

dataset2vec in AutoML

Ljupčo Todorovski

Univerza v Ljubljani, Fakulteta za matematiko in fiziko
Institut Jožef Stefan, Odsek za tehnologije znanja (E8)

Marec 2023

Pregled vsebine

1 dataset2vec

- Splošno o vpetjih objektov v vektorski prostor
- Podatkovne množice in naloga napovedovanja
- NM za vpetje podatkovnih množic

2 AutoML

- Motivacija
- SMBO in osnove Bayesove optimizacije
- Nadgradnje osnovnega SMBO

Vpetje, *embedding*: definicija

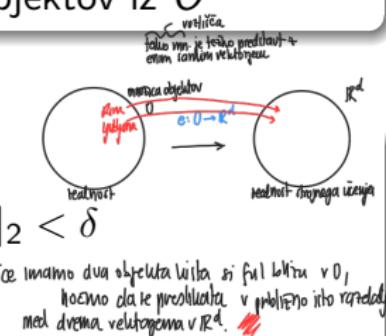


Vpetje je funkcija $e : O \rightarrow \mathbb{R}^d$

- O : množica objektov, običajno kompleksnega, ne-vektorskoga tipa
- d je razsežnost vektorskoga prostora za vpetje objektov iz O

Ključna lastnost funkcije e

$$\text{dist}(o_1, o_2) < \varepsilon \implies \|e(o_1) - e(o_2)\|_2 < \delta$$



- $\text{dist} : O \times O \rightarrow \mathbb{R}$ je mera razdalje med objekti iz O
- $\|\cdot\|_2$ je Evklidska norma v vektorskem prostoru \mathbb{R}^d
- **Podobni objekti so vpeti v podobne vektorje**

$e(\text{man}) - e(\text{woman}) \dots$ približujemo da $\approx e(\text{king}) - e(\text{queen})$

če nam sestaljeno ugotovljamo če sta si blizu besedi, na momra (na podlagi če + besedila Valda ne momra)

Primer vpetja word2vec

embedding gleda pomensko podobnost
(in ne sintaktično)

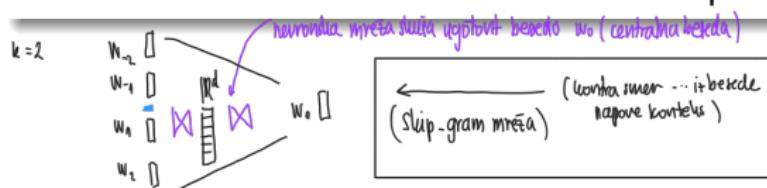
Osnovna opredelitev word2vec

- Elementi O so besede, $dist$ je pomenska podobnost besed
- Znani primer $e(\text{queen}) - e(\text{woman}) \approx e(\text{king}) - e(\text{man})$

Pot do vpetja word2vec: nalogi napovedovanja

- Osnovna: iz besed v kontekstu napovedati centralno besedo
- Alternativna: iz centralne besede napovedati besede iz konteksta

Vremem dve besedi prej in dve besedi pozneje in normo centralno besedo.



Znamo nas $e(w_0)$??

Vhodni in učni podatki za word2vec

Iz podanega, vhodnega zaporedja besed

$$w_1, w_2, w_3, w_4, w_5, w_6, \dots, w_{n-4}, w_{n-3}, w_{n-2}, w_{n-1}, w_n$$

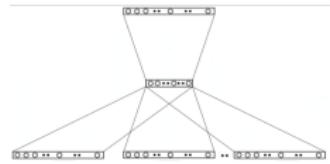
Zgradimo učno množico za NM (dolžina konteksta $k = 2$)

X_{-2}	X_{-1}	X_0	X_1	X_2
w_1	w_2	w_3	w_4	w_5
w_2	w_3	w_4	w_5	w_6
\vdots				\vdots
w_{n-4}	w_{n-3}	w_{n-2}	w_{n-1}	w_n

Drseče okno dolžine $2 \cdot k + 1 = 5$ za zajemanje konteksta.

Nevronski mreži za word2vec

CBOW: atributi X_{-2}, X_{-1}, X_1, X_2 , ciljna spremenljivka X_0



Skip-Gram: atribut X_0 , ciljne spremenljivke X_{-2}, X_{-1}, X_1, X_2



V obeh primerih vpetje besede preberemo iz stanj skritih nevronov.

Objekti so podatkovne množice

Torej v tem primeru velja $O = \mathbb{D}$

Množica O je podatkovni prostor, poglej prejšnja predavanja.

Funkcija podobnosti $dist$ in naloga napovedovanja

? ?

