

# Teorijska pitanja

Strojno učenje 1, UNIZG FER, ak. god. 2023./2024.

Jan Šnajder, v1.3

## 2 Osnovni koncepti

1. (T) Pogreška modela definirana je kao očekivanje funkcije gubitka na primjerima iz  $\mathcal{X} \times \mathcal{Y}$ . Međutim, u praksi tu pogrešku aproksimiramo empirijskom pogreškom, koju računamo kao srednju vrijednost funkcije gubitka na skupu označenih primjera  $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$ . **Zašto pogrešku modela aproksimiramo empirijskom pogreškom i na kojoj se pretpostavci temelji ta aproksimacija?**

- ☐ A Različitih primjera iz  $\mathcal{X} \times \mathcal{Y}$  potencijalno ima beskonačno mnogo, pa pogrešku računamo na uzorku  $\mathcal{D}$  za koji pretpostavljamo da je reprezentativan
- ☐ B Ne možemo izračunati očekivanje gubitka jer nam nije poznata distribucija primjera iz  $\mathcal{X} \times \mathcal{Y}$ , no pretpostavljamo da je  $\mathcal{D}$  reprezentativan uzorak iz te distribucije
- ☐ C Očekivanje gubitka ne možemo izračunati jer primjera iz  $\mathcal{X} \times \mathcal{Y}$  ima potencijalno beskonačno, stoga pogrešku računamo na temelju skupa  $\mathcal{D}$  za koji pretpostavljamo da je konačan
- ☐ D Funkciju gubitka jednostavnije je definirati nego funkciju pogreške, a aproksimacija je točna uz pretpostavku i.i.d.

2. (T) Model  $\mathcal{H}$  je skup svih parametriziranih funkcija  $h(\mathbf{x}; \boldsymbol{\theta})$  indeksiran parametrima  $\boldsymbol{\theta}$ . To jest:

$$\mathcal{H} = \{h(\mathbf{x}; \boldsymbol{\theta})\}_{\boldsymbol{\theta}}$$

**Što to zapravo znači?**

- ☐ A Da model sadrži beskonačno mnogo funkcija  $h$  čija konkretna definicija ovisi o vrijednostima parametara  $\boldsymbol{\theta}$
- ☐ B Da različite funkcije  $h$  imaju različite parametre  $\boldsymbol{\theta}$ , i da su sve one sadržane u modelu, to jest za sve njih vrijedi  $h \in \mathcal{H}$
- ☐ C Da za različite parametre  $\boldsymbol{\theta}$  dobivamo različite funkcije  $h$ , i da su sve one sadržane u modelu, to jest za sve njih vrijedi  $h \in \mathcal{H}$
- ☐ D Da su funkcije  $h$  definirane sa slobodnim parametrima  $\boldsymbol{\theta}$  i da broj različitih funkcija odgovara broju parametara

3. (T) Modeli strojnog učenja tipično imaju i parametre i hiperparametre. **Koja je razlika između parametara i hiperparametara?**

- ☐ A Parametre optimira algoritam strojnog učenja, dok optimizacija hiperparametara nije u nadležnosti tog algoritma
- ☐ B Hiperparametri određuju jačinu regularizacije, a parametri stupanj nelinearnosti hipoteze
- ☐ C Parametri određuju iznos empirijske pogreške na skupu za učenje, a hiperparametri iznos te pogreške na skupu za provjeru
- ☐ D Hiperparametri mogu biti diskretni ili kontinuirani, dok su parametri uvijek kontinuirani

