0 Shluková analýza

Shluková analýza (clustering) je metoda nesupervizovaného učení, která rozřadí data do nějakých shluků tak, aby byly blízké body v jednom shluku a vzdálené v různých. Jak přesně formalizovat takovou úlohu je problematické, proto může mít různá řešení v závislosti na tom, jak definujeme vzdálenost.

Vzdálenost Pro připomenutí, definice metriky byla zevedena v sekci **??**.

Formalizace úlohy shlukování Úloha na vstupu dostane metrický prostor $\mathcal{X}$ se vzdáleností d, množinu dat $\mathcal{D} \subset \mathcal{X}$ a počet požadovaných shluků k. Na výstupu se požaduje nějaký rozklad množiny $\mathcal{D}$ na k shluků. To znamená $C_1, \ldots, C_k \subset \mathcal{D}$, kde $C_i \neq C_j$ pro všechna $i \neq j$, přičemž

$$\mathcal{D} = \bigcup_{i=1}^{k} C_i.$$

0.1 Hierarchické shlukování

Hierarchické shlukování používá hladový aglomerativní přístup popsaný v následujícím algoritmu.

Algorithm Hierarchické shlukování

Require:
Množina dat $\mathcal{D}$ a metrika pro vzdálenost shluků.
Požadovaný počet shluků k (jinak $k = 1$).

1: Uvažuj každý bod jako jednoprvkový shluk (celkem $|\mathcal{D}|$ shluků).
2: **while** počet shluků $> k$ **do**
3: Nalezni dva nejbližší shluky.
4: Tyto dva shluky spoj do jednoho.
5: **end while**

0.1.1 Měření vzdáleností shluků

Po volbě metriky dvou bodů $d(a, b)$ je ještě nutné zvollit vzdálenost dvou shluků $D(A, B)$. Obvykle se použije jedna z následujících:

- Single linkage, který generuje dlouhé řetězce

$$D(A, B) = \min_{\boldsymbol{x} \in A, \boldsymbol{y} \in B} d(\boldsymbol{x}, \boldsymbol{y}).$$

- Complete linkage, který generuje kompaktní buňky

$$D(A, B) = \max_{\boldsymbol{x} \in A, \boldsymbol{y} \in B} d(\boldsymbol{x}, \boldsymbol{y}).$$

- Average linkage, který měří podle průměrné vzdálenosti všech bodů

$$D(A,B) = \frac{1}{|A||B|} \sum_{\boldsymbol{x}\in A, \boldsymbol{y}\in B} d(\boldsymbol{x},\boldsymbol{y}).$$

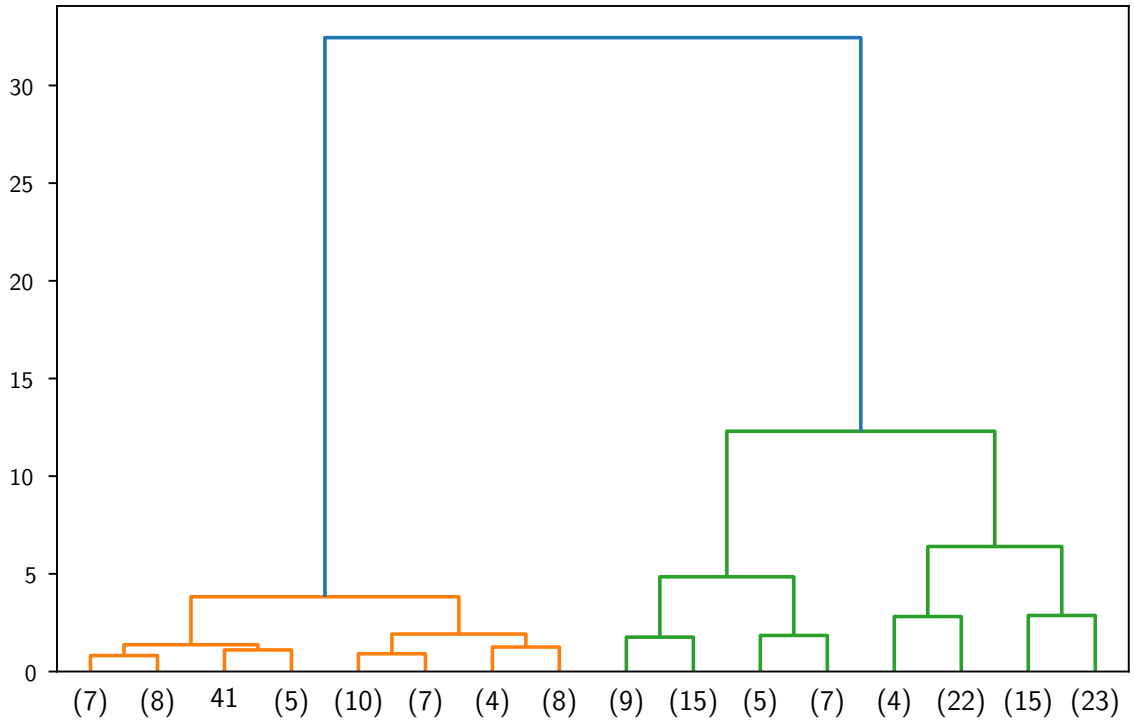
- Wardova metoda (pouze pro $\mathcal{X} = \mathbb{R}^p$), která minimalizuje nárůst vnitřního rozptylu buněk

