Uvod v meta učenje

Ljupčo Todorovski

Univerza v Ljubljani, Fakulteta za upravo
Institut Jožef Stefan, Odsek za tehnologije znanja (E8)

Februar 2021

Zakaj meta učenje?

Ker se učimo *kako se učiti*

Učimo se iz izkušenj strojnega učenja iz prejšnjih podatkovnih množic.

Kaj je rezultat meta učenja?

Izkušnje posplošimo v model za

- napovedovanje: Kateri algoritem naj uporabim na podani množici?
- napovedovanje: Katere nastavitve algoritma naj uporabim?
- pojasnjevanje: Kateri algoritem deluje kje oz. kdaj?

Pojasnjevanje

Vprašanje 4W: What Works Where and When
(under what circumstances)?

Je vprašanje 4W smiselno?

Ni, če bi obstajal univerzalno superioren algoritem

- A univerzalno superiornega algoritma za strojno učenje ni!
- Izrek o neobstoju brezplačnega kosila (*no free lunch theorem*)

Pregled vsebine

Izrek o neobstoju brezplačnega kosila

- Pričakovana testna napaka algoritma
- Dokaz in posledice za strojno in meta učenje
- Zakon o ohranjanju posplošitvene zmogljivosti

Splošni okvir za meta učenje

- Meta podatki: meta primeri in meta spremenljivke
- Naloge meta učenja: meta ciljne spremenljivke
- Meta atributi: vektorsko vpetje podatkovnih množic

Notacija

- $L : D_Y \times D_Y \to \{0, 1\}$ je funkcija izgube 0-1
- S: učna množica primerov $e = (\boldsymbol{x}, y = f(\boldsymbol{x}))$, f je ciljna funkcija

- $\mathcal{M}_A$: množica vseh možnih modelov (hipotez) m učnega algoritma A
- $P_A(m|S)$: posteriorna porazdelitev modelov $m \in \mathcal{M}_A$, rezultat učnega algoritma A na podatkovni množici S

- $\mathcal{F}$: množica vseh možnih ciljnih funkcij $f : D_1 \times D_2 \times \ldots \times D_p \to D_Y$
- $P(f|S)$: posteriorna porazdelitev ciljnih funkcij $f \in \mathcal{F}$

kar me zanima? Algoritem je toliko uspešen kolikor mu uspe zmanjšat napako!

Napaka: ... na letošnjem ppt-ju

Superioren algoritem A^* strojnega učenja

Za poljuben algoritem A velja

$$\|P_{A^*}(m|S) - P(f|S)\| \leq \|P_{A}(m|S) - P(f|S)\|$$

algoritem A^* — katerikoli drugi algoritem

Res je: razlika v zgornji formuli ni dobro definirana – NEREALISTIČNO ZA RAČUNAT (lohk sam ocenmo)

- Koliko dobro se izbrani model m približa ciljni funkciji f?
- Poskusimo bolje → lažje bo s funkcijo izgube

~~Zmogljivost algoritma~~ A: pričakovana testna napaka

Ocena napake algoritma

pričakovana napaka algoritma na izbrani množici S

$$\mathbb{E}_A[L|S] = \sum_{m \in \mathcal{M}_A} \sum_{f \in \mathcal{F}} \sum_{(\boldsymbol{x},y) \notin S} P(\boldsymbol{x})\, \mathbb{I}(f(\boldsymbol{x}) \neq m(\boldsymbol{x}))\, P_A(m|S)\, P(f|S)$$

ocena

prava vrednost funkcije v x

- Vsota čez vse možne pare (model/hipoteza m, ciljna funkcija f)
- In čez vse možne testne primere $(\boldsymbol{x}, y) \notin S$
- K vsoti prispevajo le napačno razvrščeni primeri $f(\boldsymbol{x}) \neq m(\boldsymbol{x})$
- Ker $\mathbb{I}(\cdot) = 0$ za pravilno razvrščene primere $f(\boldsymbol{x}) = m(\boldsymbol{x})$
- $P(\boldsymbol{x})$: apriorna porazdelitev vhodnega prostora $\boldsymbol{x} \in D_{\boldsymbol{X}}$

- Pozor: poznati moramo posteriorno porazdelitev ciljnih funkcij!