4. (T) U strojnom učenju, model je skup funkcija  $\mathcal{H}$  indeksiran parametrima  $\theta$ . Što to znači?
- ☐ A Svaki  $\theta$  jednoznačno određuje funkciju koja primjer  $\mathbf{x}$  preslikava u oznaku  $y$  u ovisnosti o parametrima  $\theta$
  - ☐ B Svaki skup funkcija  $\mathcal{H}$  ima svoj vektor parametara  $\theta$  i svaki vektor parametara  $\theta$  određuje skup funkcija  $\mathcal{H}$
  - ☐ C Svaki  $\mathbf{x}$  određuje parametar  $\theta$  kojim se oznaka  $y$  preslikava u primjer  $\mathbf{x}$
  - ☐ D Svaka funkcija koja primjeru  $\mathbf{x}$  dodjeljuje oznaku  $y$  jednoznačno određuje točku  $\theta$  u višedimenzijskome prostoru parametra
5. (T) Hipoteza  $h$  je funkcija koja primjerima iz  $\mathcal{X}$  pridjeljuje oznake iz  $\mathcal{Y}$ . Za  $h$  kažemo da je definirana “do na parametre  $\theta$ ”. Što to znači?
- ☐ A Funkcija  $h$  jednoznačno određuje parametre  $\theta$  iz skupa svih mogućih parametara, koji nazivamo prostor parametara
  - ☐ B Svaka vrijednost parametara  $\theta$  daje jednu konkretnu funkciju  $h$  koja se razlikuje od svih drugih funkcija u modelu  $\mathcal{H}$
  - ☐ C Funkcija  $h$  definirana je bez parametara, i njih treba odrediti naknadno postupkom odabira modela  $\mathcal{H}$
  - ☐ D Različite vrijednosti za  $\theta$  mogu dati različite funkcije  $h$ , a skup svih takvih različitih funkcija definira model  $\mathcal{H}$
6. (T) Svaki algoritam strojnog učenja ima neku induktivnu pristranost. Bez induktivne pristranosti, učenje na temelju podataka ne bi imalo smisla, odnosno algoritam bez induktivne pristranosti ne bi mogao ništa naučiti. **Zašto strojno učenje bez induktivne pristranosti nije moguće?**
- ☐ A Model bi bio prejednostavan te ne bi postojala hipoteza s empirijskom pogreškom nula
  - ☐ B Primjeri ne bi nužno bili linearno odvojjivi
  - ☐ C Oznaka niti jednog neviđenog primjera ne bi bila jednoznačno određena
  - ☐ D Prostor parametara bio bi neograničen, tj. postojalo bi beskonačno mnogo vektora parametara
7. (T) Modeli strojnog učenja općenito su različite složenosti. S porastom složenosti modela raste vjerojatnost da model bude prenaučan. Ta vjerojatnost raste s količinom šuma u podatcima. **Zašto šum u podatcima za učenje može dovesti do prenaučnosti klasifikacijskog modela?**
- ☐ A Zbog šuma granica između klasa izgleda nelinearnijom nego što ona to zapravo jest, pa primjeri blizu granice znatno više doprinose pogrešci učenja nego primjeri koji su udaljeni od granice
  - ☐ B Efekt šuma je slučajna, pa će hipoteza koja se previše prilagodi šumu na skupu za učenje očekivano imati veliku pogrešku na ispitnom skupu gdje je šum drugačiji ili ga nema
  - ☐ C Povećanjem količine šuma granica između klasa postaje sve nelinearnija, pa raste i složenost modela te dobivena hipoteza očekivano neće odgovarati granici između klasa na ispitnom skupu
  - ☐ D Zbog šuma su oznake nekih primjera u skupu za učenje pogrešne, pa sve hipoteze iz modela imaju na tom skupu pogrešku koja je veća od nula, a još veća na ispitnom skupu
8. (T) Svaki model strojnog učenja ima neku induktivnu pristranost. Što je induktivna pristranost?
- ☐ A Kriterij koji, na temelju modela, jednoznačno određuje hipotezu sa minimalnom empirijskom pogreškom
  - ☐ B Svaka pretpostavka koja jednoznačno određuje model na temelju hipoteze i skupa za učenje
  - ☐ C Odstupanje procjene parametra na temelju podataka u odnosu na pravu vrijednost parametra u populaciji
  - ☐ D Minimalan skup pretpostavki koje, uz skup za učenje, jednoznačno određuju klasifikaciju svakog primjera

