Meta učenje

Ljupčo Todorovski

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

Marec 2019

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 novi množici?
- pojasnjevanje: Kateri algoritem deluje kje oz. kdaj?

Pojasnjevanje

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

Je vprašanje 4W smiselno?

Ni, če obstaja 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 spremenljivke

Opis 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
- $\mathcal{H}$: množica vseh možnih hipotez (modelov) h učnega algoritma
- $\mathcal{F}$: množica vseh možnih ciljnih funkcij $f : D_{\boldsymbol{X}} \to D_Y$, $D_{\boldsymbol{X}} = D_{X_1} \times D_{X_2} \times \ldots \times D_{X_p}$
- $P(h|S)$: posteriorna porazdelitev hipotez $h \in \mathcal{H}$, pravzaprav rezultat učnega algoritma na podatkovni množici S
- $P(f|S)$: posteriorna porazdelitev ciljnih funkcij $f \in \mathcal{F}$

- Ali obstaja algoritem, ki je superioren čez vse $f \in \mathcal{F}$?

Zmogljivost algoritma: pričakovana testna napaka

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

- vsota čez vse možne pare (hipoteza h, ciljna funkcija f)
- in čez vse možne testne primere $(\boldsymbol{x}, y) \notin S : f(\boldsymbol{x}) \neq h(\boldsymbol{x})$
- $\mathbb{I} : \{\top, \bot\} \rightarrow 0, 1,\ \mathbb{I}(\top) = 1,\ \mathbb{I}(\bot) = 0$
- $P(\boldsymbol{x})$: apriorna porazdelitev vhodnega prostora $\boldsymbol{x} \in D_{\boldsymbol{X}}$

- Pozor: poznati moramo posteriorno porazdelitev ciljnih funkcij!

Predpostavka: deterministični algoritem

Deterministični učni algoritem vrne en model h^*

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

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

Pričakovana testna napaka za znano ciljno funkcijo f

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

Ena možna verzija izreka NBK

Za poljuben par algoritmov A_1 in A_2 velja

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

- Pogosta formulacija: vsoto zamenjamo s povprečjem
- Alternativna verzija: povprečje čez vse apriorne porazdelitve $P(f)$

Omejitev splošnosti dokaza

Predpostavka: Boolove ciljne funkcije

$$f : \{0,1\}^p \rightarrow \{0,1\}$$

- $D_{\boldsymbol{X}} = \{0,1\}^p$, $|D_X| = 2^p$
- $|\mathcal{F}| = 2^{2^p}$

Izbor algoritmov A_1 in A_2 (brez škode za splošnost)

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

Poglejmo si primer Boolove ciljne funkcije f

	$\boldsymbol{x} \in \{0,1\}^3$			f	h_1	h_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

- $Err(h_1|f) = 2/5 = 0.4,\ Err(h_2|f) = 3/5 = 0.6$
- $Err(h_1|f) - Err(h_2|f) = -0.2$

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

	$\boldsymbol{x} \in \{0,1\}^3$			$\overline{f}$	h_1	h_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

- $Err(h_1|\overline{f}) = 3/5 = 0.6,\ Err(h_2|\overline{f}) = 2/5 = 0.4$
- $Err(h_1|\overline{f}) - Err(h_2|\overline{f}) = 0.2$

Za poljubno hipotezo h torej velja

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

Zato velja tudi

$$(Err(h_1|f) - Err(h_2|f)) + (Err(h_1|\overline{f}) - Err(h_2|\overline{f})) = 0$$

- za poljubno (Boolovo) ciljno funkcijo f in njen komplement $\overline{f} = \neg f$
- za poljubni hipotezi h_1 in h_2 (algoritma A_1 in A_2)

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

$$\begin{aligned} 0 &= \sum_{f \in \mathcal{F}} ((Err(h_1|f) - Err(h_2|f)) + (Err(h_1|\bar{f}) - Err(h_2|\bar{f}))) \\ &= 2 \sum_{f \in \mathcal{F}} (Err(h_1|f) - Err(h_2|f)) \end{aligned}$$

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

Če upoštevamo arbitrarni izbor A_1 in A_2, smo dokazali NBK

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

Zakon o ohranitvi posplošitvene zmogljivosti

Za vsak učni algoritem je vsota 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 strojnega učenja

- Ne obstaja "ultimativni" 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 predpostavka

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

Močna predpostavka in posledica

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

Posledica, izpeljava na tabli

Poznavanje $P(f)$, t.j., verjetnosti ciljne funkcije f, je ekvivalentno poznavanju posteriorne porazdelitve vrednosti ciljne spremenljivke $P(Y = y|e)$ za podan primer $e = (\boldsymbol{x}, y)$.

