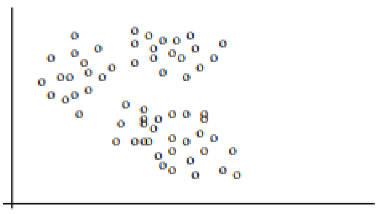
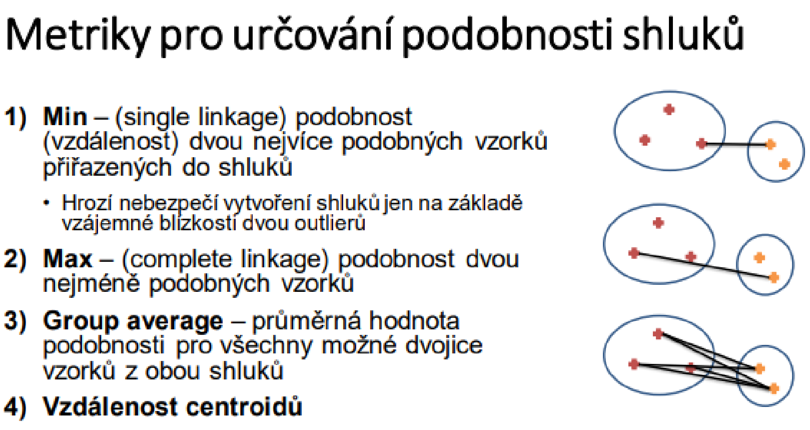
17. Metody učení bez učitele, shlukování- algoritmus K-means a LGB, hierarchické shlukování.

Metody učení bez učitele

- Algoritmy se učí na základě dat, u kterých nejsou člověkem připraveny žádné značky
  - Neučí se klasifikovat
  - Neučí se ani predikovat
- Algoritmy bez učitele místo toho hledají **vnitřní strukturu dat**
  - Ani tato struktura není ale předem označkována
- Cílem je rozdělit data do několika skupin, resp. shluků (angl. clusters)
- **Musí přitom platit, že:**
  - Data uvnitř jednoho shluku jsou si vzájemně podobná
  - Data uvnitř jednoho shluku se liší od dat ve všech ostatních shlucích

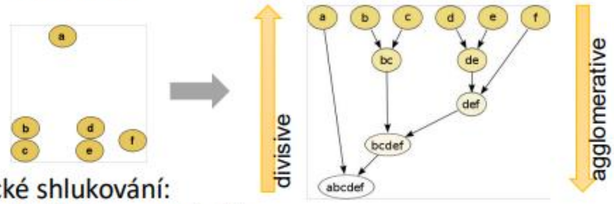


Pro měření vzdálenosti lze využít běžné vzdálenostní metriky (Euklid, L1..)



Metody shlukování

- Hierarchické shlukování:



- Nehierarchické shlukování:
  - Samoorganizující neuronové sítě
  - K-Means

shlukování- algoritmus K-means a LGB

K-means

Algoritmus K-průměrů (K-means) - princip

- Pro zvolené číslo K se hledá **rozklad trénovací množiny** na K podmnožin (shluků) tak, že každý $j-tý$ shluk má svého reprezentanta (centroid) $\boldsymbol{\mu}_j$ a je charakterizován součtem vzdáleností mezi ostatními prvky shluku a centroidem.
- **Kritériem** vhodnosti rozkladu je **součet všech dílčích vzdáleností** přes všechny shluky.

$$J = \sum_{j=1}^{K} \sum_{i=1}^{N} \left\| \boldsymbol{x}_i^{(j)} - \boldsymbol{\mu}_j \right\|$$

- Toto kritérium se snažíme **minimalizovat**:

Algoritmus K-průměrů (K-means)

Algoritmus je založen na **iteračním postupu**:

1. Zvolíme K prvků TrM jako prvotní odhady centroidů.
2. Zařadíme každý prvek TrM do jedné z K skupin na základě nejmenší vzdálenosti k centroidu.
3. Pro každou skupinu vypočteme nový centroid tak, aby měl nejmenší vzdálenost ke všem prvkům skupiny.
4. Vyhodnotíme celkové kritérium a v případě, že se jeho hodnota liší od předchozí hodnoty o méně než •, iterační proces zastavíme, jinak návrat na krok 2.