9. (T) Model  $\mathcal{H}$  je skup hipoteza  $h(\cdot; \boldsymbol{\theta})$  koje su indeksirane vektorom parametara  $\boldsymbol{\theta}$ . Neka  $\boldsymbol{\theta} \in \mathbb{R}^{n+1}$ , gdje je  $n$  dimenzija ulaznog prostora. **Može li skup  $\mathcal{H}$  biti beskonačan?**
- ☐ A Da, primjerice ako  $\mathcal{X} = \mathbb{R}^n$  i  $h(\mathbf{x}; \boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{x}$
  - ☐ B Ne, jer je skup primjera  $\mathcal{D}$  uvijek konačan, neovisno o dimenzionalnosti ulaznog prostora  $n$
  - ☐ C Da, primjerice ako je  $\mathcal{X} = \{0, 1\}^n$  i  $h(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{1}\{\boldsymbol{\theta}^T \mathbf{x} \geq 0\}$
  - ☐ D Ne, jer za beskonačan skup  $\mathcal{H}$  optimizacijski problem  $\operatorname{argmax}_{h \in \mathcal{H}} E(h|\mathcal{D})$  nije definiran
10. (T) Za odabir modela optimalne složenosti tipično se koristi metoda unakrsne provjere. Neka  $\alpha$  i  $\boldsymbol{\theta}$  označavaju hiperparametre odnosno parametre nekog modela te neka je  $\mathcal{A}$  skup razmatranih vrijednosti hiperparametara. Neka su  $E(\boldsymbol{\theta}|\mathcal{D}_{\text{train}})$  i  $E(\boldsymbol{\theta}|\mathcal{D}_{\text{test}})$  pogreške hipoteze  $h(\mathbf{x}; \boldsymbol{\theta})$  na skupu za učenje odnosno ispitivanje. **Kako funkcionira metoda unakrsne provjere?**
- ☐ A Za svaki  $\alpha \in \mathcal{A}$  odabire se  $\boldsymbol{\theta}$  koji minimizira  $E(\boldsymbol{\theta}|\mathcal{D}_{\text{test}})$
  - ☐ B Za svaki  $\boldsymbol{\theta}$  provodi se minimizacija  $E(\boldsymbol{\theta}|\mathcal{D}_{\text{test}})$  po  $\alpha \in \mathcal{A}$
  - ☐ C Za svaki  $\boldsymbol{\theta}$  koji minimizira  $E(\boldsymbol{\theta}|\mathcal{D}_{\text{train}})$  odabire se  $\alpha \in \mathcal{A}$  koji minimizira  $E(\boldsymbol{\theta}|\mathcal{D}_{\text{test}})$
  - ☐ D Za svaki  $\alpha \in \mathcal{A}$  provodi se minimizacija  $E(\boldsymbol{\theta}|\mathcal{D}_{\text{train}})$  po  $\boldsymbol{\theta}$

### 3 Linearna regresija

1. (T) Svaki algoritam strojnog učenja ima neku induktivnu pristranost. Induktivna pristranost sastoji se od pristranosti jezika i pristranosti preferencije. **Kako glasi induktivna pristranost preferencije (neregulariziranog) modela linearne regresije?**
- ☐ A Hipoteza  $h$  je linearna kombinacija težina  $\mathbf{w}$  i značajki  $\mathbf{x}$
  - ☐ B Težine  $\mathbf{w}$  maksimiziraju iznos  $\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$
  - ☐ C Težine  $\mathbf{w}$  minimiziraju iznos  $\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$
  - ☐ D Hipoteza  $h$  je funkcija iz  $\mathbb{R}^n$  u  $\mathbb{R}$
2. (T) Rješenje najmanjih kvadrata za vektor težina  $\mathbf{w}$  jest:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

**Pod kojim uvjetima ćemo težine moći izračunati na ovaj način, i o čemu dominantno ovisi složenost tog postupka?**

- ☐ A Ako je rang matrice  $\mathbf{X}$  jednak  $N + 1$ , a složenost izračuna dominantno ovisi o  $N$
  - ☐ B Ako je rang matrice  $\mathbf{X}^T \mathbf{X}$  jednak  $N$ , a složenost izračuna dominantno ovisi o  $n$
  - ☐ C Ako je rang matrice  $\mathbf{X}$  jednak  $n + 1$ , a složenost izračuna dominantno ovisi o  $n$
  - ☐ D Ako je matrica  $\mathbf{X}^T \mathbf{X}$  kvadratna i punog ranga, a složenost izračuna dominantno ovisi o  $N$
3. (T) Svaki algoritam strojnog učenja ima neku induktivnu pristranost. Induktivna pristranost sastoji se od pristranosti jezika i pristranosti preferencije. **Koja je razlika između induktivnih**