Naloga napovedovanja: ali sta dve množici podobni?

na vhodu vstane dva data seta
 Pomožna meta naloga $meta_Y : \mathbb{D} \times \mathbb{D} \rightarrow \{\ominus, \oplus\}$

- Napovedujemo podobnost med dvema podatkovnima množicama
- Meta naloga dvojiške klasifikacije $\tilde{Y} : D_{\tilde{Y}} = \{\ominus, \oplus\}$
- $\tilde{y} = \oplus$, če sta množici podobni, $\tilde{y} = \ominus$ sicer

Predmem dve množici S_1 in S_2 in nalaganju hočem vrednoti v enem samem mestu $(0$ ali 1) or ($+$ ali $-$) in bo povedal če sta množici podobni ali ne.

$$\begin{matrix} S_1 \\ \sqcup \\ S_2 \end{matrix} \quad \boxed{\quad} \quad \begin{matrix} \ominus \\ \oplus \end{matrix}$$

Ideja: podobni množici sta dva naključna vzorca iste podatkovne množice.

upamo da je nas pogaj
velja platit teden
komo odobrati e(m)

Kaj je naključni vzorec $\text{sample}(S)$ podatkovne množice S ?

S	$X_1 \ X_2 \ \dots \ X_p$	$Y_1 \ Y_2 \ \dots \ Y_q$	
	vhodne sprem.	izhodne/alične sprem.	
e_1	$x_{11} \ x_{12} \ \dots \ x_{1p}$	$y_{11} \ y_{12} \ \dots \ y_{1q}$	ponavljajoča se + eno samo ciljno sprem. (Y) (tukaj jih je q)
e_2	$x_{21} \ x_{22} \ \dots \ x_{2p}$	$y_{21} \ y_{22} \ \dots \ y_{2q}$	
\vdots	$\vdots \ \vdots \ \ddots \ \vdots$	$\vdots \ \vdots \ \ddots \ \vdots$	
e_n	$x_{n1} \ x_{n2} \ \dots \ x_{np}$	$y_{n1} \ y_{n2} \ \dots \ y_{nq}$	

Naključni vzorci $V \neq \emptyset$ brez ponavljanja za

- Primere $V_e \subseteq \{e_1, e_2, \dots, e_n\}$
- Vhodne spremenljivke $V_X \subseteq \{X_1, X_2, \dots, X_p\}$
- Ciljne spremenljivke $V_Y \subseteq \{Y_1, Y_2, \dots, Y_q\}$, če bi jih bilo $q > 1$

Kako je podatkovne mn. S podeljeno do vzorcev (vzročijev)? Vzorimo klic ali primere al vhodne sprem. al izhodne sprem.

Gradnja enega primera ŕ je meta podatkovne množice \tilde{S}

VZORČENJE META PRIMEROV

Require: \mathbb{S} je množica podatkovnih množic

Ensure: \tilde{e} je meta primer za dataset2vec

function META_EXAMPLE(\mathbb{S})

$S_1 \leftarrow \text{choose}(\mathbb{S})$

if $U[0, 1] < 0.5$ **then**

$S_2, \tilde{y} \leftarrow \text{choose}(\mathbb{S} \setminus \{S_1\}), \ominus$

else

$S_2, \tilde{y} \leftarrow S_1, \oplus$

end if

return $\tilde{e} = (\text{sample}(S_1), \text{sample}(S_2), \tilde{y})$

end function

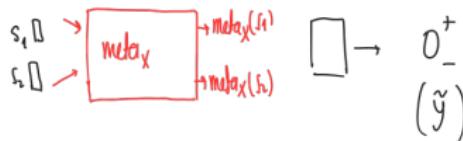
min. podatkovnih množic

na umeten način podeljujemo
toda primerov bo lejih zelel

če izberemo dve isti mn. \rightarrow razglašimo pravljica
poslovca
 $\neg\neg$ - vrednost $\neg\neg \Rightarrow \neg\neg$ negativna

Meta model \tilde{m}

$$\tilde{m}(S_1, S_2) = P(S_1 \text{ je podobna } S_2) = e^{-\gamma \|meta_X(S_1) - meta_X(S_2)\|}$$

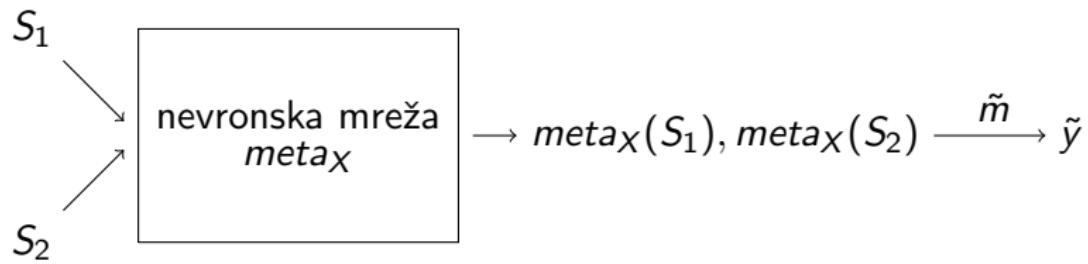


Funkcija izgube L za meta primer (S_1, S_2, \tilde{y})

funkcija izgube ko imamo pozitiven primer

$$L(S_1, S_2) = \begin{cases} \log \tilde{m}(S_1, S_2) & \text{če } \tilde{y} = \oplus \\ 1 - \log \tilde{m}(S_1, S_2) & \text{če } \tilde{y} = \ominus \end{cases}$$