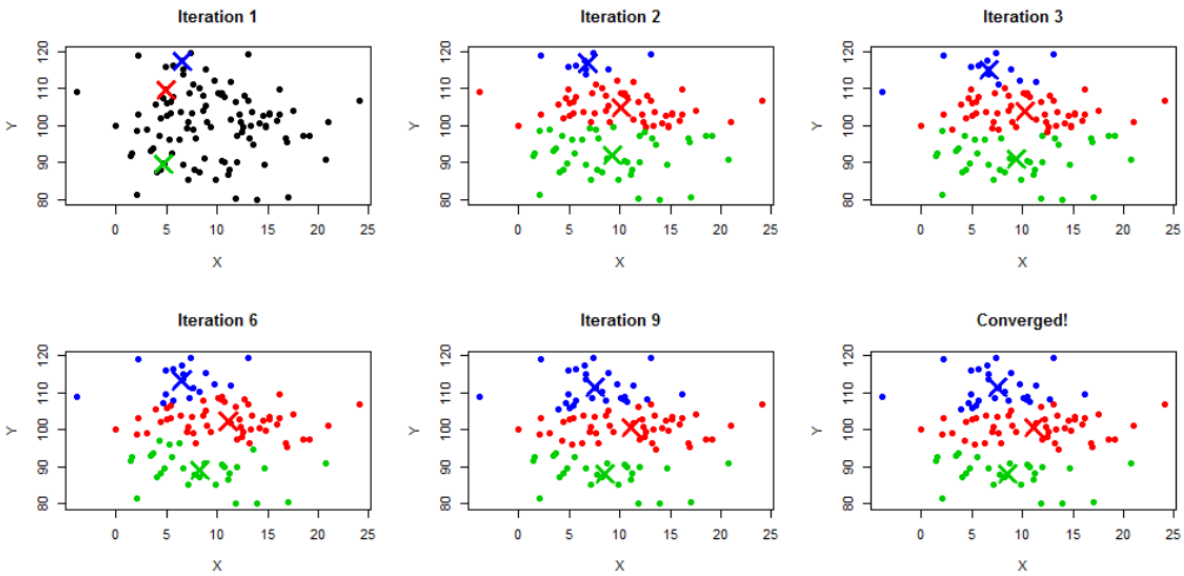
Silné stránky:

• Výpočetní náročnost je O(TKN) (kde N je počet dat, K je počet shluků, T je počet iterací

• Protože K a T je obvykle malé, je K-means považován za lineární

Slabé stránky:

- Není zajištěno dosažení globálního minima kriteriální funkce
- Výsledné centroidy mohou záviset na volbě počátečních centroidů
- Proces lze opakovat s různými počát. centroidy a vybrat pak to řešení, které má minimální hodnotu kriteriální funkce
- Náchylný na přítomnost outlierů
- Lze částečně eliminovat – viz dále
- Nefunguje pro všechna rozložení příznaků
- Na shlukovaných datech je nutné umět spočítat střední hodnotu
- Algoritmus neřeší otázku nejvhodnějšího čísla **K**



Algoritmus LGB

Algoritmus LBG (Linde, Buzo, Gray)

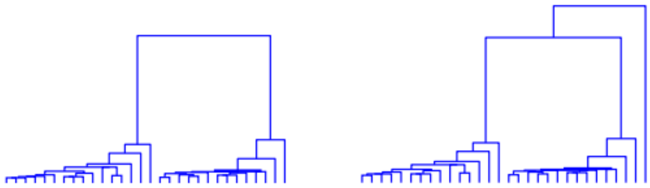
Na rozdíl od K-means řeší též nalezení čísla K