**pristranosti regularizirane i neregularizirane linearne regresije?**

- ☐ A Algoritmi imaju različite pristranosti, i to različitu pristranost preferencije jer regularizirana regresija preferira jednostavnije hipoteze, a onda i različitu pristranost jezika jer je model neregularizirane regresije nadskup modela regularizirane regresije
- ☐ B Oba algoritma imaju isti model, definiran kao linearnu kombinaciju značajki i težina, pa dakle imaju istu pristranost jezika, ali se razlikuju u pristranosti preferencije jer imaju različito definiranu empirijsku pogrešku (osim ako je regularizacijski faktor jednak nuli)
- ☐ C Algoritmi se ne razlikuju po pristranosti preferencijom budući da koriste istu funkciju gubitka (kvadratni gubitak), međutim regularizirana regresija ima jaču induktivnu pristranost jezika od regularizirane regresije budući da prvi model uključuje drugi model
- ☐ D Za razliku od neregularizirane regresije, regularizirana regresija preferira jednostavnije hipoteze, međutim pristranosti su im identične jer su oba algoritma definirana kao linearna kombinacija značajki i težina te oba koriste identičan optimizacijski postupak (pseudoinverz matrice dizajna)
4. (T) Model linearne regresije je poopćeni linearni model i ima probabilističku interpretaciju. Prijetite se, tu smo interpretaciju upotrijebili smo kako bismo opravdali empirijsku funkciju pogreške definiranu na temelju kvadratnog gubitka. **Kako formalno glasi probabilistička pretpostavka modela linearne regresije?**
- ☐ A  $p(\mathbf{x}|y) = \mathcal{N}(f(\mathbf{x}), \sigma^2)$
- ☐ B  $p(y|\mathbf{x}) = \mathcal{N}(0, \sigma^2)$
- ☐ C  $p(y|\mathbf{x}) = \mathcal{N}(h(\mathbf{x}), \sigma^2)$
- ☐ D  $p(y) = \mathcal{N}(0, \sigma^2)$
5. (T) Svaki algoritam strojnog učenja ima neku induktivnu pristranost. Induktivna pristranost sastoji se od pristranosti jezika i pristranosti preferencije. **Što je induktivna pristranost preferencije linearnog modela regresije?**
- ☐ A Pretpostavka  $P(\mathbf{y}|\mathbf{w}) = \prod_{i=1}^N P(y^{(i)}|\mathbf{w})$
- ☐ B Minimizacija iznosa  $\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2$
- ☐ C Odabir linearnog modela  $h(\mathbf{w}; \mathbf{y}) = \mathbf{w}^T \mathbf{x}$
- ☐ D Maksimizacija iznosa  $-\ln \mathcal{L}(\mathbf{w}|\mathbf{y})$
6. (T) Optimizacija modela hrbatne regresije ( $L_2$ -regularizirane linearne regresije) ima rješenje u zatvorenoj formi. Neka je  $\lambda$  regularizacijski faktor,  $n$  broj značajki u ulaznom prostoru (bez “dummy” jedinice),  $m$  broj značajki u prostoru značajki (također bez “dummy” jedinice) te  $N$  broj primjera. Glavna komponenta rješenja je izračun inverza matrice izračunate na temelju matrice dizajna  $\Phi$ . **Koliko redaka odnosno stupaca ima matrica koju invertiramo?**
- ☐ A  $m + 1$    ☐ B  $m + \lambda$    ☐ C  $n + \lambda$    ☐ D  $N$
7. (T) Postupak najmanjih kvadrata (OLS) temelji se na izračunu pseudoinverza  $\mathbf{X}^+$  matrice dizajna  $\mathbf{X}$ , što je poopćenje običnog inverza  $\mathbf{X}^{-1}$ . **U kojoj situaciji je rješenje dobiveno pseudo-inverzom identično rješenju dobivenom običnim inverzom?**
- ☐ A Kada je broj primjera veći od broja značajki
- ☐ B Kada je broj primjera jednak broju značajki plus jedan i nema multikolinearnosti
- ☐ C Kada je broj značajki manji od broja primjera i nema multikolinearnosti
- ☐ D Kada nema multikolinearnosti i matrica dizajna je dobro kondicionirana

8. (T) Minimizacija funkcije kvadratne pogreške linearne regresije odgovara maksimizaciji log-izglednosti oznaka pod modelom. **Pod kojim uvjetom vrijedi ova korespondencija?**
- ☐ A Primjeri  $(\mathbf{x}, y)$  u skupu  $\mathcal{D}$  nezavisno su uzorkovani iz zajedničke distribucije  $P(\mathbf{x}, y)$
  - ☐ B Funkcija pogreške  $E(\mathbf{w}|\mathcal{D})$  je neprekidna i unimodalna
  - ☐ C Matrica dizajna  $\mathbf{X}$  nije singularna ili je regularizacijski faktor  $\lambda$  veći od 0
  - ☐ D Oznaka  $y$  primjera  $(\mathbf{x}, y)$  je normalna varijabla sa srednjom vrijednošću  $\mathbf{w}^T \mathbf{x}$
9. (T) Algoritam linearne regresije koristi kvadratni gubitak. **Koje je probabilističko opravdanje za korištenje kvadratnog gubitka?**
- ☐ A Vjerojatnost parametara za fiksni označeni skup podataka raste kvadratno s očekivanjem gubitka
  - ☐ B Primjeri za koje je predviđanje hipoteze pogrešno pokoravaju se normalnoj razdiobi sa središtem u nuli
  - ☐ C Vjerojatnost skupa primjera najveća je za parametre koji minimiziraju očekivanje kvadratnog gubitka
  - ☐ D Minimizacija kvadratnog gubitka istovjetna je maksimizaciji negativnog logaritma vjerojatnosti parametara