Reševanje naloge napovedovanja



Struktura nevronske mreže *metax*

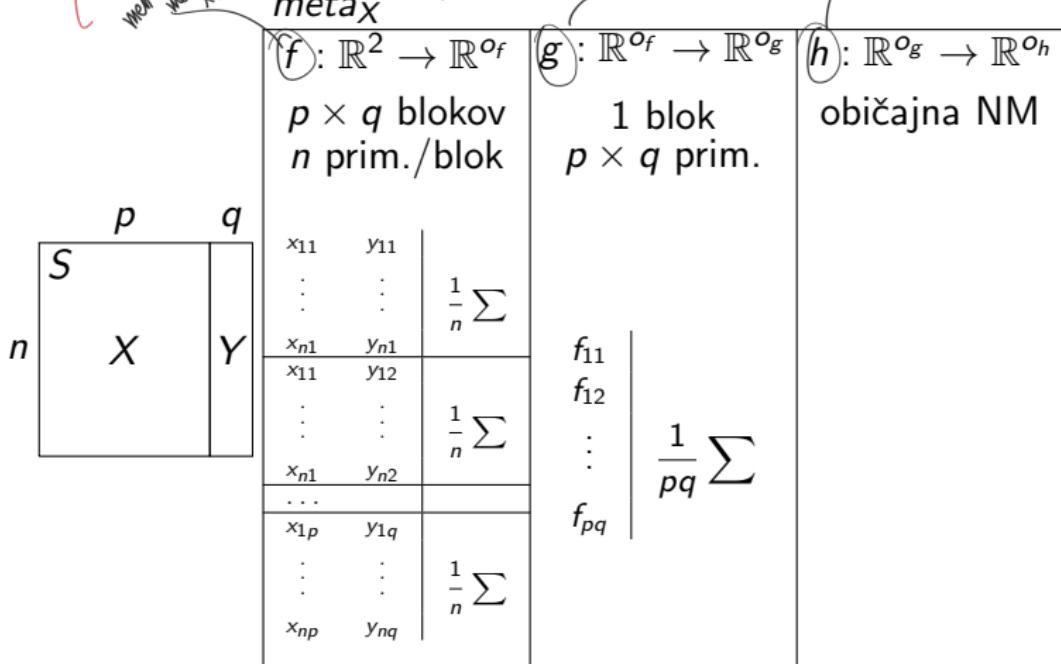
to napisano na
nastavku slajdu

med korelacijo
med zvezko spred
 X (vločna) in Y (predvoda)

nevronska mreža izdrena sprem.

poteka da korelacija
zmerja v veliki
stevetu

poteka da je
spred koren velik
(tu je ta embedding)
"vpetje"



2. bloku vreme te korelacijs in jih zemora u eno sumo ţenika

v tem 1. bloku se želite naučiti korelacijs med X in Y

Struktura nevronskega mrež f in g

$$g_{out} = \frac{1}{pq} \sum_{k=1}^q \sum_{j=1}^p g \left(\frac{1}{n} \sum_{i=1}^n f(x_{ij}, y_{jk}) \right)$$

Nevronska mreža f

- Vhodi (x_{ij}, y_{ik}) , $i = 1 \dots n$, $j = 1 \dots p$, $k = 1 \dots q$.
- Modelira korelacijo med X_j in eno ciljno spremenljivko Y_k

Nevronska mreža g

- Vhod so povprečja izhodov f : $\frac{1}{n} \sum_{i=1}^n f(x_{ij}, y_{jk})$
- Modelira povprečne korelacije in interakcije med spremenljivkami

Obe NM *deepSet*, t.j., NM za obravnavo množic.

Dva načina izračuna vpetja za podano množico S

Na celotni množici

$$\text{meta}_X(S) = h \left(\frac{1}{pq} \sum_{k=1}^q \sum_{j=1}^p g \left(\frac{1}{n} \sum_{i=1}^n f(x_{ij}, y_{jk}) \right) \right)$$

