Učení bez učitele

Klastrování, Analýza nákupního koše

Vzdálenost – pro klastrování klíčová a aplikačně závislá

- Na volbě vzdálenosti $d(x_i, x_{i'})$ závisí výsledek,
- volba vzdálenosti nemá jasná kriteria, závisí na aplikaci – datech a prioritách uživatele.
- Jedna možnost je vzdálenost definovat přímo – maticí, která má na diagonále nuly, měla by být symetrická a pro dost algoritmů splňovat trojúhelníkovou nerovnost $d_{ii'} \leq d_{ik} + d_{i'k}$.
- Nebo definujeme vzdálenost pro každý atribut
  - celková vzdálenost bude (vážený) součet vzdálenosti v atributech,
  - rovnocenné atributy dělají váhy $w_j = \frac{1}{\hat{d}_j}$, nikoli $w_j = 1$, kde $\hat{d}_j = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_j(x_{ij}, x_{i'j}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} (x_{ij} - x_{i'j})^2$.

Klastrování k–medoids (reprezentanti z dat)

1. náhodně zvol K příkladů z dat, které budou určovat klastry $C(.)$.
2. Pročítej data, dokud se mění přiřazení bodů ke klastrům
   (a) každý bod zařaď k nejbližšímu klastru
   (b) spočti nový centroid, tj. příklad i_k^* daného klastru k, minimalizující součet vzdáleností v klastru

$$i_k^* = argmin_{\{i:C(i)=k\}} \sum_{C(i')=k} d(x_i, x_{i'})$$

- Složitější na výpočet než k–means, protože potřebuje kvadratický čas.
- Výhoda je, že vzdálenost nemusí být kvadratická, ale libovolná – např. počet rozdílů u atributů typu ANO/NE.
- Použitelné i v případě, že máme jen vzdálenosti, ne prostor atributů.

Odbočka – zobrazení
Multidimensional Scaling

- Chtěli bychom zobrazit data "rozdílnosti zemí", známe jen vzdálenosti, ne metrický prostor.
- Snažíme se zachovat vzdálenosti dvojic (*least squares scaling*).
- Zvolíme počet dimenzí k.
- Hledáme $z_1, \ldots, z_N \in R^k$ minimalizující stressovou funkci

$$S_D(z_1, \ldots, z_N) = \left[\sum_{i \neq i|} (d_{ii|} - ||z_i - z_{i|}||)^2 \right]^{\frac{1}{2}}.$$

- Řeší se gradientně.

Je mnoho jiných způsobů, např. klasické scaling s vektorovým součinem, které má přímé řešení vlastními vektory, ale není ekvivalentní s tímto, vzdálenosti neurčují jednoznačně vektorový součin ani počátek souřadnic.

Opakování k–means (průměry)

Předem zvolíme počet klastrů K a
náhodně zvolíme K bodů, které zvolíme za počáteční středy klastrů.

Dokud se mění přiřazení bodů ke klastrům, iterujeme:

1. **každý bod zařadíme k nejbližšímu klastru a**
2. **spočteme nové středy = průměry bodů v jednotlivých klastrech.**

Příklad použití: vektorová kvantizace ve zpracování obrazu.

Počet klastrů

- Je–li dán, není co řešit.
- Součet vzdáleností instancí uvnitř K klastrů:

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'})$$

  klesá s rostoucím K i pro rovnoměrně rozložená data.
- Hledáme ”zpomalení klesání” W jakožto funkce K, respektive ”maximální rozdíl” od klesání na rovnoměrně rozložených datech.

Hierarchické klastrování – zdola nahoru

Každý bod ve vlastním klastru, spojuji vždy dva nejbližší klastry, dokud mám víc než jeden klastr. **Míry pro spojení:**

- **nejbližší body** (single linkage) $d_{SL}(G, H) = min_{i \in G, j \in H} d_{i,j}$.
  Má tendenci vytvářet "řetězce", krajní pozorování nejsou příliš podobná, klastr není kompaktní.
- **nejvzdálenější body** (complete linkage)
  $d_{CL}(G, H) = max_{i \in G, j \in H} d_{i,j}$. Opačný extrém – kompaktní klastr, ale neručí za blízkost, pozorování může být daleko blíže pozorováním z jiného klastru než kam samo patří.
- **průměr**, $d_{GA}(G, H) = \frac{1}{N_G N_C} \sum_{i \in G, j \in H} d_{i,j}$.
  Groop average – kompromis, citlivé na monotónní transformaci souřadnic.

Jsou–li klastry kompaktní a dobře separované, vše dá stejné výsledky.

Hierarchické klastrování – zhora dolů

- Na začátku je vše v jednom klastru.
- Dělím vždy na dva klastry, každou část znovu, dokud vše nerozdrobím
  - vyberu prvek s nevětší průměrnou vzdáleností od ostatních
  - dokud existuje prvek v původním klastru, jehož průměrná vzdálenost od starého klastru je větší než průměrná vzdálenost od nového klastru, přeřaď prvek s největším rozdílem.
- Pro další dělení vyberu klastr:
  - s největším průměrem, tj. vzdáleností nejvzdálenějších členů
  - nebo největší průměrnou vzdáleností.

Analýza nákupního koše

Apriori algoritmus

- Které věci lidé často kupují zároveň?
- Hodně velká data, i co do počtu sloupců ($p \approx 10^4$, $N \approx 10^8$).
- Zadám práh, minimální četnost kombinace tzv. **support** $T(.)$, $T(A \to B) = T(A\&B)$, která mě ještě zajímá.
- Při i–tém průchodu daty počítám pro všechny možné kandidáty délky i jejich četnost.
- Kandidátem do dalšího průchodu jsou jen ty množiny správné délky, že **všechny jejich podmnožiny** mají nadprahovou četnost.
- Množiny mohu přepsat na pravidla tak, že z možných kandidátů vyberu ta pravidla, která mají velkou přesnost (confidence) $C(A \to B) = \frac{T(A \to B)}{T(A)}$. Uvádí se i lift $L(A \to B) = \frac{C(A \to B)}{T(B)}$.

Příklad v Hastie, Tibshirani, Friedman

- Demografická data, $N = 9409$, vybrali jen 14 otázek.
- Vypustili záznamy s chybějícími hodnotami,
- ordinální proměnné rozdělili mediánem na dvě kategorie,
- kategoriální o k hodnotách rozdělili na k binárních poměnných, pozor – kódy 0 a 1 nejsou volně zaměnitelné, 1 má být to zajímavé.
- Výsledná matice měla 50 proměnných a 6876 záznamů.
- Algoritmus našel 6288 asociačních pravidel s 5 a méně prediktorů a support nejméně 10%.
- Málo četné hodnoty atributů se neprosadí, nejfrekventovanější jsou ty s vysokou četností (language=English).

Učení bez učitele jako Učení s učitelem

- V datech vytvoříme cílový atribut, všude dáme 1.
- Vezmeme referenční pravděpodobnostní rozložení, např. rovnoměrné, vygenerujeme stejný počet dat, hodnotu cílového atributu dáme 0.
- Narozdíl od APRIORI můžeme vzít jako referenční i např. gausovské rozložení, resp. předpoklad nezávislých veličin, tj. součinovou distribuci.
- Na vzniklá sjednocená data pustíme algoritmus na učení s učitelem.
- Vhodné např. rozhodovací stromy či pravidla (PRIM apod.).