## 4 Linearna regresija II

1. (T) Rješenje najmanjih kvadrata s L2-regularizacijom (hrbatna regresija) je:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$$

gdje  $\lambda \mathbf{I} = \text{diag}(0, \lambda, \dots, \lambda)$ . **Koji je efekt regularizacije na Gramovu matricu?**

- ☐ A Dodavanje vrijednosti  $\lambda$  na dijagonale Gramove matrice povećava njezin rang
  - ☐ B Dodavanje vrijednosti  $\lambda$  na dijagonale Gramove matrice povećava normu težina  $\|\mathbf{w}\|$
  - ☐ C Minimizacija norme težina  $\|\mathbf{w}\|$  čini Gramovu matricu kvadratnom i singularnom
  - ☐ D Minimizacija norme težina  $\|\mathbf{w}\|$  povećava multikolinearnost Gramove matrice i smanjuje složenost modela
2. (T) Kao regularizacijski faktor kod modela linearne regresije tipično se koristi neka p-norma vektora težina,  $\|\mathbf{w}\|_p$ . **Na kojoj se činjenici temelji korištenje norme kao regularizacijskog izraza?**
- ☐ A Ako su težine hipoteze velike magnitude, model je prenaučan
  - ☐ B Ako je model prenaučan, hipoteza će imati velike magnitude težina
  - ☐ C Ako je model optimalne složenosti, hipoteza će imati male magnitude težina
  - ☐ D Ako su težine hipoteze male magnitude, model je podnaučan
3. (T)  $L_1$ -regularizacija ili LASSO kao regularizacijski izraz koristi prvu normu vektora težina,  $\|\mathbf{w}\|_1$ . **Što je prednost a što nedostatak  $L_1$ -regularizacije?**
- ☐ A Prednost je da  $L_1$ -regulariziranu pogrešku možemo minimizirati gradijentnim spustom, a nedostatak je da rezultira rijetkim modelima
  - ☐ B Prednost je da izbacuje značajke iz modela, a nedostatak je da  $L_1$ -regularizirana pogreška nema minimizator u zatvorenoj formi
  - ☐ C Prednost je da zadržava sve značajke u modelu, a nedostatak je da Gramova matrica može biti blizu singularne ako u podacima postoji multikolinearnost
  - ☐ D Prednost je da postoji rješenje u zatvorenoj formi (pseudoinverz), a nedostatak da izračun  $L_1$ -regulariziranog pseudoinverza ovisi o broju značajki ali i o broju primjera

4. (T) Optimizacijom regularizirane funkcije pogreške smanjuje se prenaučenos modela. **Kako je definirana  $L_2$ -regularizirana pogreška kod linearne regresije?**
- ☐ A Zbroj očekivanja funkcije gubitka unakrsne entropije i druge norme vektora težina
  - ☐ B Zbroj prosjeka kvadratnog gubitka na svim primjerima i kvadrata druge norme vektora težina bez težine  $w_0$
  - ☐ C Zbroj neregularizirane pogreške i izraza proporcionalnog s kvadratom norme vektora težina bez težine  $w_0$
  - ☐ D Zbroj funkcije gubitka po svim primjerima i neregularizirane pogreške bez težine  $w_0$
5. (T) Optimizacijom regularizirane funkcije pogreške smanjuje se prenaučenos modela. **Kakve parametre modela nalazi optimizacija  $L_2$ -regularizirane pogreške?**
- ☐ A Parametre koji uz što veću magnitudu vrijednosti daju što manje očekivanje gubitka na skupu za ispitivanje
  - ☐ B Parametre koji uz što manju magnitudu vrijednosti daju što manje očekivanje gubitka na skupu za učenje
  - ☐ C Parametre koji uz što veću magnitudu vrijednosti daju što veće očekivanje gubitka na skupu za učenje
  - ☐ D Parametre koji uz što manju magnitudu vrijednosti daju što veće očekivanje gubitka na skupu za ispitivanje
6. (T) Multikolinearnost značajki jedan je od problema koji može nastupiti kod primjene modela regresije na stvarnim podacima. Efekt multikolinearnosti i savršene multikolinearnosti dobro je uočljiv kod optimizacijskoga postupka običnih najmanjih kvadrata (OLS) kada se on provodi izračunom pseudoinverza matrice dizajna. Neka je  $m$  broj značajki,  $\Phi$  je matrica dizajna i  $\mathbf{G} = \Phi^T \Phi$  je Gramova matrica. **Koji je efekt savršene multikolinearnosti kod postupka OLS?**
- ☐ A  $\text{rang}(\Phi) < m + 1$ ,  $\mathbf{G}$  nema puni rang i nema inverz, no ima pseudoinverz koji nije numerički stabilan
  - ☐ B  $\Phi$  nema puni rang,  $\text{rang}(\mathbf{G}) < m + 1$  i  $\mathbf{G}$  nema pseudoinverz
  - ☐ C  $\Phi$  ima puni rang,  $\text{rang}(\mathbf{G}) > m$  i  $\mathbf{G}$  ima inverz, ali s visokim kondicijskim brojem
  - ☐ D  $\text{rang}(\Phi) = N$ , no  $\text{rang}(\mathbf{G}) < N$ , pa  $\mathbf{G}$  ima pseudoinverz, ali nema numerički stabilan inverz
7. (T) Premda je linearna regresija linearan model, ona može modelirati nelinearne hipoteze. Nelinearnost možemo ostvariti uporabom prikladne funkcije preslikavanja  $\phi$  koja primjere iz ulaznog prostora preslikava u prostor značajki. **Koja pretpostavka stoji iza takvog postupka?**
- ☐ A Primjeri koji u ulaznome prostoru leže na nelinearnoj hiperpovršini u prostoru značajki leže na hiperravnini
  - ☐ B Primjeri koji su linearno odvojivi u ulaznome prostoru postaju linearno neodvojivi u prostoru značajki dovoljno visoke dimenzije
  - ☐ C Hiperravnina koja minimizira pogrešku u ulaznome prostoru odgovara nelinearnoj hiperpovršini u prostoru značajki
  - ☐ D Primjeri koji se u prostoru značajki nalaze blizu nelinearne hiperpovršine u ulaznome prostoru leže na hiperravnini
8. (T) Pomoću regularizacije može se smanjiti prenaučenos modela. Jedna od tehnika regularizacije je  $L_2$ -regularizacija, koja se linearne regresije naziva "hrbatna regresija". **Koji je učinak  $L_2$ -regularizacije kod linearne regresije?**
- ☐ A Povećava normu vektora težina i tako privlači težine prema ishodištu prostora parametara
  - ☐ B Zakrivljuje područje oko točke minimuma funkcije pogreške u prostoru parametara
  - ☐ C Formira dugačak hrbat na površini funkcije pogreške u prostoru parametara
  - ☐ D Povećava kondicijski broj matrice dizajna pritezanjem težina na nulu

