium pro ukončení

a. Opakuj iterační krok až do té doby, než

„Při iteraci nedojde ke **změně zařazení do shluku** pro žádný prvek původní množiny.“



k-means algoritmus pro ***N*** objektů

1. Stanov požadovaný počet ***k*** shluků.
2. Vyber náhodně výchozích ***k*** jader.
3. Přiřaď každému z ***N*** objektů číslo shluku, které odpovídá číslu nejbližšího jádra.
4. Předefinuj pozice jader všech ***k*** shluků tak, že bude použit průměr hodnot prvků v daném shluku.
5. Opakuj kroky 3. a 4. až do té doby, že se příslušnost do shluků stabilizuje (po iteraci není žádný objekt zařazen do jiného shluku než před ní).

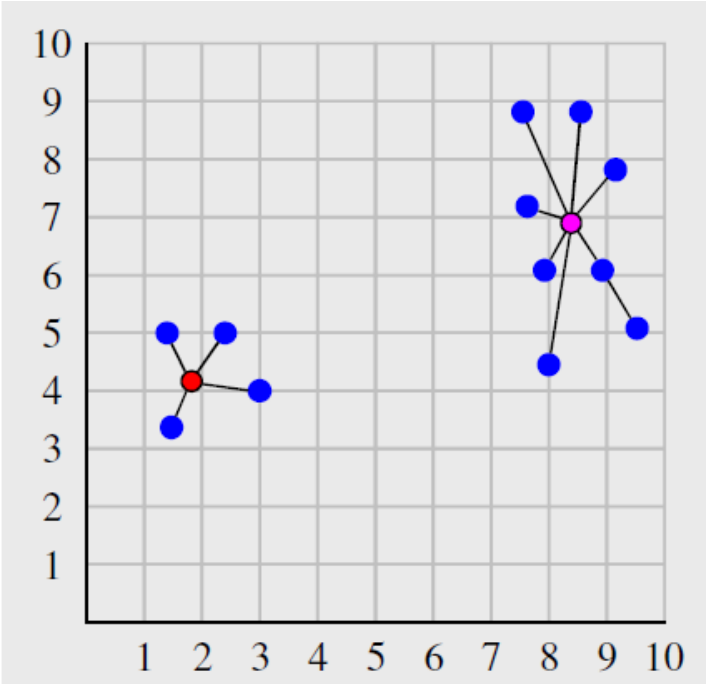
Proč algoritmus *k*-means funguje?

- **Předpoklad:** Dobré shlukování zajišťuje vysokou podobnost uvnitř shluku.
- ***k***-means minimalizuje průměrnou vzdálenost mezi prvky téhož shluku vypočtenou jako

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} \left\| x_{ki} - x_{kj} \right\|^2$$

- Tato hodnota se rovná 2x suma vzdáleností ke středům jednotlivých shluků, čili výsledné střední chybě

$$\sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$



Shrnutí vlastností *k*-means algoritmu

- **Výhody**
  - Jednoduchý s lehkou implementací i laděním.
  - Intuitivní objektivní funkce, která optimalizuje podobnost uvnitř shluků.
  - *Poměrně efektivní*: složitost $\mathcal{O}(t\,k\,n)$, kde n je počet objektů, k je počet shluků a t počet iterací. Obvykle bývají hodnoty t a $k \ll n$.
- **Nevýhody**
  - Použitelné, jen tam, kde *umíme spočítat průměr*. Co kategorická data?
  - Velmi záleží na inicializaci – nebezpečí uvíznutí v *lokálním minimu*.
  - Požaduje se znalost *počtu shluků*.
  - Nevhodné pro zašuměná data s *výjimkami* (outliers).
  - Nevhodné pro situace, kdy *shluky nemají kulový tvar*

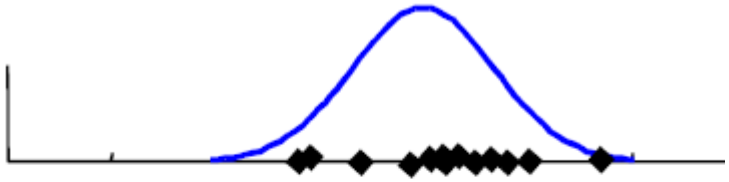
Osnova přednášky

- Motivace
- Míra vzdálenosti
- Hierarchické shlukování
- Shlukování rozkladem
  - k-means (k-středů)
  - **EM (expectation maximization) algoritmus, Gaussovská směs**
  - Odhad počtu shluků

