

Modeliranje

Modeliranje

- ▶ Opisovanje odvisnosti podatkov.
- ▶ Statistični model - matematični model, ki ob nekih predpostavkah opisuje neke podatke (vzorec).
- ▶ Modele lahko tudi “učimo” - strojno učenje:
 - ▶ nadzorovano učenje (npr. linearna regresija),
 - ▶ nenadzorovano učenje (npr. razvrščanje),
 - ▶ vzpodbujevalno učenje.
- ▶ Ogledali si bomo osnovne primere prvih dveh.

Razvrščanje

- ▶ U - množica vseh enot podatkov
- ▶ $C_i \subseteq U$ - skupina enot
- ▶ $R = \{C_i\}_{i=1}^k$ - razvrstitev
- ▶ Φ - množica dopustnih razvrstitev
- ▶ $P : \Phi \mapsto \mathbb{R}_0^+$ - kriterijska funkcija
- ▶ R je **razbitje**, če velja:
 - ▶ $\bigcup_{i=1}^k C_i = U$,
 - ▶ $i \neq j \rightarrow C_i \cap C_j = \emptyset$.

Mera različnosti

- ▶ Funkcija $d(x, y) : U \times U \rightarrow \mathbb{R}$ je **mera različnosti**, če za vse $x, y \in U$ velja:
 - ▶ **nenegativnost**: $d(x, y) \geq 0$,
 - ▶ **refleksivnost**: $d(x, x) = 0$ in
 - ▶ **simetričnost**: $d(x, y) = d(y, x)$.
- ▶ Mera lahko ustreza še dodatnim pogojem (potem je **razdalja**):
 - ▶ **razločljivost**: $d(x, y) = 0 \Rightarrow x = y$,
 - ▶ **trikotniška neenakost**: za vsak $z \in U$ velja $d(x, y) \leq d(x, z) + d(z, y)$

Primeri različnosti

- ▶ Podatkovni enoti: $x = (x_1, \dots, x_m)$, $y = (y_1, \dots, y_m)$.
- ▶ Evklidska razdalja: $d(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$.
- ▶ Razdalje Minkowskega: $d(x, y) = (\sum_{i=1}^m |x_i - y_i|^r)^{\frac{1}{r}}$, $r > 0$.
- ▶ Manhattanska razdalja, pri $r = 1$.
- ▶ Trdnjavska razdalja: $d(x, y) = \max_{i=1, \dots, n} |x_i - y_i|$.

Razpršenost podatkov

- ▶ Primer: $x = (x_1, x_2)$, $x_1 \in [1000000, 10000000]$, $x_2 \in [0, 0.1]$
 - ▶ Vpliv x_2 je zanemarljiv pri izračunu omenjenih različnosti.
- ▶ Primer: $x = (x_1, x_2)$, $x_1 \in [1000000, 1000001]$, $x_2 \in [0, 1]$
 - ▶ razpršitev (razpon) podatkov je za obe komponenti enaka 1
 - ▶ vpliv enakega “premika” x_2 v območju vrednosti je zanemarljiv.
- ▶ Kako postaviti dimenzije podatkov v enakopraven položaj glede izračuna različnosti?
- ▶ Želimo enakopravnost tako glede prispevka same vrednosti kot tudi primerljivosti relativnega premika znotraj območij vrednosti.

Standardizacija

- ▶ Izenačenje skal podatkov.
- ▶ Najpogostejši način standardizacije.
- ▶ n - število podatkovnih enot, m - število dimenzij.
- ▶ $x^j = (x_1^j, \dots, x_m^j)$, $j = 1, \dots, n$ - podatkovne enote (vrstice).
- ▶ $\mu_i = \frac{\sum_{j=1}^n x_i^j}{n}$ - povprečje za komponento/dimenzijo i .
- ▶ $\sigma_i = \sqrt{\frac{\sum_{j=1}^n (x_i^j - \mu_i)^2}{n}}$ - standardni odklon (mera razpršenosti).
- ▶ Podatke x^j preslikamo v $z^j = (z_1^j, \dots, z_m^j)$, da velja $z_i^j = \frac{x_i^j - \mu_i}{\sigma_i}$.
- ▶ Dobimo spremenljivke, ki imajo povprečje 0 in so na podoben način razpršene okoli 0 (standardni odklon je 1).