## 5 Linearni diskriminativni modeli

1. (T) Na predavanjima smo za klasifikaciju pokušali upotrijebiti algoritam regresije. Zaključili smo da to ne funkcionira, tj. da algoritam linearne regresije jednostavno nije klasifikacijski algoritam. **Koje bismo minimalne preinake trebale učiniti u algoritmu linearne regresije, a da dobijemo algoritam koji dobro funkcionira kao klasifikacijski algoritam?**
  - ☐ A Promijeniti funkciju gubitka
  - ☐ B Promijeniti model i funkciju gubitka
  - ☐ C Promijeniti funkciju gubitka i optimizacijski postupak
  - ☐ D Promijeniti model, funkciju gubitka i optimizacijski postupak
2. (T) Algoritam strojnog učenja idealno bi minimizirao gubitak 0-1. Međutim, funkciju gubitka 0-1 u praksi ne možemo koristiti za optimizaciju parametara modela. **Zašto gubitak 0-1 ne možemo koristiti za optimizaciju?**
  - ☐ A Gradijent gubitka 0-1 svugdje je nula osim za  $h(\mathbf{x}) = 0$ , pa funkcija pogreške ima zaravni po kojima se gradijentni spust ne može spuštati
  - ☐ B Gubitak 0-1 pored neispravno klasificiranih primjera kažnjava i ispravno klasificirane primjere, i to tim više što su oni dalje od granice između klasa
  - ☐ C Funkcija gubitka 0-1 nije diferencijabilna, pa ne postoji rješenje u zatvorenoj formi i ne postoji gradijent
  - ☐ D Funkcija gubitka 0-1 nije konveksna, pa ni funkcija pogreške nije konveksna već ima lokalne minimume te ne postoji minimizator u zatvorenoj formi
3. (T) Algoritam linearne regresije može se pokušati primijeniti na klasifikacijski problem, kao što smo pokušali na predavanjima, međutim to nije dobro funkcioniralo. Razmotrite tri komponente algoritma linearne regresije: model (M), funkciju gubitka (G) i optimizacijski postupak (O). Također, prisjetite se algoritma logističke regresije, koji je dobar klasifikacijski algoritam. Želimo preinačiti komponente algoritma linearne regresije tako da iz njega dobijemo nov algoritam koji klasifikaciju radi bolje od linearnog modela regresije, ali koji je drugačiji od logističke regresije. **Uz koje tri komponente bismo dobili takav algoritam?**
  - ☐ A M:  $h(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$ ; G:  $L(y, h(\mathbf{x})) = -y \ln h(\mathbf{x}) - (1 - y) \ln (1 - h(\mathbf{x}))$ , O:  $\mathbf{w}^* = \mathbf{X}^+ \mathbf{y}$
  - ☐ B M:  $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ ; G:  $L(y, h(\mathbf{x})) = -y \ln h(\mathbf{x}) - (1 - y) \ln (1 - h(\mathbf{x}))$  O:  $\mathbf{w}^*$  izračunat gradijentnim spustom
  - ☐ C M:  $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ ; G:  $L(y, h(\mathbf{x})) = (y - h(\mathbf{x}))^2$ , O:  $\mathbf{w}^* = \mathbf{X}^+ \mathbf{y}$
  - ☐ D M:  $h(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$ ; G:  $L(y, h(\mathbf{x})) = (y - h(\mathbf{x}))^2$ , O:  $\mathbf{w}^*$  izračunat gradijentnim spustom
4. (T) Višeklasni problem može se riješiti binarnim klasifikatorom uz primjenu sheme OVO ili sheme OVR. Obje sheme imaju svoje prednosti i nedostatke. Pretpostavite da raspolazemo sa  $K$  klasa i da svaka klasa ima  $N/K$  primjera, gdje je  $N$  ukupan broj primjera u skupu za učenje. **Što su prednosti odnosno nedostatci OVO i OVR sheme u takvom slučaju?**
  - ☐ A OVR treba  $K$  puta više klasifikatora nego OVO, ali su kod OVO pozitivne klase  $K/2$  puta manje zastupljene nego OVR
  - ☐ B OVO iziskuje  $(K - 1)/2$  puta više parametara nego OVR, ali svaki OVR klasifikator ima  $K - 1$  puta manje pozitivnih primjera nego negativnih
  - ☐ C OVR iziskuje  $K - 1$  puta više klasifikatora od sheme OVO, ali kod OVO pozitivne klase imaju  $K - 1$  puta manje primjera nego kod OVR
  - ☐ D OVO svaki klasifikator trenira s  $K/2$  puta manje primjera nego OVR, ali pozitivne klase kod OVR imaju  $K$  puta manje primjera nego kod OVO