Predpostavka: deterministični učni algoritem A

$$\mathbb{E}_A[L|S] = \sum_{f \in \mathcal{F}} \sum_{(\boldsymbol{x},y) \notin S} P(\boldsymbol{x})\, \mathbb{I}(f(\boldsymbol{x}) \neq m(\boldsymbol{x}))\, P(m(\boldsymbol{x})|S)\, P(f|S)$$

Poenostavitve (16 min)

Algoritem A vrne le en model m^*

- Velja $P(m^*|S) = 1$ in $\forall m, m \neq m^* : P(m|S) = 0$
- Zato poenostavitev formule, kjer namesto m^* uporabljamo m
- $P(m(\boldsymbol{x})|S)$ je posteriorna porazdelitev vrednosti ciljne spremenljivke

Predpostavka: znana ciljna funkcija f

$$\mathbb{E}_A[L|f,S] = \sum_{(\boldsymbol{x},y)\notin S} P(\boldsymbol{x})\, \mathbb{I}(f(\boldsymbol{x}) \neq m(\boldsymbol{x}))\, P(m(\boldsymbol{x})|S)$$

Več možnih formulacij izreka NBK

Osnovna: Za poljuben par algoritmov A_1 in A_2

$$\sum_{f \in \mathcal{F}} (\mathbb{E}_{A_1}[L|f, S] - \mathbb{E}_{A_2}[L|f, S]) = 0$$

čez vse funkcije

poljuben par algoritmov

Alternative

- Pogosto: vsoto zamenjamo s povprečjem
- Ali pa s povprečjem čez vse apriorne porazdelitve $P(f)$

različne literature na različne načine to interpretirajo

Omejitev splošnosti izreka

Predpostavka: Boolove ciljne funkcije, $B = \{0, 1\}$

$$f : \underbrace{B \times B \times \ldots \times B}_{p \text{ krat}} \rightarrow B$$

- $\forall i : D_i = B$, $|D_1 \times D_2 \times \ldots \times D_p| = 2^p$
- Vseh možnih ciljnih funkcij je $|\mathcal{F}| = 2^{2^p}$

Izbor algoritmov A_1 in A_2 (brez škode za splošnost)

- A_1 vedno napoveduje 1, razen če se nauči drugače
- A_2 vedno napoveduje 0, razen če se nauči drugače

Poglejmo si primer Boolove ciljne funkcije f

	X_1	X_2	X_3	f	m_1	m_2
učni primeri $(\boldsymbol{x}, y) \in S$	0	0	0	1	1	1
	0	0	1	0	0	0
	0	1	0	1	1	1
testni primeri $(\boldsymbol{x}, y) \notin S$	0	1	1	0	1	0
	1	0	0	1	1	0
	1	0	1	0	1	0
	1	1	0	1	1	0
	1	1	1	1	1	0

- $Err(m_1|f) = 2/5 = 0.4$
- $Err(m_2|f) = 3/5 = 0.6$
- $Err(m_1|f) - Err(m_2|f) = -0.2$

In poglejmo še komplementarno funkcijo $\overline{f} = \neg f$

	X_1	X_2	X_3	$\overline{f}$	m_1	m_2
učni primeri $(\boldsymbol{x}, y) \in S$	0	0	0	0	0	0
	0	0	1	1	1	1
	0	1	0	0	0	0
testni primeri $(\boldsymbol{x}, y) \notin S$	0	1	1	1	1	0
	1	0	0	0	1	0
	1	0	1	1	1	0
	1	1	0	0	1	0
	1	1	1	0	1	0