Na množici \mathbb{V} naključnih vzorcev S

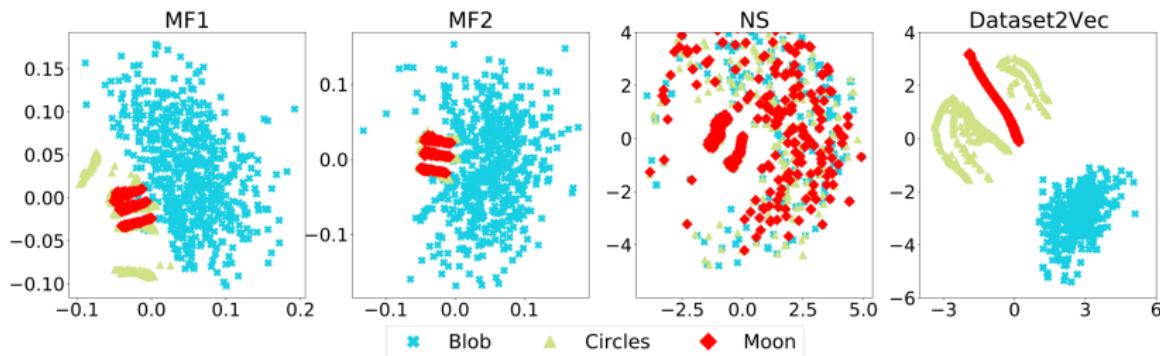
$$\text{meta}_X(S) = \frac{1}{|\mathbb{V}|} \sum_{V \in \mathbb{V}} \text{meta}_X(V)$$

$\text{meta}_X(V)$ izračunamo po enaki formuli kot za celotno množico.

Primerjava (ročnih) meta atributov in dataset2vec

Predstavljeno da se bodo te tri barve določile na diagramu,
ker gre za težo razliku množic.

Pomembno je dati redno zapomnimo kako do embeddingu prideamo



Priporočena literatura in programska oprema

- (Jomaa in ost. 2021): dataset2vec
- Paket *dataset2vec* na github.com/hadijomaa/dataset2vec

AutoML: optimalna konfiguracija algoritma \mathcal{A}

$$\max_{\theta \in \Theta} p(\mathcal{A}, \theta, S)$$

↓ *max vrednost po vseh možnih konfiguracijah*
 ↑ *nebomo kako naložiti parametre*
 ↗ *z podano*
 ↗ *algoritmu je podan*

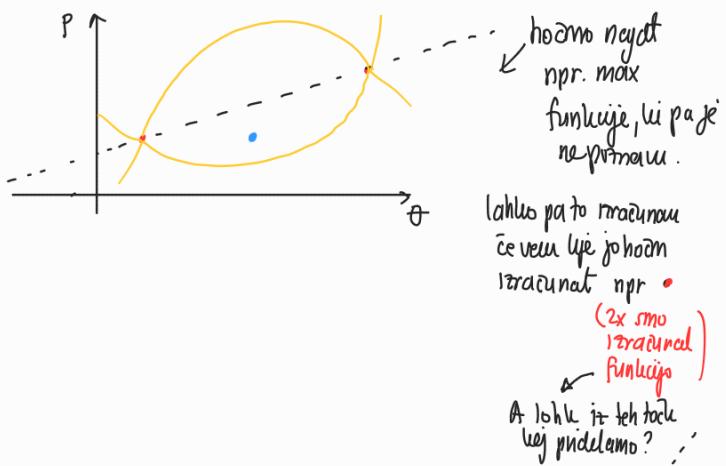
np.

\mathcal{A} = najboljši model
 $\Theta = \{N \dots \bar{N}\}$, točkovo
Zanjutka nas optimalni θ ?
 Rečemo $\theta = \{4, 2, 5, 10, 25\}$
 preizkusimo in vidimo ker θ vrže
 najboljši rezultat.

Posebna vrsta problema numerične optimizacije

- Časovno (pre)zahtevna ciljna funkcija
- Imamo na voljo omejeno število izračunov ciljne funkcije

SHBO algortem?

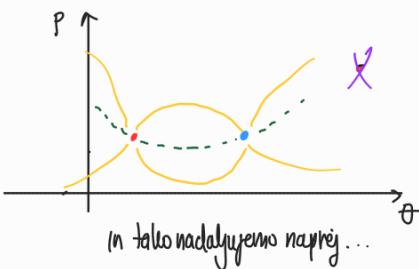


napovedane vrednosti p-ja
mislimo si da modul zna napovedati
confidence interval.

Kdor bi bila res dobra rezlva
za narednjo točko? Načelo ima
vsičke pogledale lyje je resrega
negotovost ali pa gledeči lyje
ki rezultati bili ta man.

- ta nova točka res
(točku racunal
torej je resrega
negotovost)

ta nov confidence interval



Osnovni algoritem SMBO: vhodi in izhodi

Sequential Model-Based Optimization: Zaporedna, na modelih temelječa optimizacija

Vhodi

- Algoritem strojnega učenja \mathcal{A}
- Podatkovna množica S
- Prostor vrednosti Θ nad-parametrov: konfiguracije algoritma \mathcal{A}
- Metoda za merjenje zmogljivosti $p : \mathbb{A} \times \Theta \times \mathbb{S} \rightarrow \mathbb{R}$
- Algoritem \mathcal{A}_s za učenje nadomestnega modela za funkcijo p

Izhod

Optimalna konfiguracija $\theta^* \in \Theta$ algoritma \mathcal{A} za množico S .