5. (T) Jedna od triju komponenta svakog algoritma strojnog učenja je funkcija gubitka. Razmotrite funkcije gubitka perceptrona, logističke regresije (LR) i SVM-a. **Što je specifično funkciji gubitka perceptrona u odnosu na funkcije gubitka LR-a i SVM-a?**
- ☐ A Svaki primjer nanosi gubitak, ali manji za točno klasificirane primjere nego za netočno klasificirane primjere
  - ☐ B Gubitak za sve točno klasificirane primjere je nula, a za netočno klasificirane može biti manji od 1
  - ☐ C Točno klasificirani primjer nanosi gubitak manji od 1 te gubitak opada što je primjer bliže granici
  - ☐ D Gubitak netočno klasificiranih primjera raste linearno s udaljenošću od granice
6. (T) Funkcija gubitka perceptrona nalikuje funkciji gubitka SVM-a (funkciji zglobnice). Međutim, postoji ključna razlika između tih dviju funkcija gubitka, koje vode do različitog ponašanja algoritma perceptrona i algoritma SVM-a. **Po čemu se gubitak zglobnice razlikuje od gubitka perceptrona?**
- ☐ A Za ispravno klasificirane primjere gubitak zglobnice je manji od gubitka za neispravno klasificirane primjere
  - ☐ B Gubitak zglobnice je nula za primjere koji su ispravno klasificirani i daleko od granice
  - ☐ C Za neispravno klasificirane primjere gubitak zglobnice raste linearno s udaljenošću od hiper-ravnine
  - ☐ D Gubitak zglobnice kažnjava sve primjere koji se nalaze unutar margine, čak i one koji su ispravno klasificirani
7. (T) Jedan od nedostataka algoritma perceptrona jest taj što se za linearno neodvojive probleme algoritam ne zaustavlja. **Zašto je tomu tako?**
- ☐ A Sve točke minimuma funkcije pogreške u prostoru parametara leže na istoj ravni
  - ☐ B Površina funkcije pogreške u prostoru parametara po dijelovima je linearna i stoga nederivabilna
  - ☐ C Svaki lokalni minimum funkcije pogreške u prostoru parametara veći je od nule
  - ☐ D Gradijent funkcije gubitka uvijek je različit od nul-vektora za barem jedan označeni primjer