- $Err(m_1|\overline{f}) = 3/5 = 0.6 = 1 - Err(m_1|f)$
- $Err(m_2|\overline{f}) = 2/5 = 0.4 = 1 - Err(m_2|f)$
- $Err(m_1|\overline{f}) - Err(m_2|\overline{f}) = 0.2 = -(Err(m_1|f) - Err(m_2|f))$

Za poljubni model m torej velja

$$Err(m|f) + Err(m|\overline{f}) = 1$$

Zato tudi

$$[Err(m_1|f) - Err(m_2|f)] + \left[Err(m_1|\overline{f}) - Err(m_2|\overline{f})\right] = 0$$

- Za poljubno (Boolovo) ciljno funkcijo f in njen komplement $\overline{f} = \neg f$
- Za poljubna modela m_1 in m_2 oziroma algoritma A_1 in A_2

Če seštejemo za vse možne $f \in \mathcal{F}$

$$\begin{aligned} 0 &= \sum_{f \in \mathcal{F}} [Err(m_1|f) - Err(m_2|f)] + \left[Err(m_1|\overline{f}) - Err(m_2|\overline{f})\right] \\ &= 2 \sum_{f \in \mathcal{F}} Err(m_1|f) - Err(m_2|f) \end{aligned}$$

V prvi vsoti smo vsako ciljno funkcijo f upoštevali dva krat.

Ker je izbor A_1 in A_2 poljuben, smo dokazali NBK

$$\sum_{f \in \mathcal{F}} (\mathbb{E}_{A_1}[L|f,S] - \mathbb{E}_{A_2}[L|f,S]) = 0$$

Zakon o ohranitvi posploševalne zmogljivosti algoritma

20 min

Za vsak algoritem strojnega učenja je vsota njegovih zmogljivosti čez vse možne ciljne funkcije $f \in \mathcal{F}$ nespremenljiva.

Za primerjavo algoritmov

Izjave oblike A_1 je bolj zmogljiv od A_2

- Oziroma A_1 ima manjšo pričakovano napako od A_2
- Morajo vedno sloneti na predpostavki o ciljni funkciji $f \in \mathcal{F}$
- Ali predpostavki o apriorni in posteriorni porazdelitvi $p(f)$ in $p(f|S)$

Za teorijo (razvoj algoritmov) strojnega učenja

- Ne obstaja superiorni algoritem strojnega učenja
- Večina teoretičnih rezultatov o možnosti učenja je negativnih

Za prakso strojnega učenja

- Ne glede na popularnost ali teoretično podprtost algoritma za strojno učenje, lahko najdemo ciljno funkcijo, za katero bo njegova napaka velika (in zmogljivost majhna)

- Ekspertiza omejena na en razred algoritmov, čeprav zelo močnih, ne zadostuje za uspešno napovedno modeliranje
- Izkušnje z uporabo širokega nabora algoritmov so zelo pomembne pri reševanju novega problema

Šibka in močna predpostavka strojnega učenja

Proces nastajanja problemov strojnega učenja ustvarja neenakomerno porazdelitev ciljnih funkcij $P(f)$ čez $\mathcal{F}$.