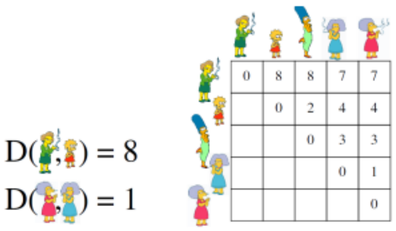
- Dvojnásobně iterační procedura (pro K, i pro určení centroidů)
  1. **Inicializace:** Nastav K = 1. Najdi centroid.
  2. **Rozdělení** ($K = 2K$): Pro každou dosavadní skupinu urči dva nové počáteční centroidy.
  3. **Nalezení nových centroidů:** S celou TrM a s číslem K proveď K-means alg. Urči hodnotu kriteriální funkce.
  4. **Ukončení:** Je-li dosaženo cílové číslo K nebo se hodnota kriteriální funkce již významně nemění, skonči, jinak zpět na krok 2.

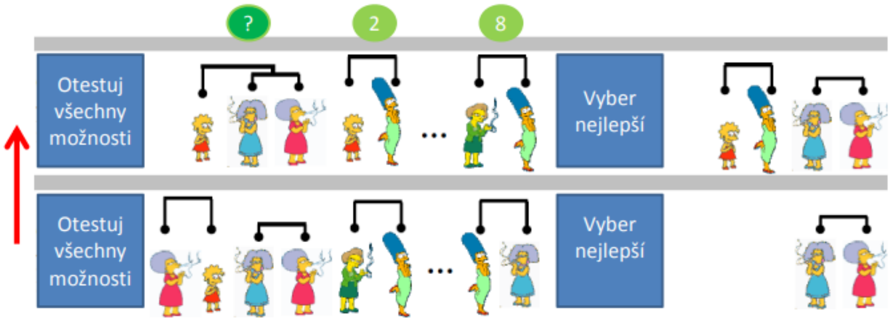
Pozn. V kroku 2 je také možné rozdělit pouze největší shluk. Pak $K = K + 1$

Hierarchické shlukování

- Jednotlivé kroky procesu shlukování je možné zaznamenat pomocí tzv. dendogramu
- Dendogram reflektuje míru podobnosti jednotlivých shluků, umožňuje tak odhadnout optimální počet shluků; příp. identifikovat tzv. outliery
- Nevýhodou je, že data jednou zařazená do shluku již nemohou být vyňata





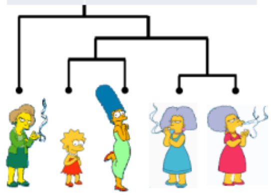


Hierarchické shlukování: nejdůležitější vlastnosti

Označme ***d*(*m*)** počet všech různých dendrogramů s ***m*** listy (pro ***m*** objektů). Jistě platí, že

$d(m+1) = (m+1)*m\ *d(m)\ /2$

# listů	# Dendrogramů
2	1
3	3
4	18
5	180
…	…
10	Asi 40 000 000



d(*m*) má složitost vyšší než $\mathcal{O}(m\,!)$. Prohlédnutí všech kandidátů je NP úloha → k řešení je nutná **heuristika** (např. *„aglomerativní postup vede k cíli "*)

Postup zdola-nahoru (aglomerativní):

Najdeme 2 nejbližší shluky, které sloučíme.

Začátek: každý objekt tvoří vlastní shluk.

Postup: Proces opakujeme až do okamžiku, kdy všechny objekty jsou ve stejném shluku.

Postup shora-dolů (postupné dělení):

Začátek: jediný shluk tvořený všemi daty. Otestujeme všechny možnosti, jak shluk rozdělit na 2 disjunktní části a vybereme nejlepší variantu.

Postup: Rekurzivně pokračujeme na obou vzniklých podmnožinách.