Osnovni algoritam in nadomestna funkcija

Require: \mathcal{A} in S sta algoritam in podatkovna množica

Require: Θ je prostor konfiguracij \mathcal{A}

Require: p je funkcija za vrednotenje zmogljivosti

Require: \mathcal{A}_s je algoritam za učenje nadomestkov

Ensure: θ^* je optimalna konfiguracija algoritma \mathcal{A}

```

1: function SMBO( $\mathcal{A}, S, \Theta, p, \mathcal{A}_s$ )
2:    $\theta, S_s = InitialSample(\Theta), \emptyset$  ← vzememo dve vrednosti (1. korak je odločit se k kateri nečetni konfiguraciji)
3:   repeat
4:      $S_s = S_s \cup \{(\theta, p(\mathcal{A}, \theta, S)) : \theta \in \Theta\}$  ← v mn. za učenje dodamo t/ točko iz prostora  $\Theta$ 
5:      $(\theta^*, p^*) = \arg \max_{(\theta, p) \in S_s} p$  in ta t/ točko moramo še proračunati p
6:      $m = \mathcal{A}_s(S_{surrogate})$  izberemo katera vrednost je maksimalna
7:      $\theta = Select(m, \Theta, \theta^*, S_s)$  ta korak je izbira redene točke (glej slise k jmo nihal tegor)
8:   until Terminate()
9:   return  $\theta^*$  funkcija ki te odloči kdo pushila naprej ali ne
10: end function

```

Ključne komponente SMBO

- ① *InitialSample*: začetni vzorec konfiguracij za izračun p , vrstica 2
- ② *Learn*: učenje nadomestnega modela z algoritmom \mathcal{A}_s , vrstica 6
- ③ *Select*: naslednji vzorec konfiguracij za izračun p , vrstica 7
- ④ *Terminate*: ustavitevni kriterij, vrstica 8

Komponenta *InitialSample*

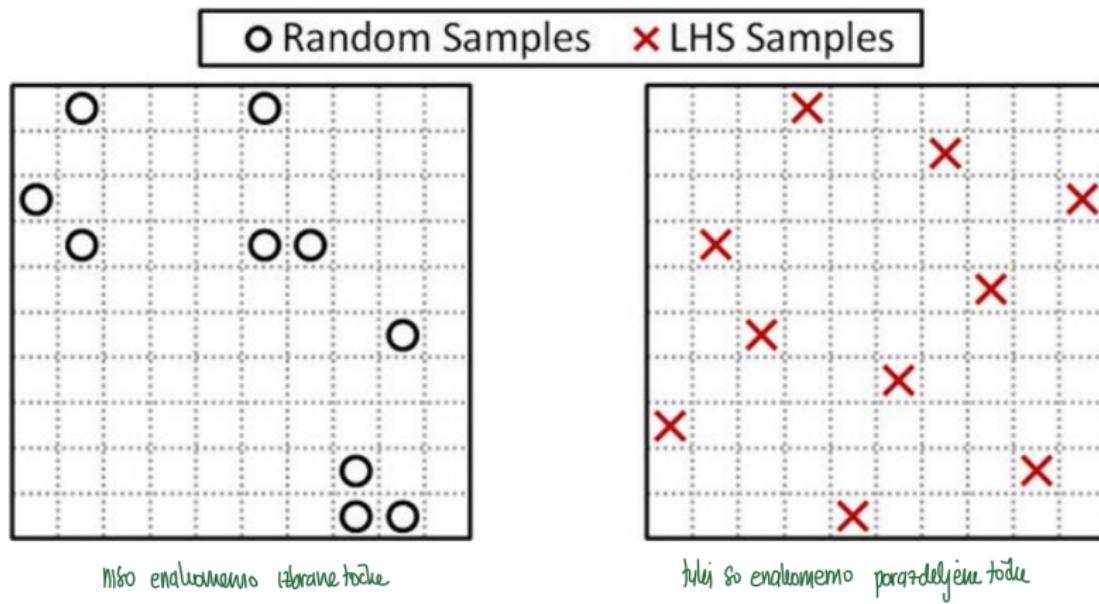
Poimenovana tudi začetni dizajn, *initial design*

Naključni vzorec θ konfiguracij $\theta \in \Theta$

- ① Naključno vzorčenje brez omejitev
- ② Vzorčenje po latinskem kvadratu, pravzaprav latinski hiper-kocki

Vzorčenje brez in z latinsko hiper-kocko

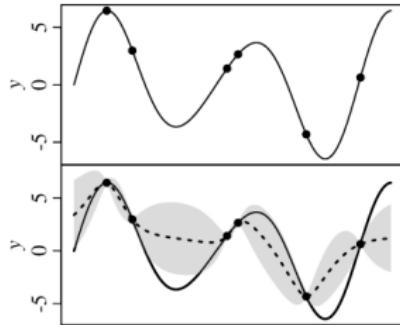
Prekletstvo dimenzionalnosti, prva predavanja pri osnovnem predmetu.