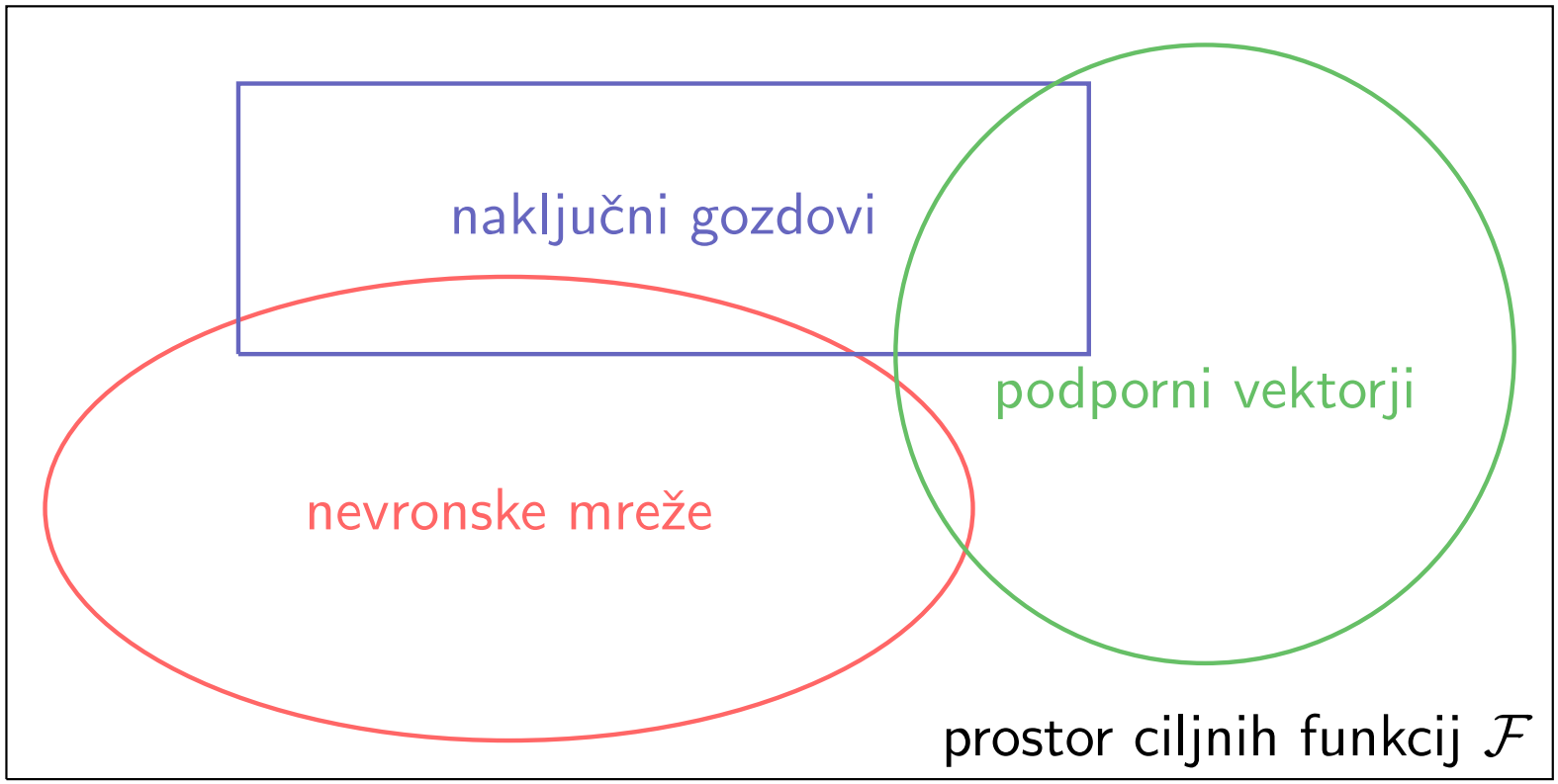
Porazdelitev $P(f)$ čez $f \in \mathcal{F}$ je znana vsaj v obliki uporabnega približka.

ŠIBKA PREDPOSTAVKA: predpostavt da so vse funkcije enako verjetne ni ok. Ena funkcije so lohke bolj verjetne (se bolj pogosto pojavljajo)