$$D(A,B) = \sum_{\boldsymbol{x}\in A\cup B} \|\boldsymbol{x} - \bar{\boldsymbol{x}}_{A\cup B}\|^2 - \sum_{\boldsymbol{x}\in A} \|\boldsymbol{x} - \bar{\boldsymbol{x}}_A\|^2 - \sum_{\boldsymbol{x}\in B} \|\boldsymbol{x} - \bar{\boldsymbol{x}}_B\|^2,$$

kde $\bar{\boldsymbol{x}}_S$ značí geometrický střed (centroid) shluku S.

0.1.2 Dendrogram

Celý proces aglomerativního shlukování lze reprezentovat dendrogramem. To je strom, který má v listech počáteční jednoprvkové shluky a ve vrcholech spojení. Obvykle algoritmus běží, dokud nespojí všechny shluky, v takovém případě má v kořeni finální shluk. Strom se vykresluje tak, že listy jsou ve výšce 0 a jednotlivé vrcholy ve výšce podle vzdálenosti shluků, které se v daném bodě spojily (vzdálenost, kterou musely "překonat").



Máme-li požadované k, příslušně dendrogram rozřízneme tak, aby vodorovná čára protla přesně k hran. Každá taková hrana odpovídá nějakému shluku. Případně můžeme zvolit nějakou danou výšku (přijatelnou vzdálenost shluků).

Shrnutí Výhodou Hierarchického shlukování je flexibilita. Můžeme algoritmus provést a pak následně rozhodovat o rozdělení. Při změně počtu shluků stačí jen změnit místo, ve kterém provádíme řez (struktura dendrogramu se nemění).

Nevýhodou hierarchického shlukování je výpočetní náročnost $\mathcal{O}(n^3)$, v případě single/complete linkage lze zefektivnit na $\mathcal{O}(n^2)$. Pro velké datové soubory se proto nehodí.

0.2 Algoritmus k-means

0.2.1 Shlukování jako optimalizační úloha

Ke shlukování lze přistupovat jako k optimalizační úloze, ve které minimalizujeme účelovou funkci (objective function).

Pro dané k hledáme rozklad $C = (C_1, \ldots, C_k)$ množiny dat $\boldsymbol{X}$ v metrickém prostoru $\mathcal{X} = \mathbb{R}^p$ vybavenou Eukleidovskou vzdáleností minimalizující účelovou funkci

$$G(C) = \sum_{C_i} \frac{1}{2|C_i|} \sum_{\boldsymbol{x},\boldsymbol{y} \in C_i} \|\boldsymbol{x} - \boldsymbol{y}\|^2$$

Jedná se o průměrnou kvadratickou vzdálenost vnitřních bodů.

Souvislost účelové funkce s geometrickým středem Lze ukázat, že takový součet lze vyjádřit jako součet kvadrátů vzdálenosti od geometrického středu příslušných shluků.

$$\frac{1}{2|C_i|} \sum_{\boldsymbol{x},\boldsymbol{y} \in C_i} \|\boldsymbol{x} - \boldsymbol{y}\|^2 = \sum_{\boldsymbol{x} \in C_i} \|\boldsymbol{x} - \bar{\boldsymbol{x}}_{C_i}\|^2 = \min_{\boldsymbol{\mu} \in \mathbb{R}^p} \sum_{\boldsymbol{x} \in C_i} \|\boldsymbol{x} - \boldsymbol{\mu}\|^2$$

0.2.2 k-means

Algoritmus k-means využívá předchozí rovnosti a jediné, s čím "manipuluje", jsou středy shluků. Nalezení globálního minima je NP-těžká úloha, proto se používá iterativní způsob, který hladově zmenšuje hodnotu účelové funkce.