Komponenta *Learn*: nadomestni verjetnostni model m

Verjetnostni model za regresijo

- Napoveduje pričakovano vrednost ciljne spremenljivke μ_θ
- IN zanesljivost napovedi: interval zaupanja ali odklon σ_θ
- Najbolj pogosto uporabljen \mathcal{A}_s so Gaussovi procesi



Komponenta Select: Pričakovana izboljšava

Izboljšava, ki jo prinese konfiguracija θ

p_θ je napoved, ki jo dobimo pri danevi θ .

$$I(\theta) = \max(p^* - p_\theta, 0)$$

- p^* je trenutna optimalna zmogljivost
 - p_θ ne poznamo, lahko pa ocenimo z verjetnostnim modelom m
 - m vrne oceno pričakovane vrednosti napovedi μ_θ in odklon σ_θ
- p^{star} smo že
 do našega
 neločljivega
 izboljšanja od
 p^*
- ce najdeš
 $p_\theta < p^*$
 dejansko
 neždemo
 izboljšava, ker

Pričakovana izboljšava (*Expected Improvement*)

$$EI(\theta) = \mathbb{E}[W], \quad W = \max(p^* - Y, 0)$$

Naključna spremenljivka Y ima pričakovano vrednost μ_θ in varianco σ_θ .

$$E[W] = ?$$

Delamo shemo izpelyave (izraun pricakovane izbojivave)

$\bar{Y} \sim N(\mu, \sigma)$

$$W = \max(p^* - Y, 0) = \begin{cases} p^* - Y & ; Y \leq p^* \\ 0 & ; Y > p^* \end{cases}$$

$$E[W] = E[W | Y \leq p^*] \cdot P(Y \leq p^*) + E[W | Y > p^*] \cdot P(Y > p^*)$$

$$= E[p^* - Y | Y \leq p^*] F_Y(p^*) =$$

ker $Y \leq p^*$
 $W = p^* - Y$

$$= p^* F_Y(p^*) - E[Y | Y \leq p^*] \cdot F_Y(p^*)$$

p^* konstanta

= (*)

fajn bi bilo znat izraunat $F_{Y|Y \leq p^*}(y)$

$$= P(Y \leq y | Y \leq p^*) =$$

$$= \frac{P(Y \leq y \wedge Y \leq p^*)}{P(Y \leq p^*) - F_Y(p^*)} = \begin{cases} \frac{F_Y(y)}{F_Y(p^*)} ; y \leq p^* \\ 1 ; y > p^* \end{cases}$$

po def.
ker v prosledu ostanu
schony

še funkcija gostote

$$f_{Y|Y \leq p^*}(y) = \begin{cases} \frac{f_Y(y)}{F_Y(p^*)} ; y \leq p^* \\ 0 ; y > p^* \end{cases}$$

odvajaju

$$E[Y | Y \leq p^*] = \int_{-\infty}^{\infty} y \cdot f_{Y|Y \leq p^*}(y) dy = \frac{1}{F_Y(p^*)} \int_{-\infty}^{p^*} y \cdot f_Y(y) dy$$

ker ostalo je pol ite 0

$$\text{ker } Y \sim N(\mu, \sigma) \Rightarrow f_Y(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

$$\begin{aligned} 1. \text{ uvedemo sprem.} \\ u = y - \mu \\ 2. \text{ } r = -\frac{u^2}{2\sigma^2} \\ \vdots \end{aligned}$$

Sam naprej poracunamo
(tega ne bomo delal)

kaj dobimo

$$(*) = p^* F_Y(p^*) - \mu F_Y(p^*) + \sigma^2 f_Y(p^*) = (p^* - \mu) F_Y(p^*) + \underline{\sigma^2 f_Y(p^*)}$$

Komponenta *Select*: Izračun pričakovane izboljšave (1)

Pričakovana vrednost spremenljivke $W = \max(p^* - Y, 0)$

$$\begin{aligned}\mathbb{E}[W] &= \mathbb{E}[W|Y \leq p^*]P(Y \leq p^*) + \mathbb{E}[W|Y > p^*]P(Y > p^*) \\ &= \mathbb{E}[p^* - Y|Y \leq p^*]F_Y(p^*) + \mathbb{E}[0|Y > p^*]P(Y > p^*) \\ &= (p^* - \mathbb{E}[Y|Y \leq p^*])F_Y(p^*) \\ &= p^*F_Y(p^*) - \mathbb{E}[Y|Y \leq p^*]F_Y(p^*)\end{aligned}$$

Komponenta *Select*: Izračun pričakovane izboljšave (2)

Porazdelitvena funkcija F za spremenljivko $Y|Y \leq p^*$

$$\begin{aligned} F_{Y|Y \leq p^*}(y) &= P(Y \leq y | Y \leq p^*) \\ &= \frac{P(Y \leq y \wedge Y \leq p^*)}{P(Y \leq p^*)} \\ &= \begin{cases} \frac{F_Y(y)}{F_Y(p^*)} & \text{če } y \leq p^* \\ 1 & \text{če } y > p^* \end{cases} \end{aligned}$$

