UČENJE IZ PODATKOVNIH TOKOV (II. DEL)

Aljaž Osojnik

aljaz.osojnik@ijs.si

Odsek za tehnologije znanja,

Inštitut Jožef Štefan

POVZETEK – II. DEL

- Delitve na numeričnih atributih
- Prilagajana okna – ADWIN
- Sprotni učenje iz samovzorcev (online bagging)
  - ADWIN Bagging
- Na primerih osnovano sprotno učenje

DELITVE NA NUMERIČNIH ATRIBUTIH

POTREBNE STATISTIKE

- Za izračun redukcije variance potrebujemo sledeče statistike:
- **Število primerov** – k
- **Povprečje** – Σ_x
- **Varianca** –Σ_{x^2}

NOMINALNE DELITVE – POTREBNE STATISTIKE

- Za vrednotenje nominalnih testov (ne glede na obliko testa) potrebujemo statistike za vsako vrednost nominalnega atributa
- Statistike hranimo v tabeli
- **Opazimo:** statistike za posamezne vrednosti atributov so popolnoma neodvisne

NUMERIČNE DELITVE – POTREBNE STATISTIKE

- Za vrednotenje numeričnih delitev potrebujemo statistike za vsako možno delitev $X \leq a$
- Potrebujemo statistike za $\leq a$ in $> a$
- Statistike bi lahko hranili tabelarično
- **Problem:** velikost tabele je treba prilagajati, vstavljanje je počasno
- **Ideja:** uporabimo iskalno drevo

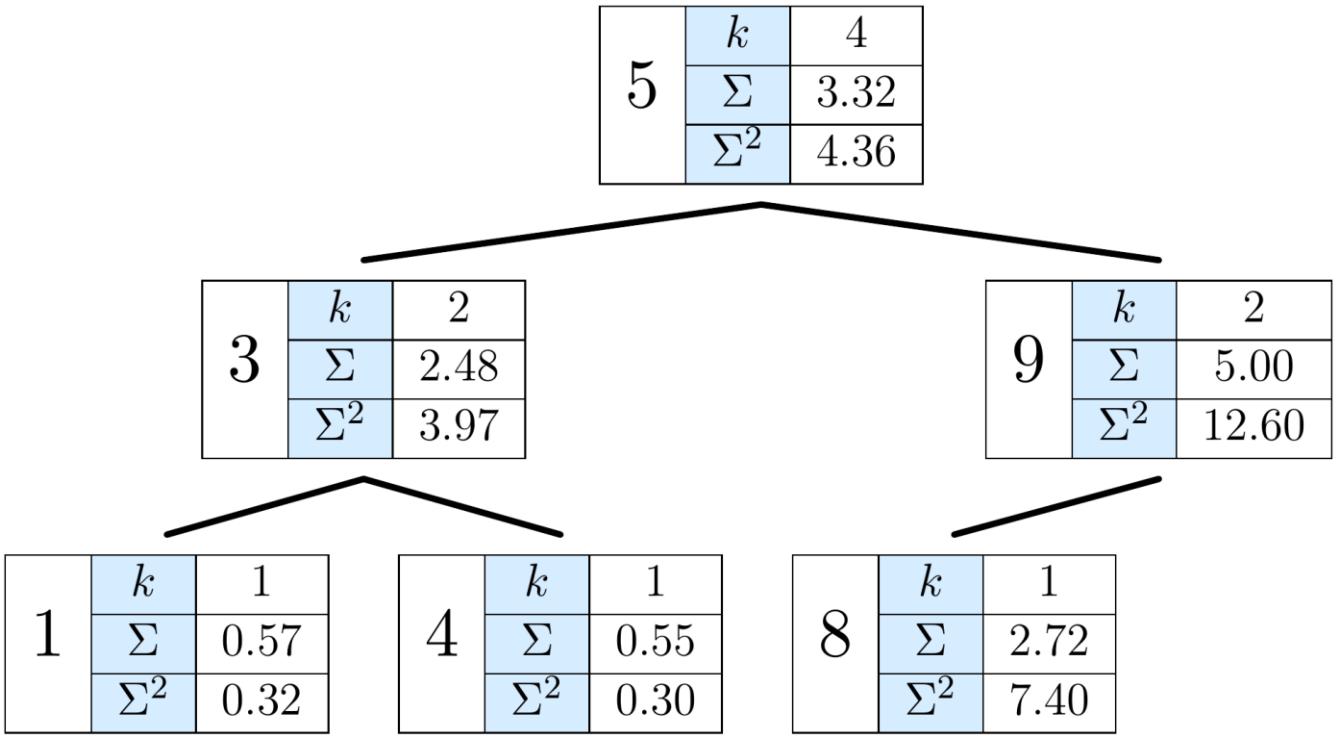
EBST - RAZŠIRJENO BINOMSKO ISKALNO DREVO