Kriterijska funkcija

- ▶ Iščemo “optimalno” razbitje za kriterijsko funkcijo f (minimum)
- ▶ Primer: Wardova kriterijska funkcija
 - ▶ $f(R) = \sum_{C_i \in R} \sum_{x \in C_i} d(x, c(C_i))$,
 - ▶ d - kvadrat evklidske razdalje,
 - ▶ $c(C_i)$ je *centroid* C_i - povprečna točka skupine, $c(C_i) = \frac{\sum_{x \in C_i} x}{|C_i|}$.
 - ▶ http://en.wikipedia.org/wiki/Hierarchical_clustering

Posplošitev različnosti na skupine

- ▶ Za dano različnost $d(x, y)$ želimo to posplošiti na $d(C_i, C_j)$, kjer sta C_i in C_j disjunktni množici podatkov.
- ▶ Želimo $d(x, C_j) = d(\{x\}, C_j)$.
- ▶ Nekateri možnosti posplošitve:
 - ▶ $d(C_i, C_j) = d(c(C_i), c(C_j))$
 - ▶ $d(C_i, C_j) = \max\{d(x, y) | x \in C_i, y \in C_j\}$
 - ▶ $d(C_i, C_j) = \min\{d(x, y) | x \in C_i, y \in C_j\}$
 - ▶ ...
- ▶ S takšnimi posplošitvami lahko pridemo različne kriterijske funkcije.

Problem razvrščanja

- ▶ V splošnem imamo pač neko kriterijsko funkcijo na razbitjih.
- ▶ Izkaže se, da je iskanje optimuma v večini zanimivih primerov kriterijskih funkcij zelo težek računski problem (NP težek).
- ▶ Torej, ni znan bistveno boljši algoritem, kot da preverimo vse možnosti.
- ▶ Zato uporabljamo približne algoritme (hevrstike).

Hierarhično razvrščanje

- ▶ Množica podatkovnih enot: $x^j, j = 1, \dots, n$.
- ▶ Na začetku je vsaka podatkovna enota v svoji skupini $C_j^0 = \{x^j\}$, imamo n skupin, torej razbitje $R^0 = \{C_j^0\}_{j=1}^n$.
- ▶ Na vsakem koraku poiščemo najbližji množici v razbitju in ju združimo; tako dobimo razbitje z eno množico manj; končamo, ko združimo vse podatkovne enote.
- ▶ Za vsak korak k od 1 do $n - 1$
 - ▶ C_1^k postane unija najbližjih množic v razbitju R^{k-1}
 - ▶ ostale množice preimenujemo v C_2^k, \dots, C_{n-k}^k ,
 - ▶ $R^k := \{C_1^k, \dots, C_{n-k}^k\}$.
- ▶ Primeri in uporabe.
 - ▶ <https://joyofdata.shinyapps.io/hclust-shiny/>

Metoda voditeljev

- ▶ Izberemo število voditeljev k . Toliko bo na koncu skupin.
- ▶ Naključno izberemo *voditelje* elemente množice U (ni nujno, da gre za podatkovne enote).
- ▶ Ponavljamo:
 - ▶ določimo razvrstitev podatkovnih enot tako, da jih priredimo v skupino k najbližjemu voditelju
 - ▶ za nove skupine izračunamo centroide in ti postanejo novi voditelji
- ▶ Postopek ponavljamo dokler se skupine ne ustalijo.
- ▶ Primeri in aplikacije
- ▶ <https://shiny.rstudio.com/gallery/kmeans-example.html>

Linearna regresija

- ▶ Primer nadzorovanega učenja.
- ▶ Dan imamo nabor parov (x_i, y_i) , $x \in \mathbb{R}$, $y \in \mathbb{R}$, $i = 1, \dots, n$.
- ▶ Ti predstavljajo meritve nekega pojava, kjer je x_i dimenzija meritve, y_i pa sama meritev.
- ▶ Primer: x_i je neka hitrost avtomobila, y_i pa meritev, kako dolga je bila zavorna pot (primer je zbirka cars v R).
- ▶ Želimo določiti funkcijo $f : \mathbb{R} \rightarrow \mathbb{R}$, ki za dan x vrne “kaj naj bi bil y ”.

Linearna regresija

- ▶ Kakšno funkcijo?
- ▶ Če želimo linearno, je z več kot dvema točkama sistem predoločen.
- ▶ Poiščemo “najbolj” primerno premico.
- ▶ Metoda najmanjših kvadratov.
- ▶ https://gallery.shinyapps.io/slr_diag/