19. Grupiranje

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

Jan Šnajder, natuknice s predavanja, v1.1

1 Nenadzirano učenje

- Raspolažemo skupom **neoznačenih primjera** (*unlabeled data*): $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$
- Primjerni su neoznačeni jer ih (1) ne znamo označiti ili (2) označavanje je preskupo
- Osnovni zadatci nenadziranog učenja:
  - grupiranje (*clustering*)
  - procjena gustoće (*density estimation*)
  - otkrivanje novih/stršećih vrijednosti (*novelty/outlier detection*)
  - smanjenje dimenzionalnosti (*dimensionality reduction*)
- **Polunadzirano učenje**: većina primjera je neoznačena

2 Grupiranje

- Razdjeljivanje primjera u grupe (*clusters*), tako da su **slični** primjeri u istoj grupi
- Nalaženje "prirodnih" (intrinzičnih) grupa u skupu neoznačenih podataka
- Vrste grupiranja: **čvrsto/meko**, **particijsko/hijerarhijsko**
- Primjene: (1) istraživanje podataka, (2) kompresija, (3) polunadzirano učenje
- Grupiranje primjera / grupiranje značajki / bi-clustering

3 Algoritam K-sredina

- Particijsko grupiranje u K čvrstih grupa (K je unaprijed određen)
- **Funkcija pogreške** (kriterijska funkcija):

$$J = \sum_{k=1}^{K} \sum_{i=1}^{N} b_k^{(i)} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|^2$$

gdje je $\boldsymbol{\mu}_k$ centroid k-te grupe, a $b_k^{(i)}$ indikatorska varijabla pripadnosti $\mathbf{x}^{(i)}$ grupi k

- Svaki primjer $\mathbf{x}^{(i)}$ svrstavamo u grupu s njemu najbližim centroidom $\boldsymbol{\mu}_k$:

$$b_k^{(i)} = \begin{cases} 1 & \text{ako } k = \underset{j}{\operatorname{argmin}} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_j\| \\ 0 & \text{inače} \end{cases}$$

- Tražimo grupiranje $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ koje minimizira pogrešku: $\operatorname{argmin}_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K} J$
- Analitička minimizacija nije moguća jer su $b_k^{(i)}$ i $\boldsymbol{\mu}_k$ međuovisni
- Alternativa: **iterativna optimizacija**
  - Fiksiramo $\boldsymbol{\mu}_k$ na neke inicijalne vrijednosti
  - Pridružimo primjere grupama (izračunamo $b_k^{(i)}$ za $i = 1, \ldots, N$)
  - Uz fiksne $b_k^{(i)}$, minimizacija J daje formulu za ažuriranje centroida:

$$\nabla_{\boldsymbol{\mu}_k} J = \mathbf{0} \quad \Rightarrow \quad 2\sum_{i=1}^{N} b_k^{(i)}(\mathbf{x}^{(i)} - \boldsymbol{\mu}_k) = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{\mu}_k = \frac{\sum_i b_k^{(i)} \mathbf{x}^{(i)}}{\sum_i b_k^{(i)}}$$

  - Ponavljamo do konvergencije $\boldsymbol{\mu}_k$ odnosno $b_k^{(i)}$

Algoritam K-sredina (k-means algorithm)

inicijaliziraj centroide $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$
ponavljaj
  za svaki $\mathbf{x}^{(i)} \in \mathcal{D}$
    $b_k^{(i)} \leftarrow \begin{cases} 1 & \text{ako } k = \operatorname{argmin}_j \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_j\| \\ 0 & \text{inače} \end{cases}$
  za svaki $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$
    $\boldsymbol{\mu}_k \leftarrow \sum_{i=1}^{N} b_k^{(i)} \mathbf{x}^{(i)} / \sum_{i=1}^{N} b_k^{(i)}$
dok $\boldsymbol{\mu}_k$ ne konvergiraju

- Vremenska složenost za T iteracija: $T\big(\mathcal{O}(nNK) + \mathcal{O}(nN)\big) = \mathcal{O}(TnNK)$
- **Konvergencija algoritma**:
  - Broj konfigurcija (particija) je konačan i iznosi K^N
  - J monotono pada kroz iteracije

  $\Rightarrow$ algoritam svaku konfiguraciju posjećuje najviše jednom $\Rightarrow$ **algoritam konvergira**
- **Optimalnost algoritma**:
  - Algoritam **pohlepno pretražuje** konfiguracije te nalazi **lokalni optimum** od J
  - Optimalnost rješenja ovisi o odabiru početnih središta
- Pristupi za odabir početnih središta:

  - Nasumičan odabir primjera kao centroida
  - K slučajnih vektora prirodanih centroidu cijelog skupa podataka
  - K centroida iz K segmenata primjera projiciranih na prvu PCA komponentu
  - **k-means++**: vjerojatnost odabira primjera $\mathbf{x}^{(i)}$ kao novog središta $\boldsymbol{\mu}_{k+1}$:

$$P(\boldsymbol{\mu}_{k+1} = \mathbf{x}^{(i)} | \mathcal{D}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k) = \frac{\min_k \|\boldsymbol{\mu}_k - \mathbf{x}^{(i)}\|^2}{\sum_j \min_k \|\boldsymbol{\mu}_k - \mathbf{x}^{(j)}\|^2}$$

    $\Rightarrow$ vjerojatnost je proporcionalna kvadratu udaljenosti od već odabranih središta