- Razširjeno binomsko iskalno drevo (*an. Extended Binary Search Tree, EBST*)
- Vsako vozlišče pripada eni vrednosti a danega atributa
- V vsakem vozlišču hranimo (delne) statistike za primere, ki imajo vrednost danega atributa $\leq a$
- Statistik za primere z vrednostmi $> a$ ne hranimo eksplicitno (prihranimo spomin)

POSODABLJANJE STATISTIK

- Pri vstavljanju novih vrednosti nočemo popravljati tudi "desnih" potomcev (vrednost razvrščamo le enkrat)
- Hranimo delne statistike
- Statistike za $X \leq a$ in $X \leq b$ $(a < b)$ niso neodvisne
- Primeri $\leq a$ so tudi $\leq b$
- Ko vrednotimo delitve, vzdržujemo trenutne statistike
  - Ko se v drevesu pomikamo v leve potomce, trenutnih statistik ne posodabljamo
  - Ko se v drevesu pomaknemo v desnega potomca, trenutnim statistikam prištejemo statistike trenutnega vozlišča

PRIMER EBST



A	T
5	0.29
9	2.28
3	1.91
1	0.57
8	2.72
4	0.55

POSODABLJANJE STATISTIK

- Za vsak numerični atribut hranimo EBST
- Za posodobitev EBST z novim primerom z vrednostjo a moramo v drevesu najti ustrezno vozlišče oz. ga ustvariti
- Povprečna časovna zahtevnost vstavljanja je $O(\log(n))$, kjer je n velikost drevesa (ker hranimo delne statistike)

RAČUNANJE STATISTIK $> a$

- Kako izračunamo statistike za vrednosti $> a$?
- Hranimo globalne statistike n, Σ_x, Σ_{x^2}
- Statistike za $> a$ izračunamo iz globalnih statistik in statistik za $\leq a$

$$n_{>a} = n - n_{\leq a}$$
$$\Sigma_{x_{>a}} = \Sigma_x - \Sigma_{x_{\leq a}}$$
$$\Sigma_{x^2{}_{>a}} = \Sigma_{x^2} - \Sigma_{x^2{}_{\leq a}}$$

- Na ta način znižamo porabo pomnilnika

PRILAGAJANA OKNA – ADWIN

PRILAGAJANA OKNA

- Prilagajana okna (an. *ADaptive WINdow*) [Bifet, Gavaldá, 2006]
- Hranimo najdaljše okno, za katerega velja, da se povprečje znotraj okna ni spremenilo
- Del okna zavržemo, kadar je njegovo povprečje dovolj različno od povprečja preostanka okna

ADWIN

- Imamo signal $x_1, \ldots, x_t, \ldots$
- x_i so omejeni
  - BŠS: $x_i \in [0, 1]$, dosežemo z normalizacijo
- Okno primerov označimo z W
- **Ideja:** če sta dela okna W_0 in W_1 "dovolj različna", lahko starejši del okna zavržemo
- Ko se W skrči, zaznamo spremembo

ADWIN (2)

Kdaj sta povprečji dovolj različni?