MOČNA PREDPOSTAVKA: lahko izracunamo kaksna je porazdelitev čez ciljne funkcije

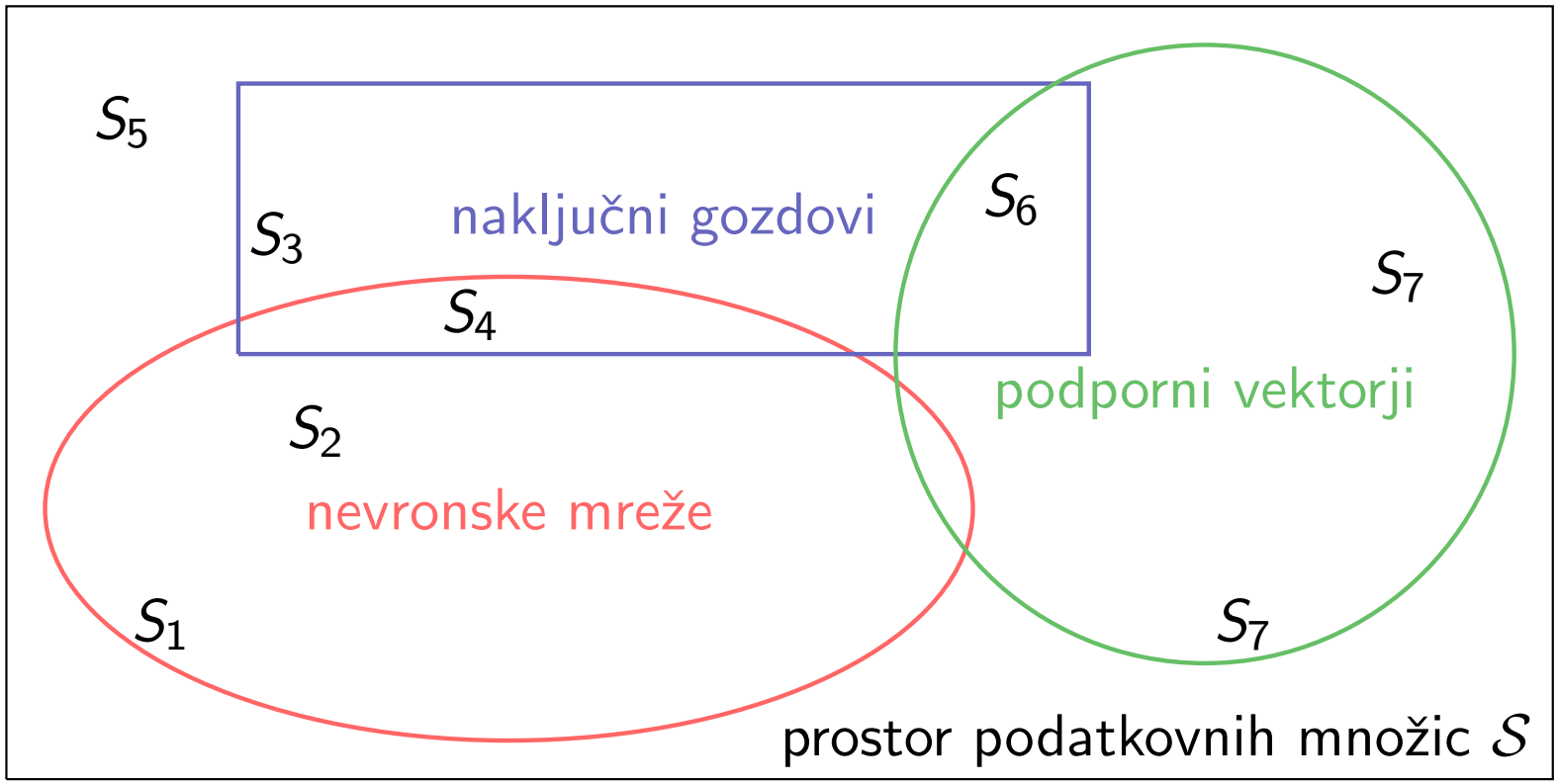
Predpostavka meta učenja: prostor ciljnih funkcij $\mathcal{F}$

Algoritmi imajo svoja "področja ekspertize" v prostoru ciljnih funkcij $\mathcal{F}$.



Predpostavka meta učenja: prostor podatkov $\mathcal{S}$

Algoritmi imajo svoja "področja ekspertize" v prostoru množic podatkov $\mathcal{S}$.