- Grupiranje treba pokrenuti više puta i uzeti rezultat s najmanjim J

4 Algoritam K-medoida

- Algoritma K-sredina: (1) primjeri moraju biti vektori, (2) udaljenost je euklidska
- **Algoritam K-medoida**: poopćenje K-sredina za općenitu mjeru sličnosti/različitosti
- Prototipi grupa nisu centroidi nego **medoidi** (odabrani primjeri u svakoj grupi)
- Funkcija pogreške:

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} b_k^{(i)} \nu(\mathbf{x}^{(i)}, \boldsymbol{\mu}_k)$$

  gdje je $\nu : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ općenita **mjera različitosti** dvaju primjera
- Tipična izvedba je **algoritam PAM** (*partitioning around medoids*)

Algoritam PAM

inicijaliziraj medoide $\mathcal{M} = \{\boldsymbol{\mu}_k\}_{k=1}^K$ na odabrane $\mathbf{x}^{(i)}$
ponavljaj
  za svaki $\mathbf{x}^{(i)} \in \mathcal{D} \setminus \mathcal{M}$
    $b_k^{(i)} \leftarrow \begin{cases} 1 & \text{ako } k = \operatorname{argmin}_j \nu(\mathbf{x}^{(i)}, \boldsymbol{\mu}_j) \\ 0 & \text{inače} \end{cases}$
  za svaki $\boldsymbol{\mu}_k \in \mathcal{M}$
    $\boldsymbol{\mu}_k \leftarrow \operatorname{argmin}_{\boldsymbol{\mu}_j \in \mathcal{D} \setminus \mathcal{M} \cup \{\boldsymbol{\mu}_k\}} \sum_i b_k^{(i)} \nu(\mathbf{x}^{(i)}, \boldsymbol{\mu}_j)$
dok $\boldsymbol{\mu}_k$ ne konvergiraju

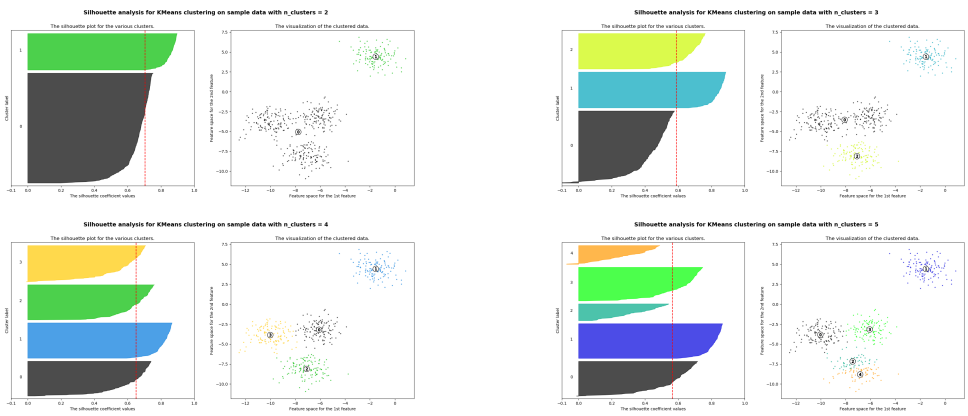
- Složenost za T iteracija: $T\big(\mathcal{O}(K(N-K)) + \mathcal{O}(K(N-K)^2)\big) = \mathcal{O}(TK(N-K)^2)$
- Nedostatak algoritma PAM: visoka vremenska složenost

5 Provjera grupa

- **Broj grupa** K koji mnogih je algoritama grupiranja potrebno odrediti unaprijed
- Odabir optimalnog broja grupa dio je **provjere grupiranja** (*cluster validation*)
- J ostvaruje minimum za $K = N \Rightarrow$ nije indikativno za optimalan broj grupa
- Jednostavnije metode za odabir broja grupa:
  - Ručna provjera kvalitete grupa
  - Redukcija dimenzija u 2D-prostor (PCA, MDS, CA, t-SNE) i vizualna provjera
  - Metoda "koljena" (*elbow method*) – nalaženje platoa funkcije $J(K)$
- **Analiza siluete** (*silhouette analysis*):
  - Silueta primjera $\mathbf{x}^{(i)}$:

$$s(i) = \frac{b(i) - a()}{\max(a(i), b(i))} \in [-1, +1]$$

  - $a(i)$ i $b(i)$ su prosjek udaljenost od $\mathbf{x}^{(i)}$ do primjera iste odnosno najbliže grupe
  - Računamo i grafički prikazujemo $s(i)$ za sve primjere svake grupe
  - Loše grupiranje: ispodprosječne siluete nekih grupa i/ili visoka varijanca silueta
  - Primjer (scikit-learn):



- **Minimizacija regularizirane funkcije pogreške**:
  - Kažnjavanje modela s velikim brojem grupa:

$$K^* = \underset{K}{\operatorname{argmin}} \left(J(K) + \lambda K \right)$$

  - **Akaikeov kriterij (AIC)** za algoritam K-sredina: $\lambda = 2n$
- **Točnost na podskupu primjera**:
  - Raspolažemo označenim podskupom primjera ili parova primjera

– **Randov indeks** – točnost grupiranja na razini parova primjera:

$$R = \frac{a+b}{\binom{N}{2}} \in [0,1]$$

– a – broj jednako označenih parova u istim grupama

– b – broj različito označenih parova u različitim grupama

– Optimalan K je onaj koji maksimizira $R(K)$