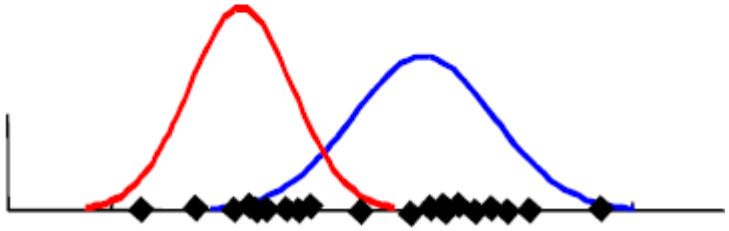
Jednorozměrný model typu **GMM** „Gaussovská směs“

- Gaussian

$$P(x) = \varphi(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$$



  - např. výška populace



- Gaussovská směs

$$P(C=i) = \omega_i, \quad P(x \mid C=i) = \varphi(x;\mu_i,\sigma_i)$$

$$P(x) = \sum_{i=1}^{K} P(x, C=i) = \sum_{i=1}^{K} P(C=i)P(x \mid C=i) = \omega_i\varphi(x;\mu_i,\sigma_i)$$

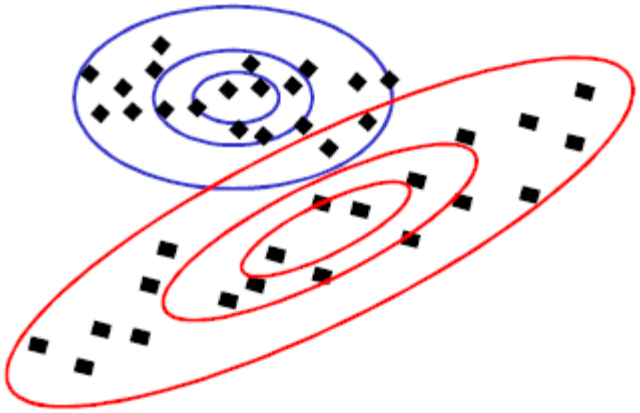
  - např. výška 2 různých populací

Vícerozměrný model typu **GMM** „Gaussovská směs“

- Směs vícerozměrných Gaussiánů

$$P(C=k)=\omega_i, \quad P(x \mid C=i)=\varphi(x;\mu_i,\Sigma_i)$$

  - Např. pro situaci, kdy **y** je hodnota krevního tlaku a **x** je věk



GMM + EM = „Soft k-means“

1. Stanov požadovaný počet ***k*** shluků.
2. Vyber náhodně výchozích ***k*** jader.
3. **E-krok**: přiřaď pravděpodobnostní hodnotu příslušnosti ke shlukům

   $p_{ij} = \boldsymbol{P}(C = i \mid \boldsymbol{x}_j) = \alpha\, \boldsymbol{P}(\boldsymbol{x}_j \mid C = i)\, \boldsymbol{P}(C = i)$

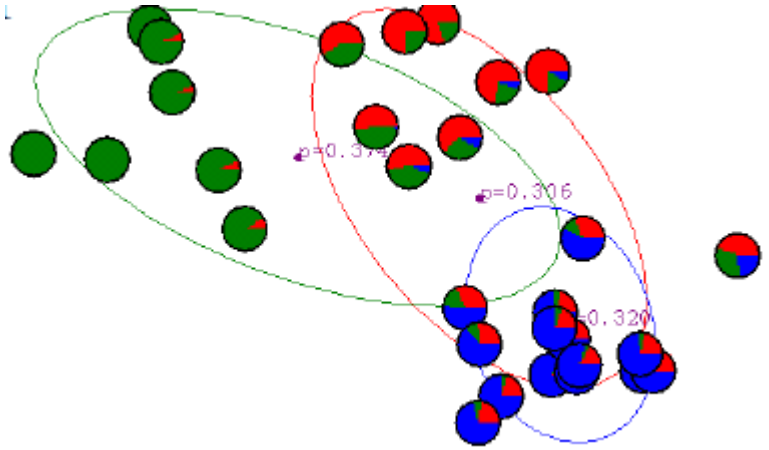
   $p_i = \sum_{\boldsymbol{j}} p_{ij}$
4. **M-krok:** proveď nové odhady parametrů s využitím právě vypočtených hodnot. / $^{\mathsf{T}}$

   $\mu_i \leftarrow \sum_{\boldsymbol{j}} p_{ij} \boldsymbol{x}_j / p_i$

   $\sum_{\boldsymbol{j}} \leftarrow \sum_{\boldsymbol{j}} p_{ij} \boldsymbol{x}_j \boldsymbol{x}_j^{\mathsf{T}} / p_i$