Algorithm k-means

Require:

Množina dat $\mathcal{D}$ a požadovaný počet shluků k.
Počáteční rozmístění $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$.

1: **while** hodnota účelové funkce "hodně" klesá **do**
2: Roztřiď $\mathcal{D}$ do shluků

$$Ci = \{\boldsymbol{x} \in \mathcal{D} \mid i = \arg\min_j \|\boldsymbol{x} - \boldsymbol{\mu}_j\|\}.$$

3: Přepočítej středy nových clusterů

$$\boldsymbol{\mu}_i \leftarrow \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{x}.$$

4: **end while**

Lze ukázat, že v každém kroku algoritmu se hodnota účelové funkce zmenší, případně zůstane stejná (v průběhu algoritmu je hodnota účelové funkce nerostoucí).

Běh algoritmu skončí, pokud se žádný ze středů nepřepočítá nebo pokud je rozdíl hodnot účelové funkce mezi iteracemi dostatečně malá.

Výsledek algoritmu hodně záleží na počátečním rozmístění středů, proto se typicky algoritmus spouští víckrát s různými počátky, z nichž si následně vybere shluk s nejnižší účelovou funkcí. Existuje rozšíření algoritmu, které vybírá počátky tak, aby byly "chytře" rozmístěné (k-means++).

Volba počtu shluků k-means algoritmus dává různé výsledky pro různá k, proto je důležité jej stanovit dopředu. S rostoucím k, zřejmě hodnota účelové funkce vždy klesá (případně neroste), proto je volba často subjektivní. Jednou z metod (i když nespolehlivých) je metoda lokte (elbow method), která předpokládá to, že existuje nějaké ideální $k*$, takové, že hodnota účelové funkce s rostoucím $k < k*$ rychle klesá, a pro $k > k*$ roste méně. Zlomový bod je předpokládaný loket, který ale není vždy úplně jednoznačný či optimální. Alternativní způsob ohodnocení clusteringu je například silhouette.

0.3 Silhouette skóre

Silhouette score je jednoduchou metodou pro evaluaci shlukování pro různá porovnání nebo i pro výběr optimálního počtu shluků (pro k-means).

Uvažujme nějaké shlukování $\mathcal{D} = C_1 \cup \ldots \cup C_k$ na metrickém prostoru $\mathcal{X}$ s metrikou $d(x, y)$, kde $\forall x \in \mathcal{D} : x \in C_{j(x)}$. Pro každý bod $x \in \mathcal{D}$ se spočítá:

- průměrná vzdálenost bodu od ostatních bodů ve stejném shluku:

$$a(x) = \frac{1}{|C_{j(x)}| - 1} \sum_{y \in C_{j(x)}, y \neq x} d(x, y)$$

- průměrná vzdálenost bodu od bodů v jiném shluku:

$$d(x, C_i) = \frac{1}{|C_{j(x)}|} \sum_{y \in C_i} d(x, y)$$

- průměrná vzdálenost bodu od nejbližšího jiného shluku:

$$b(x) = \min_{i \neq j(x)} d(x, C_i)$$

Evaluace pomocí Silhouette skóre Finální skóre bodu $x \in \mathcal{D}$ se počítá následujícím vzorcem:

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

V případě jednoho shluku je $s(x) = 0$.

Pokud je $s(x)$ blízko 1, $b(x) \gg a(x)$, pak je bod dobře zařazen. Pokud je $s(x)$ blízko 0, $b(x) \approx a(x)$, je bod na kraji svého a sousedního shluku (také v pořádku). Je-li však $s(x)$ blízko -1, $b(x) \ll a(x)$, je bod špatně zařazen. Poznamenejme, že v takovém případě nestačí bod jen přemístit do druhého – změnilo by celé ohodnocení (mohlo by být ve výsledku shlukování ještě zhoršit).

S ohodnocením jednotlivých bodů lze nyní provést evaluaci jednotlivých clusterů i celého shlukování:

$$s(C_i) = \frac{1}{|C_i|} \sum_{x \in C_i} s(x) \qquad s = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} s(x)$$

Vyšší hodnota (blíže k 1) značí lepší shlukování (lepší celkové umístění jednotlivých bodů).