Meta primeri

Meta primer ustreza podatkovni množici

- Množica podatkovnih množic $\mathcal{S} = \{S_1, S_2, \ldots S_n\},\ S_i \in \mathcal{E}_i$
- $\mathcal{E}_i = D_{X_{1i}} \times D_{X_{2i}} \times \ldots D_{X_{p_i i}} \times D_{Y_i} = D_{\boldsymbol{X}_i} \times D_{Y_i}$

- Običajna omejitev: vse ciljne spremenljivke Y_i primerljivega tipa; vse numerične ali vse diskretne

Meta spremenljivke

1. Opis podatkovnih množic
2. Obnašanje algoritmov: zmogljivost na podatkovnih množicah iz $\mathcal{S}$
   - Množica algoritmov $\mathcal{A} = \{A_1, A_2, \ldots A_m\}$
   - Množica konfiguracij algoritmov Θ

Konfiguracija algoritma $\theta_j = (A_j, \phi_j)$

- Opredelitev algoritma A_j
- Nastavitev vrednosti njegovih parametrov ϕ_j

Meta spremenljivke: opis podatkovnih množic

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

Kategorije

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

Meta spremenljivke: obnašanje algoritmov

$$a_{meta} : \Psi \times \mathcal{S} \rightarrow \mathbb{R}$$

Zmogljivost algoritma $\Psi = \mathcal{A}$

Metoda za merjenje zmogljivosti $p : \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$

Zmogljivost konfiguracije algoritma $\Psi = \Theta$

Metoda za merjenje zmogljivosti $p : \Theta \times \mathcal{S} \rightarrow \mathbb{R}$

Napovedovanje zmogljivosti algoritma A

Spremenljivke

- Vhodne: opis podatkovne množice S, $D_{\textbf{meta}}$
- Izhodna: zmogljivost algoritma A na S, $p(A, S)$

Meta model (regresijski)

$$m : D_{\textbf{meta}} \rightarrow \mathbb{R}$$

Več-ciljna regresija, če napovedujemo zmogljivost več algoritmov iz $\mathcal{A}$.

Izbira najbolj zmogljivega algoritma

Spremenljivke

- Vhodne: opis podatkovne množice S, $D_{\textbf{meta}}$
- Izhodna: najbolj zmogljiv algoritem na S, $A^* = \arg\max_{A \in \mathcal{A}} p(A, S)$

Meta model (klasifikacijski)

$$m : D_{\textbf{meta}} \rightarrow \mathcal{A}$$

Lahko bi uporabili tudi meta modele za napovedovanje zmogljivosti.

Priporočanje/rangiranje algoritmov

Spremenljivke

- Vhodne: opis podatkovne množice S, $D_{\textbf{meta}}$
- Izhodna: rangiranje algoritmov glede na zmogljivost na S, $A_{j_1} > A_{j_2} > A_{j_m}$, kjer velja $p(A_{j_1}, S) \geq p(A_{j_2}, S) \geq \ldots p(A_{j_m}, S)$

Meta model

$$m : D_{\textbf{meta}} \rightarrow S(\mathcal{A})$$

Rabimo učni algoritem, ki lahko napoveduje rangiranje; lahko bi uporabili tudi meta modele za napovedovanje zmogljivosti. $S(\mathcal{A})$ je množica vseh permutacij elementov množice $\mathcal{A}$.

Učenje iz prejšnjih modelov naučenih z algoritmom A

Učenje prenosa (*Transfer Learning*)

- Prejšnje modele $\{A(S) : S \in \mathcal{S}\}$ uporabimo pri gradnji modela $A(S_{new})$ na novi (podobni) množici $S_{new} \notin \mathcal{S}$
- 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 pri učenju novih modelov.

Optimalna konfiguracija algoritma A

$$\max_{(A,\phi)\in\Theta} p((A,\phi), S_{new})$$

Problem numerične optimizacije

Pri reševanju problema si pomagamo s podatki o optimalnih nastavitvah parametrov algoritma A na množicah $S \in \mathcal{S}$.

Optimalna konfiguracija

$$\max_{\theta \in \Theta} p(\theta, S_{new})$$

Problem numerične in celoštevilčne optimizacije

- Izbiramo hkrati optimalni algoritem A in optimalno nastavitev njegovih parametrov ϕ
- Pri tem si pomagamo s podatkih o optimalnih konfiguracijah θ na množicah $S \in \mathcal{S}$

Osnovne meta spremenljivke

Primeri

$|S|$: število primerov v podatkovni množici

Atributi (vhodne spremenljivke)

- 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
- $|S|/p$, $p/|S|$: število primerov na atribut, število atributov na primer

Razredi, le za klasifikacijske probleme

$|D_Y|$: število razredov, za klasifikacijske probleme

Posamezni 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 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

Za klasifikacijo

Center gravitacije za vsako vrednost $v \in D_Y$

- Centroid $\boldsymbol{x}_v$ primerov iz $S_v = \{(\boldsymbol{x}, y) \in S : y = v\}$

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

- Opazujemo Evklidske razdalje med centri $\boldsymbol{x}_v$ in $\boldsymbol{x}_u$ za $u, v \in D_Y, u \neq v$

Za regresijo

- Vse statistike za posamezne atribute uporabimo na ciljni spremenljivki

- $corr(X_i, Y), cov(X_i, Y)$: 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

Za posamezne atribute

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

Za klasifikacijo

- $H(Y)$: 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

Osnovna ideja

Opazujemo lastnosti napovednega modela $A(S)$

- In ne lastnosti množice S
- Za 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: prečno preverjanje za merjenje napake/točnosti

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 orientirji

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

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 $S_v : |S_v| \ll |S|$
- Zaporedje vzorcev naraščajoče velikosti $|S_v|$

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 spremenljivke

Programska oprema in viri za meta učenje

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