Meta podatki

Meta primeri

- Podatkovne množice $S \in \mathcal{S}$
- $\mathcal{S}$ je (neskončni) prostor *vseh* možnih podatkovnih množic

Meta spremenljivke

- Meta ciljne spremenljivke: različne naloge meta učenja
- Meta atributi: vektorski opis podatkovne množice

Napovedovanje zmogljivosti algoritmov iz $\mathcal{A}$

Ciljne spremenljivke $\boldsymbol{Y} = p(A, S)|_{A \in \mathcal{A}}$

- Algoritem A, običajno z znanimi nastavitvami parametrov θ
- p je način vrednotenja zmogljivosti, npr. 10-kratno prečno preverjanje
- $D_Y \subseteq R$, regresijska naloga nadzorovanega učenja

Meta model je več-ciljni regresijski, $m : \mathcal{S} \rightarrow \mathbb{R}^{|\mathcal{A}|}$

Lahko bi uporabili več navadnih modelov, po enega za vsak $A \in \mathcal{A}$.

Izbira najbolj zmogljivega algoritma

Ciljna spremenljivka $Y = \arg\max_{A \in \mathcal{A}} p(A, S)$

- Najbolj zmogljiv algoritem A^* za S, $A^* = \arg\max_{A \in \mathcal{A}} p(A, S)$
- Algoritem A, običajno z znanimi nastavitvami parametrov θ
- p je način vrednotenja zmogljivosti, npr. 10-kratno prečno preverjanje
- $D_Y = \mathcal{A}$, klasifikacijska naloga nadzorovanega učenja

Meta model je klasifikacijski, $m : \mathcal{S} \to \mathcal{A}$

Lahko bi uporabili tudi meta model za napovedovanje zmogljivosti.

Priporočanje/rangiranje algoritmov

Ciljne spremenljivke $\boldsymbol{Y} = \mathit{rangiranje}(\mathcal{A})$

- Rangiranje algoritmov iz A glede na zmogljivost na S:
  $A_{j_1} > A_{j_2} > \ldots > A_{j_{|\mathcal{A}|}}$, kjer velja
  $p(A_{j_1}, S) \geq p(A_{j_2}, S) \geq \ldots p(A_{j_{|\mathcal{A}|}}, S)$
- $j_1, j_2, \ldots j_{|\mathcal{A}|}$ permutacija naravnih števil $1 \ldots |\mathcal{A}|$
- p je način vrednotenja zmogljivosti, npr. 10-kratno prečno preverjanje

Meta model je več-ciljni regresijski $m : \mathcal{S} \rightarrow \mathbb{R}^{|\mathcal{A}|}$

Lahko bi uporabili tudi meta model za napovedovanje zmogljivosti.

Učenje iz prejšnjih modelov naučenih z algoritmom A

Učenje prenosa, *Transfer Learning*

- Prejšnje modele $\{A(S) : S \in \mathcal{S}_{train}\}$ uporabimo kot osnovo za učenje modela $A(S_{new})$ na novi (podobni) množici $S_{new} \notin \mathcal{S}_{train}$
- A lahko nastavimo tako, da bo nov model podoben prejšnjim
- Umetne nevronske mreže: strukturo in uteži mreže prej naučene na podobni množici S uporabimo za učenje iz S_{new}

Večopravilno učenje, *Multi-Task Learning*

Učenje iz množice podobnih podatkovnih množic, kjer prej naučene modele uporabimo kot pristranskost/predsodek pri učenju novih modelov.

Optimalne nastavitve algoritma A

Ciljne spremenljivke $\boldsymbol{Y} = \arg\max_{\theta \in \Theta_A} p(A, \theta, S)$

- Optimalna nastavitev parametrov θ^* za algoritem A na S
- $\theta^* = \arg\max_{\theta \in \Theta_A} p(A, \theta, S)$
- p je način vrednotenja zmogljivosti, npr. 10-kratno prečno preverjanje
- $D_Y = \Theta_A$

Meta model je več-ciljni regresijski $m : \mathcal{S} \rightarrow \Theta_A$

Na naslednjih predavanjih bomo to nalogo formulirali kot optimizacijski problem, AutoML.