Funkcija gostote f za spremenljivko $Y|Y \leq p^*$ je odvod F

$$f_{Y|Y \leq p^*}(y) = \begin{cases} \frac{f_Y(y)}{F_Y(p^*)} & \text{če } y \leq p^* \\ 0 & \text{če } y > p^* \end{cases}$$

Komponenta Select: Izračun pričakovane izboljšave (3)

tega se ni hukl, ker se ta člen slavnava v

$$\mathbb{E}[Y|Y \leq p^*] = \frac{1}{F_Y(p^*)} \int_{-\infty}^{\infty} y f_{Y|Y \leq p^*}(y) dy$$

tukl je že

$$= \frac{1}{F_Y(p^*)} \int_{-\infty}^{p^*} y f_Y(y) dy$$

tukl je treba predpostavljati

$$Y \sim N(\mu, \sigma^2)$$

$$= \frac{1}{F_Y(p^*)} (\mu_F F_Y(p^*) - \sigma_F^2 f_Y(p^*))$$

Torej $\mathbb{E}[Y|Y \leq p^*] F_Y(p^*) = \mu_F F_Y(p^*) - \sigma_F^2 f_Y(p^*)$.

Komponenta *Select*: Izračun pričakovane izboljšave (4)

Če upoštevamo enakost iz prejšnje prosojnice, dobimo

$$\begin{aligned}\mathbb{E}[W] &= p^* F_Y(p^*) - \mathbb{E}[Y|Y \leq p^*] F_Y(p^*) \\ &= p^* F_Y(p^*) - \mu_\theta F_Y(p^*) - \sigma_\theta^2 f_Y(p^*) \\ &= (p^* - \mu_\theta) F_Y(p^*) - \sigma_\theta^2 f_Y(p^*)\end{aligned}$$

Komponenta *Select*: Izračun pričakovane izboljšave (5)

Predpostavimo, da je $Y \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$

to smo že pričakovali

$$EI(\theta) = (p^* - \mu_\theta) \Phi_Y(p^*) + \sigma_\theta^2 \phi_Y(p^*)$$

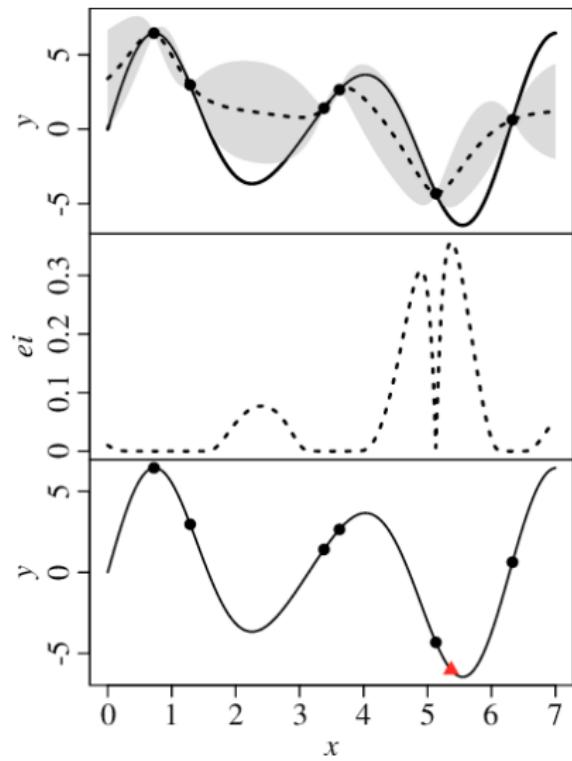
Φ_Y in ϕ_Y sta porazdelitvena funkcija in funkcija gostote za $\mathcal{N}(\mu_\theta, \sigma_\theta^2)$.

Kaj pa izračun Φ_Y in ϕ_Y ?

- Zamenjavo $u = (y - \mu_\theta)/\sigma_\theta$
- $\Phi_Y(y) = \Phi(u)$, kjer je Φ porazdelitvena funkcija za $\mathcal{N}(0, 1)$
- $\phi_Y(y) = \phi(u)/\sigma_\theta$, kjer je ϕ funkcija gostote za $\mathcal{N}(0, 1)$

Komponenta Select: Največja pričakovana izboljšava

max od $E[w]$
naniemo njo ker se hitro računa



Komponenta *Select*: Optimizacijski problem

$$\max_{\theta \in \Theta} EI(\theta)$$

- Ciljna funkcija EI nezahtevna za izračun, nadomestni model m
- Zato lahko uporabimo poljubno optimizacijsko metodo
- Pogosto: **metoda sistematičnega iskanja z rešetko, grid search**

Komponenta *Terminate*

- USTNI :
- Morno vedljaj je osnova poteka Bayesove optimizacije
 - mogoča tut del te izpeljave k svojo delu