Kadar se povprečji W_0 in W_1 razlikujeta za več kot

$$\varepsilon = \sqrt{\frac{1}{2m} \ln\left(\frac{4|W|}{\delta}\right)},$$

kjer je

$$m = \frac{1}{1/|W_0| + 1/|W_1|}$$

harmonična sredina dolžin W_0 in W_1. Tedaj skrčimo W.

ADWIN – ZAGOTOVILI

- **Omejitev napačno zaznanih sprememb (*an. false positive rate*):** Če se povprečje v W ni spremenilo, je verjetnost da smo okno skrčili manjša od δ.
- **Omejitev zgrešenih sprememb (*an. false negative rate*):** Denimo, da za neko particijo $W_0W_1 = W$ velja $|\mu_0 - \mu_1| > 2\varepsilon$. Tedaj ADWIN z verjetnostjo $1 - \delta$ okno W skrajša na vsaj W_1 (kjer W_1 vsebuje novejše primere).

HOEFFDINGOVA NEENAKOST

Posledica: Predpostavimo kot prej. Tedaj velja

$$\Pr\left(\left|\overline{X} - \mathbb{E}\left[\overline{X}\right]\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{2n^2\varepsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right) =: \delta.$$

ε izrazimo z δ:

$$\varepsilon \leq \sqrt{\frac{\sum_{i=1}^{n}(b_i - a_i)^2}{2n^2}\ln\left(\frac{2}{\delta}\right)}.$$

ADWIN – PRIPRAVA

- $W_0 W_1 = W$ za neko delitev W
- $|W| = n, |W_0| = n_0, |W_1| = n_1, n = n_0 + n_1$
- $m = \frac{n_0 n_1}{n_0 + n_1}$
- $\mu, \hat{\mu}$ – pravo in izmerjeno povprečje na W
- $\mu_0, \hat{\mu}_0$ – pravo in izmerjeno povprečje na W_0
- $\mu_1, \hat{\mu}_1$ – pravo in izmerjeno povprečje na W_1

OMEJITEV NAPAČNO ZAZNANIH SPREMEMB

- Če se pravo povprečje μ ni spremenilo, velja
$$\mu = \mu_0 = \mu_1$$
- Za vsak $k \in (0, 1)$ velja:
$$\Pr(|\hat{\mu}_1 - \hat{\mu}_0| \geq \varepsilon) \leq \Pr(|\hat{\mu}_1 - \mu_1| \geq k\varepsilon) + \Pr(|\hat{\mu}_0 - \mu_0| \geq (1-k)\varepsilon)$$
- Uporabimo Hoeffdingovo neenakost:
$$\Pr(|\hat{\mu}_1 - \hat{\mu}_0| \geq \varepsilon) \leq 2\exp(-2(k\varepsilon)^2 n_0) + 2\exp\left(-2\left((1-k)\varepsilon\right)^2 n_1\right)$$
- Izberemo k pri katerem sta člena enaka (minimira vsoto)
$$(k\varepsilon)^2 n_0 = \left((1-k)\varepsilon\right)^2 n_1$$

OMEJITEV NAPAČNO ZAZNANIH SPREMEMB (2)

$$k = \frac{\sqrt{n_1/n_0}}{\left(\sqrt{n_0} + \sqrt{n_1}\right)}$$

Za ta k velja:

$$(k\varepsilon)^2 n_0 = \frac{n_0 n_1}{\left(\sqrt{n_0} + \sqrt{n_1}\right)^2}\varepsilon^2 \leq \frac{n_0 n_1}{n_0 + n_1}\varepsilon^2 = m\varepsilon^2$$