Definicija meta atributa X_{meta}

Računa neko lastnost podatkovne množice S

$$X_{meta} : \mathcal{S} \rightarrow \mathbb{R}$$

1. Ročno načrtovan/izbran: statistična lastnost podatkovne množice
2. Avtomatsko izračunan: dimenzija prostora vpetja podatkovnih množic

Omejitev

Nizka računska kompleksnost izračuna.

Kategorije ročno načrtovanih meta atributov

1. Osnovne: število primerov, spremenljivk in podobno
2. Statistične: statistike izmerjene na numeričnih spremenljivkah
3. Informacijske: količina informacije v diskretnih spremenljivkah
4. Mere kompleksnosti: izmerjene na podatkovni množici
5. Modelske: opis modela naučenega na podatkovni množici
6. Algoritmične: zmogljivost preprostih algoritmov

Osnovni za primere in atribute

Primeri podatkovne množice S

$|S|$: število primerov v podatkovni množici

Atributi podatkovne množice S

- p: število atributov
- $p_n = |\{D_i : D_i = \mathbb{R}\}|$: število numeričnih atributov
- $p_d = p - p_n$: število diskretnih atributov
- $p_b = |\{D_i : |D_i| = 2\}|$: število binarnih atributov
- $|S|/p$: število primerov na atribut
- $p/|S|$: število atributov na primer

Osnovni za ciljno spremenljivko

Za regresijske probleme

- Značilke porazdelitve vrednosti D_Y
- Primeri: povprečje, mediana, standardna deviacija
- Glej naslednjo prosojnico

Za klasifikacijske probleme

- Značilke porazdelitve vrednosti D_Y
- $|D_Y|$: število razredov
- $\max_{v \in D_Y} |\{(\boldsymbol{x}, y) \in S : y = v\}|$: delež primerov v največjem razredu

Posamezni (numerični) atributi

- *median*, *mean*: lokacijski parametri porazdelitev
- *min*, *max*, Q_1, Q_3: parametri razpona
- $max - min$, $Q_3 - Q_1$, σ: statistike razpona
- *kurtosis*, *skewness*: statistike oblike porazdelitve
- $na = |\{e \in S : X_i(e) = NA\}|$, $na/|S|$: število, delež neznanih vrednosti

Agregati: *min*, *max*, *mean*, σ, histogrami

Dva ali več (numeričnih) atributov

- $corr(X_i, X_j), cov(X_i, X_j)$: korelacija in kovarianca
- $ncorr = \sum_{i=1}^{p} \sum_{j=i+1}^{p} \mathbb{I}(|corr(X_i, X_j)| > 0.5)$, $ncorr/(p(p-1)/2)$: število, delež paroma koreliranih atributov
- $nn = \sum_{i=1}^{p} \mathbb{I}(isN(X_i))$, nn/p: število/delež normalno porazdeljenih atributov, eno vzorčni test Kolmogorov-Smirnov
- PCA-λ: lastne vrednosti kovariančne matrike za numerične atribute
- PCA-95%: število glavnih komponent, ki pojasni vsaj 95% variance

Numerični atributi in ciljna spremenljivka: klasifikacija

Centri gravitacije $\boldsymbol{x}_v$ za razrede $v \in D_Y$

$$\boldsymbol{x}_v = \frac{1}{|S_v|} \sum_{(\boldsymbol{x},y)\in S: y=v} \boldsymbol{x}$$

- Centroid $\boldsymbol{x}_v$ primerov iz razreda v
- Opazujemo lahko Evklidske razdalje med centri $\boldsymbol{x}_v$ in $\boldsymbol{x}_u$ za $u, v \in D_Y : u \neq v$

Numerični atributi in ciljna spremenljivka: regresija

- Primerjaj s prosojnico za dva ali več (numeričnih) atributov.