   $\omega_i \leftarrow p_i$

4. Opakuj kroky 3. a 4. až do té doby, kdy změny všech parametrů jsou menší než zvolená hranice.



Hodnocení GMM

Výhody

- Interpretovatelnost: vzniká dokonce generativní model!
- Efektivita srovnatelná s ***k***-means
- Intuitivní objektivní funkce
- Lze zobecnit i pro směsi různých typů dat:
  - Kategorická data
  - Místo průměru lze použít např. max.
  - Citlivost na šum a výjimky záleží na distribuční funkci

Nevýhody

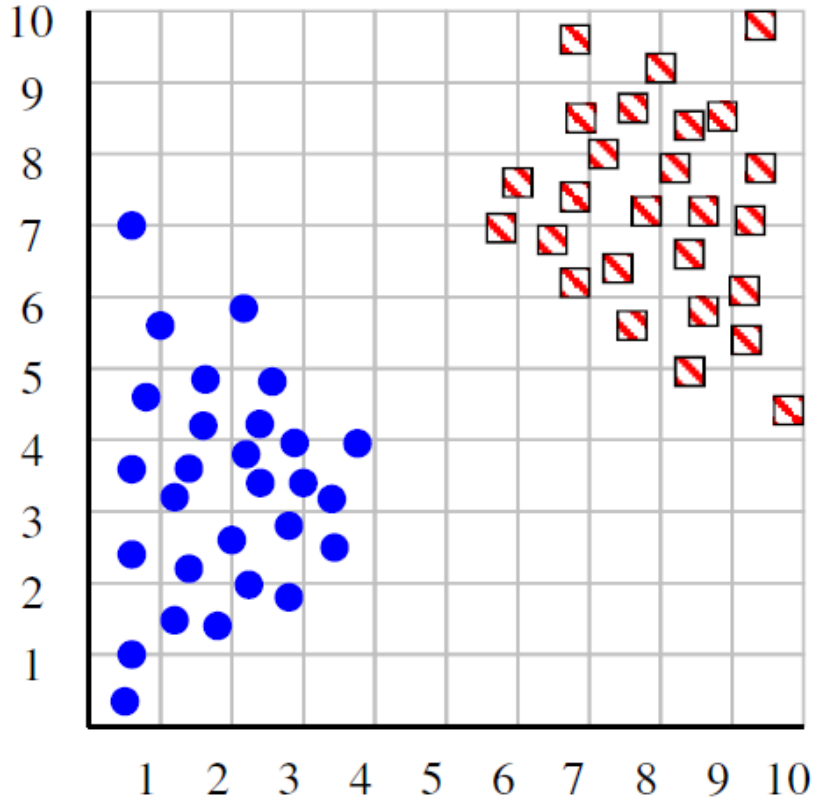
- Často uvázne v lokálním optimu – vliv inicializace!
- Je nutné správne zvolit hodnotu ***k.***
- Nevhodné v případě, že shluky nejsou konvexní!
- $\mathcal{O}(n^3)$

Osnova přednášky

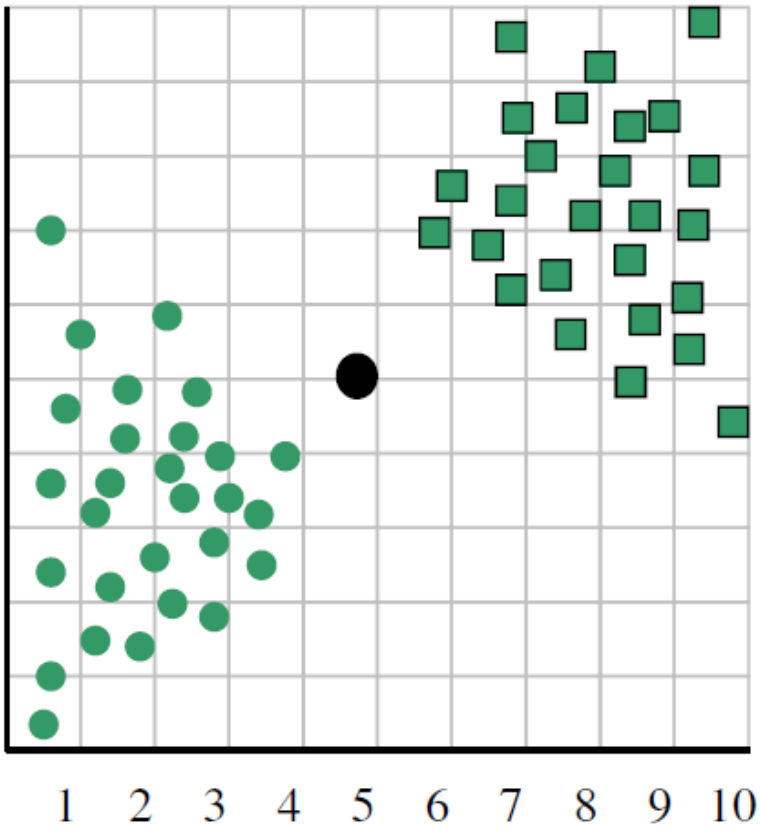
- Motivace
- Míra vzdálenosti
- Hierarchické shlukování
- Shlukování rozkladem
  - k-means (k-středů)
  - EM (expectation maximization) algoritmus, Gaussovská směs
  - **Odhad počtu shluků**

Jak se rozhodneme pro správný počet shluků?