Če želimo, da

$$\Pr(|\hat{\mu}_1 - \hat{\mu}_0| \geq \varepsilon) \leq \frac{\delta}{n}$$

zadostuje:

$$4\exp(-2m\varepsilon^2) \leq \frac{\delta}{n} \textbf{ oz. } \varepsilon = \sqrt{\frac{1}{2m}\ln\frac{4n}{\delta}}$$

OMEJITEV NAPAČNO ZAZNANIH SPREMEMB (3)

- Da dobimo končni rezultat (verjetnost napačno zaznane spremembe $< \delta$), seštejemo verjetnosti vseh možnih delitev (teh pa je ravno n)

$$\Pr(|\hat{\mu}_1 - \hat{\mu}_0| \geq \varepsilon \text{ za vse } W_0 W_1 = W) \leq \delta$$

OMEJITEV ZGREŠENIH SPREMEMB

- Privzemimo $|\mu_0 - \mu_1| > 2\varepsilon$
- Za vsak $k \in (0,1)$:

  $\Pr(|\hat{\mu}_0 - \hat{\mu}_1| \geq \varepsilon) \leq \Pr\big((|\hat{\mu}_0 - \mu_0| \geq k\varepsilon) \cup (|\hat{\mu}_1 - \mu_1| \geq (1-k)\varepsilon)\big)$

  $\Pr(|\hat{\mu}_0 - \hat{\mu}_1| \geq \varepsilon) \leq \Pr(|\hat{\mu}_0 - \mu_0| \geq k\varepsilon) + \Pr(|\hat{\mu}_1 - \mu_1| \geq (1-k)\varepsilon)$
- Hoeffdingova neenakost:

  $$\Pr(|\hat{\mu}_0 - \hat{\mu}_1| \geq \varepsilon) \leq \exp(-2(k\varepsilon)^2 n_0) + \exp\left(-2\big((1-k)\varepsilon\big)^2 n_1\right)$$
- Izberemo k kot prej in dobimo:

$$\Pr(|\hat{\mu}_1 - \hat{\mu}_0| \geq \varepsilon) \leq 4\exp(-2m\varepsilon^2) \leq \frac{\delta}{n} \leq \delta$$

UČENJE IZ SAMOVZORCEV NA PODATKOVNIH TOKOVIH

ANSAMBLI

- Uporabljamo več osnovnih modelov, njihove napovedi združujemo
- **Ideja:** če modeli ansambla delajo raznolike napake, z združevanjem njihovih napovedi dobimo boljšo napoved (znižujemo pristranskost)

RAZNOLIKOST MODELOV PRI UČENJU IZ SAMOVZORCEV

- Kako dosežemo raznolikost modelov?
- Naučimo jih na različnih (samo)vzorcih podatkovnega nabora
- Samovzorce danega podatkovnega nabora dobimo z vzorčenjem s ponovitvami
- Vsak model se uči iz svojega samovzorca zato je bolj prilagojen na ponovljene primere v vzorcu
- V povprečju je v samovzorcu delež ponovljenih primerov $1/e$, preostali se pojavijo le enkrat

KAKO DOBIMO SAMOVZOREC NA PODATKOVNEM TOKU?

- Samovzorcev ne moremo vzorčiti kot običajno
- Spremenimo zorni kot:
  - Samovzorca ne opazujemo kot množice (nabora)
  - samovzorec opazujemo primer po primer
- Opazimo: vsak primer se v samo vzorcu pojavi 0-, 1-ali več-krat, do največ n-krat
- Samovzorec za vsak primer natančno določa kolikokrat se pojavi ("vektor" ponovitev)
- Kolikšna je verjetnost, da se dani primer pojavi natanko k-krat ($0 \leq k \leq n$)?