- $corr(X_i, Y), cov(X_i, Y)$: korelacija/kovarianca s ciljno spremenljivko
- učinkovitost atributa X_i: število/delež primerov, ki jih moram izbrisati, da bi $corr(X_i, Y) > 0.9$
- kolektivna učinkovitost atributov: število preostalih primerov po iterativnem brisanje primerov, ki imajo ostanke večje od 0.1

Posamezni (diskretni) atributi

- $H(X_i) = -\sum_{v \in D_i} P(v) \log_2 P(v)$: entropija (nečistost)
- $H(X_i)/\log_2 |S|$: količina informacije

Diskretni atributi in ciljna spremenljivka: klasifikacija

- $H(Y)$: entropija ciljne spremenljivke

$$
\begin{aligned}
MI(X,Y) &= H(X) + H(Y) - H(X,Y) \\
H(X,Y) &= -\sum_{(v_x,v_y)\in(D_X,D_Y)} p(v_x,v_y)\log_2 p(v_x,v_y)
\end{aligned}
$$

- $MI(X_i,Y), MI(X_i,Y)/H(Y)$: vzajemna informacija
- $H(X_i,Y), H(X_i,Y)/H(Y)$: skupna entropija

- $H(Y)/(\sum_{i=1}^{p} MI(X_i,Y)/p)$: lastna dimenzionalnost

Dimenzija podatkovne množice

- PCA-95%: število glavnih komponent, ki pojasni vsaj 95% variance
- Fraktalna dimenzija podatkovne množice
- Notranja dimenzija podatkovne množice

Pozor: računska kompleksnost!

Osnovna ideja

Opazujemo lastnosti napovednega modela $m = A(S)$

- In ne lastnosti množice S
- Za poljubno izbrani algoritem A

Pogosto uporabljeni algoritmi

- Odločitvena drevesa
- Linearni modeli

Odločitvena drevesa

Zgradimo odločitveno drevo brez predhodnega ali naknadnega rezanja.

Opazovane lastnosti

- število vseh, notranjih ali končnih vozlišč v drevesu
- globina drevesa
- povprečna globina končnih vozlišč
- zmanjševanje nečistosti v korenskem vozlišč

- število učnih primerov v končnih vozliščih
- število končnih vozlišč za posamezno vrednost iz D_Y (klasifikacija)
- število vozlišč v posameznem nivoju drevesa

Drugi modeli

Metoda podpornih vektorjev

- Izbrano jedro običajno polinomsko
- Število podpornih vektorjev

Linearni modeli

Število koeficientov značilno različnih od 0

Osnovna ideja: orientirji (*landmarks*)

Meta spremenljivke $p(A, S)$

- Izbor A: hitri, preprosti algoritmi (orientirji)
- Izbor p: zmogljivost ocenjena na učni množici (hitrost!)

Enostavni algoritmi

Najbližji sosed

$k = 1$: metoda najbližjega soseda

Linearni model

Linearna oziroma logistična regresija

Odločitvena drevesa

- Štor: odločitveno drevo z enim notranjim vozliščem
- Naključni štor: Odločitveno drevo z naključno izbranim enim notranjim vozliščem.

Relativni in vzorčni orientirji

Relativni orientir $p(A_1, S) - p(A_2, S)$

Za dva izbrana orientirja A_1 in A_2.

Vzorčni orientirji

- Izbor algoritmov A lahko širši (ne le preprosti)
- Hitrost zagotovimo tako, da jemljemo majhne vzorce $V : |V| \ll |S|$
- Zaporedje vzorcev naraščajoče velikosti $|V|$

Literatura in praktični napotki

Priporočena literatura

- (Wolpert 1996): izrek o neobstoju zastonjskega kosila
- (Vanschoren 2018): Splošni okvir za meta učenje
- (Rivolli in ost. 2018, Lorena in ost. 2018): meta atributi

Programska oprema in viri za meta učenje

- CRAN paket *mfe* za izračun meta atributov in pripadajoča vadnica *cran.r-project.org/web/packages/mfe/vignettes/mfe-vignette.html*
- Spletni repozitorij *openml.org* in CRAN paket *OpenML*