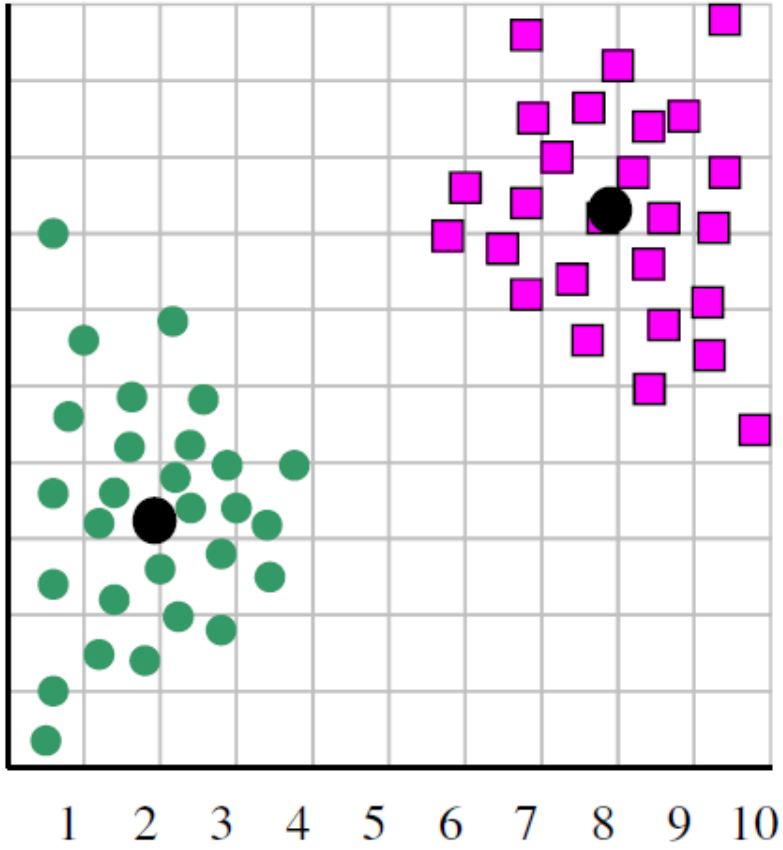
- Obecně je to otevřený problém.
- Používá se řada heuristik
- Jedna z nich srovnává hodnoty objektivní funkce (celkový součet vzdáleností) pro různé volby počtu shluků



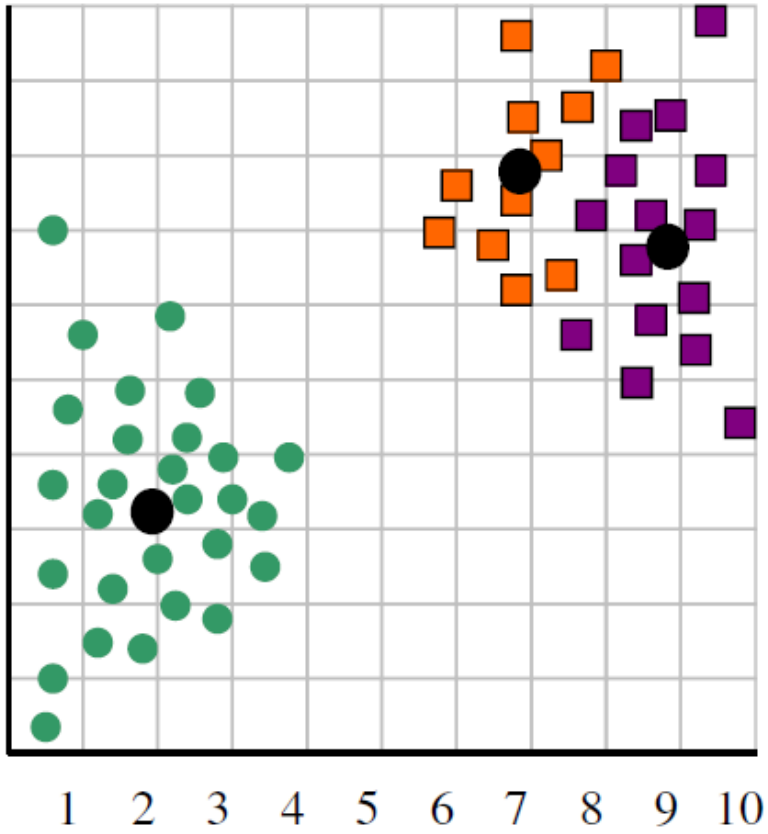
- Je-li ***k*** = 1, je výsledná hodnota obj. funkce 873