SAMOVZOREC – PO EN PRIMER NAENKRAT

- Samovzorec dobimo z vzorčenjem s ponovitvami
- Število ponovitev danega primera je med 0 in n
- Porazdeljeno je $\sim \mathrm{B}\left(n, \frac{1}{n}\right)$
- Verjetnost k ponovitev danega primera je:

$$\Pr(X = k) = \binom{n}{k} \left(\frac{1}{n}\right)^{k} \left(1 - \frac{1}{n}\right)^{n-k}$$

SAMOVZOREC NA PODATKOVNEM TOKU

- Podatkovni tok lahko opazujemo kot podatkovni nabor, katerega velikost narašča proti ∞
- Število ponovitev danega primera:

$$\Pr(X = k) = \lim_{n\to\infty} \binom{n}{k} \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} =$$

$$= \lim_{n\to\infty} \frac{n!}{(n-k)!\,k!} \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^n \left(1 - \frac{1}{n}\right)^{-k} =$$

$$= \frac{1}{e \cdot k!} \lim_{n\to\infty} \frac{n! n^k}{(n-k)! n^k (n-1)^k} = \frac{1}{e \cdot k!} \lim_{n\to\infty} \frac{n(n-1)\cdots(n-k+1)}{(n-1)^k} =$$

$$= e^{-1} \cdot \frac{1}{k!} = e^{-\lambda} \cdot \frac{\lambda^k}{k!}, \text{ kjer } \lambda = 1 = n \cdot \frac{1}{n} = n \cdot p = \mathbb{E}[X]$$

SAMOVZOREC NA PODATKOVNEM TOKU

- Ker $n \to \infty$, je lahko k poljubno visok
- Ta porazdelitev je natanko $\mathrm{Poisson}(1)$
- Za vsak primer znamo izračunati koliko ponovitev je v danem samovzorcu
- Namesto eksplicitnega računanja samovzorcev obravnavamo vsak primer posamično
- Tedaj:
  - Za vsak primer in osnovni model vzorčimo $k \sim \mathrm{Poisson}(1)$
  - Osnovni model naučimo s k ponovitvami danega primera

SPROTNO UČENJE IZ SAMOVZORCEV

- Online bagging [Oza, Russel, 2001]
- Drugačen pogled kot pri običajnem učenju iz samovzorcev
- Samovzorcev ne računamo eksplicitno
- Za vsak primer izračunamo, kolikokrat se pojavi za dani osnovni model

SPROTNO UČENJE IZ SAMOVZROCEV – h_0 IN u

- Začetna hipoteza h_0:
  - Nabor začetnih n hipotez osnovnega modela g_0

$$\mathrm{h}_0 = [g_0^1, g_0^2, \dots, g_0^n]$$

- Posodobitveni operator u:
  - Za primer (x, y)
  - Za vsak osnovni model g_j^i s posodobitvenim operatorjem u_g (i šteje člane ansambla, j primere)
    - Vzorčimo $k \sim \mathrm{Poisson}(1)$
    - $g_{j+1}^i = \underbrace{u_g\left(u_g\left(\dots\left(u_g\left(g_j^i, (x,y)\right), (x,y)\right)\dots\right), (x,y)\right)}_{k-\text{krat}}$

ADWIN BAGGING

- Kombinacija sprotnega učenja iz samovzorcev in ADWIN [Bifet et al., 2009]
- ADWIN kot mehanizem za zaznavanje sprememb (en ADWIN na osnovni model)
  - Opazujemo napako
- Kakšno je prilagajanje na spremembe?
- Ker imamo ansambel, ne izgubimo preveč, če zavržemo posamezni model
- Ko za osnovni model zaznamo spremembo, ga zavržemo in začnemo učiti novega

NA PRIMERIH OSNOVANO SPROTNO UČENJE

NA PRIMERIH OSNOVANO UČENJE

- Poseben pomen dajemo primerom
- "Leno" učenje – računamo šele, ko smo "vprašani"
- Posplošitev metode najbližjih sosedov (kNN)
- Potrebujemo razdaljo na vhodnem prostoru Δ
- Trenutne izkušnje predstavlja baza primerov (pri običajnem učenju najpogosteje učna množica)