Ustavitev kriterij

- Najbolj pogosto je to **omejitev uporabljenega procesorskega časa**
- Kar je ekvivalentno **omejitvi števila izračunov ciljne funkcije**

Smeri nadgradnje

- ① Stohastične ciljne funkcije: večkratni izračuni p_θ za podan θ
- ② Več podatkovnih množic namesto ene
- ③ Diskretni parametri

Stohastične ciljne funkcije: SKO

Sequential Kriging Optimization, Zaporedna optimizacija z regresijo zasnovano na Gaussovimi procesi

Uporabljene komponente

- *InitialSample*: večkratni izračuni p za izbrane začetne konfiguracije
- *Learn*: nadomestni model GP predpostavi šum v ciljni spremenljivki
- *Select*: optimizacija z Nelder-Mead metodo
- *Select*: pri izbiri se upošteva napoved plus en standardni odklon
- *Select*: pričakovana izboljšava prilagojena tako, da je pristranska do konfiguracij z visoko varianco

Stohastične ciljne funkcije: SPO

Sequential Parameter Optimization, Zaporedna optimizacija parametrov

ena stvar ker smo jo mi zanemarili, da smo rešili da je funk. p deterministična, ampak ni
(je stohastična)

ampak, pot bi mogel računati p za vsičko vedenat celično obrazoval kot stohastično

Uporabljeni komponente

- *InitialSample*: večkratni izračuni p za vse začetne konfiguracije
- \mathcal{A}_s : nadomestni model GP predpostavi *odsotnost šuma*
- *Select*: sistematični vzorec za iskanje največje pričakovane izboljšave, izbira vnaprej določenega števila konfiguracij za izračun ciljne funkcije
- *Select*: skrbi tudi za izbiro konfiguracij za ponovne/večkratne izračune

Več podatkovnih množic: SMAC in ROAR

sestavljen iz večjega št. dreves. Njegova naporedje poveže teh. dreves

SMAC: Sequential Model-based Algorithm Configuration

- *InitialSample*: naključno vzorčenje brez latinskih kvadratov
- *Learn*: naključni gozd kot verjetnosti model: napovedi posameznih dreves uporabljeni za oceno μ_θ in σ_θ
- *Select*: izbira podatkovnih množic za vrednotenje p ob upoštevanju števila prejšnjih izračunov

– modeljska optimizacija brez modela

ROAR: Random Online Aggressive Racing

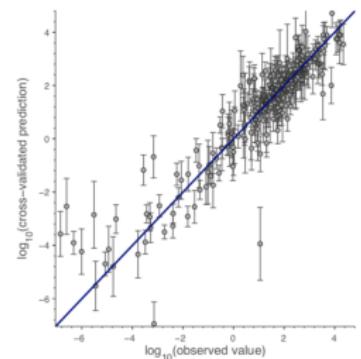
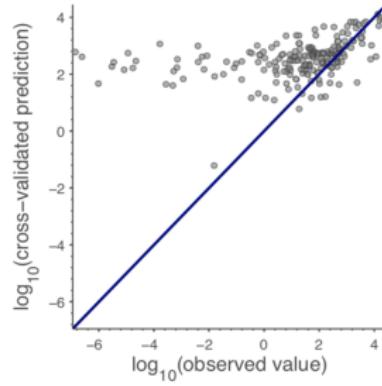
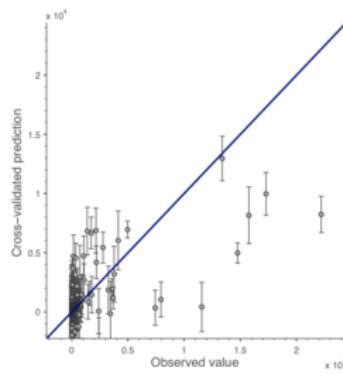
- Brez modela: naključna izbira konfiguracij za izračun ciljne funkcije
- *InitialSample*: Izbira le ene naključne konfiguracije $\theta \in \Theta$
- *Select*: Naključna izbira ene konfiguracije
- Presenetljivo dobri rezultati

Različni tipi parametrov: Naključni gozd

Modeli GP omejeni na numerične parametre

Learn: uporaba naključnega gozda odpravi to omejitev

Transformacija vrednosti ciljne funkcije (log)



Kaj pa meta učenje in meta modeli?

Uporaba meta modelov kot

- *InitialSample*: kriterij za izbiro začetnih konfiguracij
- *Select*: dodaten kriterij za izbiro nadaljnjih konfiguracij

V nadaljevanju meta model za nadomestne funkcije.

Literatura in praktični napotki

Priporočena literatura

- (Jones in ost. 1998): SMBO
- (Hutter in ost. 2009): SPO in SKO
- (Hutter in ost. 2011): SMAC in ROAR

Programska oprema in spletni viri

- Paket *hyperopt*, ki implementira orodja za SMBO
- Spletna stran *AutoML.org* in paket *auto-sklearn*