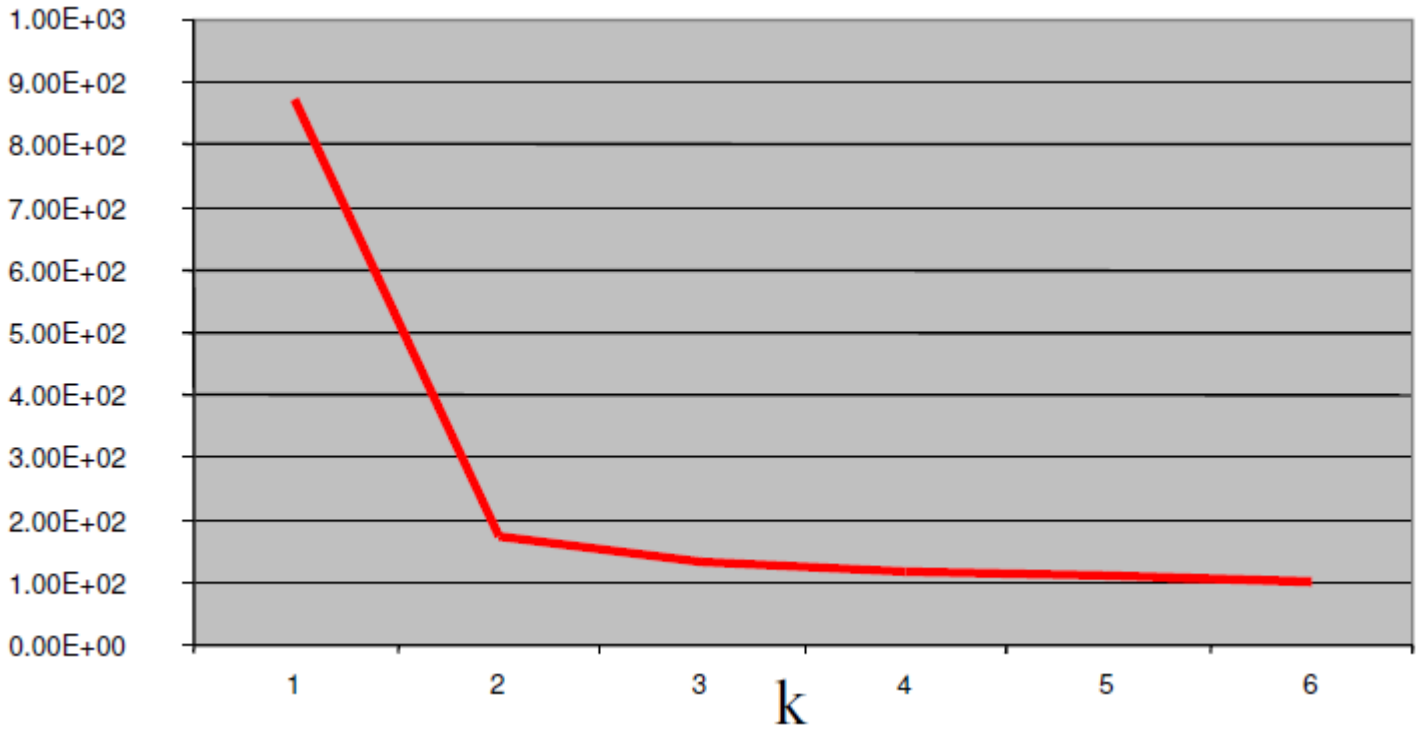
- Je-li ***k*** = 2, je výsledná hodnota obj. funkce 173,1



- Je-li ***k*** = 3, je výsledná hodnota obj. funkce 133,6



- Sledujme hodnotu objektivní funkce pro ***k***= 1,2,..,6
- Náhlý pokles pro ***k*** = 2 svědčí pro volbu 2 shluků.
- Obecně hledáme „prudký ohyb“ (knee/elbow finding)



Pozor! Průběh obj. funkce obvykle není tak jednoduchý jako v tomto jednoduchém příkladě.