NA PRIMERIH OSNOVANO UČENJE VS UČENJE Z MODELIRANJEM

- Na primerih osnovano učenje:
  - Posodabljanje je poceni (samo dodajamo ali odstranjujemo primere)
  - Računanje napovedi je relativno zahtevno
    - Iskanje ustreznih primerov
    - Računanje napovedi iz najdenih primerov
- Učenje z modeliranje
  - Posodabljanje je potratno
  - Računanje napovedi je preprosto
- Na primerih osnovano učenje uporabljamo, kadar imamo veliko učnih podatkov, ne potrebujemo pa pogostih napovedi

NA PRIMERIH OSNOVANA KLASIFIKACIJA

- Napoved praviloma za primer x izračunamo z uporabo večinskega glasovanja:

$$\hat{y} = \text{argmax}_{y \in Y}\{(x_i, y_i) \in \mathcal{N}_k(x) | y_i = y\}$$

- Posplošimo z uteženostjo primerov glede na njihovo razdaljo od x:

$$\hat{y} = \text{argmax}_{y \in Y} \sum_{(x_i, y_i) \in \mathcal{N}_k(x); y_i = y} w(x_i)$$

kjer

$$w(x_i) = \frac{f\left(\Delta(x_i, x)\right)}{\sum_{(x_j, y_j) \in \mathcal{N}_k(x)} f\left(\Delta\left(x_j, x\right)\right)}$$

$f: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ je neka padajoča funkcija (manjša razdalja vodi v večjo utež)

ZAHTEVE ZA BAZO PRIMEROV PRI SPROTNEM UČENJU

- **Časovna ustreznost:** novejši primeri so pomembnejši kot starejši.
- **Prostorska ustreznost:** baza idealno enakomerno pokriva vhodni prostor. Primeri, ki ne vplivajo na napovedane vrednosti, so odveč.
- **Konsistentnost:** ciljne vrednosti bližnjih primerov naj se ne razlikujejo preveč.

IBLSTREAMS

- IBLStreams [Shaker, Hűllermeier, 2012]
- Sprotno vzdrževanje baze primerov
- Za vsak primer x obravnavamo „okolico" C, ki vsebuje k_c najbližjih primerov
- Napoved je najpogostejši razred v C
- Pri posodabljanju baze primerov ne odstranjujemo najnovejših primerov (časovna ustreznost)

IBLSTREAMS – POSODABLJANJE BAZE PRIMEROV

- Pri posodabljanju baze primerov (učenju) pri novem primeru (x, y) opazujemo testno množico
- Testno množico določa x-u najbližjih $k_c^2 + k_c$ primerov
- Med k_c najbližjimi primeri določimo večinski razred $\hat{y}$
- Če $\hat{y} = y$, odstranimo primere, za katere $y_i \neq y$ (k_c najnovejših primerov ne odstranjujemo)
- Če z dodajanjem novega primera prekoračimo velikost C, zavržemo še najstarejši primer

IBLSTREAMS – STATIČNI IN SPREMENLJIVI TOKOVI

- Pri statičnih tokovih se natančnost povečuje z večjo bazo primerov
- Pri spremenljivih tokovih imamo več zastarelih primerov, ki jih moramo odstraniti
  - Počasnejše prilagajanje spremembam
- IBLStream eksplicitno zaznava nenadne spremembe s pomočjo napovedne napake, njene standardne deviacije in z-testa

IBLSTREAMS – PRILAGAJANJE PARAMETROV

- Neposredno prilagajanje velikosti okolice C, tj. števila relevantnih sosedov k_c
  - Na vsakem koraku k_c obdržimo ali pa povečamo ali zmanjšamo za 1
  - Hranimo povprečne napake čez zadnjih 100 primerov za $k_c - 1$, k_c in $k_c + 1$
  - k_c posodobimo na vrednost z najnižjo napako
- Implicitno prilagajanje velikosti okolice C s ustrezno utežitveno funkcijo
  - Uporaba eksponentnega ali Gaussovega jedra