## 6 Logistička regresija

1. (T) Poopćeni linearni model definirali smo kao  $h(\mathbf{x}) = f(\mathbf{w}^T \phi(\mathbf{x}))$ , gdje je  $f$  neka (moguće nelinearna) aktivacijska funkcija, a  $\phi$  je (moguće nelinearna) funkcija preslikavanja u prostor značajki. **Koji od navedenih uvjeta je dovoljan uvjet da granica između klasa u ulaznom prostoru bude linearna?**
- ☐ A  $f$  je afina funkcija    ☐ B  $\phi(\mathbf{x}) = (1, \mathbf{x})$     ☐ C  $f$  je afina funkcija i  $\phi(\mathbf{x}) = (1, \mathbf{x})$     ☐ D  $f(\mathbf{x}) = \mathbf{x}$
2. (T) Kod logističke regresije, pogrešku unakrsne entropije izveli smo modelirajući distribuciju vjerojatnosti oznaka  $y$  u skupu označenih primjera. **Na koji smo način modelirali distribuciju vjerojatnosti pojedinačnog primjera  $y$ ?**
- ☐ A  $P(y|\mathbf{x}) = (y - h(\mathbf{x}))\mathbf{x}$
  - ☐ B  $P(y|\mathbf{x}) = h(\mathbf{x})^y(1 - h(\mathbf{x}))^{1-y}$
  - ☐ C  $P(y|\mathbf{x}) = y^{h(\mathbf{x})}(1 - y)^{h(\mathbf{x})}$
  - ☐ D  $P(y|\mathbf{x}) = h(\mathbf{x})(1 - h(\mathbf{x}))$



3. (T) Optimizacija parametara logističke regresije algoritmom grupnog gradijentnog spusta tipično u svakoj iteraciji uključuje i linijsko pretraživanje. **Što nam osigurava uporaba linijskog pretraživanja kod optimizacije logističke regresije?**
- ☐ A Da postupak uvijek konvergira, neovisno o odabranoj stopi učenja  $\eta$  i početnim težinama  $\mathbf{w}$
  - ☐ B Da postupak uvijek konvergira, pod uvjetom da su primjeri linearno neodvojivi ili da regulariziramo sa  $\lambda > 0$
  - ☐ C Da postupak ne može zaglaviti u lokalnome minimumu, pod uvjetom da u skupu  $\mathcal{D}$  nema multikolinearnosti
  - ☐ D Da postupak konvergira brže, pod uvjetom da su primjeri linearno odvojivi i da stopa učenja nije  $\eta$  prevelika
4. (T) Neregularizirani model logističke regresije sklon je prenaučivosti. To je pogotovo slučaj ako se model trenira na linearno odvojivim podacima, čak i onda kada u podacima nema nikakvog šuma i kada se ne koristi nikakvo preslikavanje u prostor značajki. **Zbog čega dolazi do prenaučivosti modela neregularizirane logističke regresije na linearno odvojivim skupovima podataka?**
- ☐ A Empirijska pogreška logističke regresije smanjuje se s porastom broja primjera
  - ☐ B Ispravno klasificirani primjeri koji su vrlo udaljeni od granice nanose malen gubitak
  - ☐ C S porastom norme vektora težina gubitak na točnim primjerima teži prema nuli
  - ☐ D Netočno klasificirani primjeri nanose gubitak koji je proporcionalan normi vektora težina
5. (T) Algoritam logističke regresije za optimizaciju može koristiti grupni gradijentni spust s linijskim pretraživanjem. Takav optimizacijski algoritam ima svojstvo globalne konvergencije. Razmotrite neregulariziranu logističku regresiju na linearno neodvojivom problemu. **Što globalna konvergencija konkretno znači u tom slučaju?**
- ☐ A Optimizacijski algoritam će konvergirati prema parametrima koji minimiziraju pogrešku na skupu za učenje, ali neće doseći minimum
  - ☐ B Gradijentni spust neće krivudati u prostoru parametara jer optimizacijski algoritam koristi informaciju o zakrivljenosti površine pogreške
  - ☐ C Optimizacijski algoritam će konvergirati do minimuma, ali nema garancije da će to biti globalni minimum funkcije pogreške
  - ☐ D Neovisno o inicijalizaciji, optimizacijski će algoritam pronaći parametre koji minimiziraju pogrešku na skupu za učenje

## 7 Logistička regresija II

1. (T) Kod logističke regresije za optimizaciju tipično koristimo gradijentni spust ili Newtonov optimizacijski postupak. **Što su prednosti, a što nedostaci gradijentnog spusta u odnosu na Newtonov postupak, i to konkretno kod logističke regresije?**
- ☐ A Za razliku od Newtonovog postupka, gradijentni spust može se koristiti za L2-regulariziranu logističku regresiju, no ako je stopa učenja prevelika, postupak može divergirati, dok Newtonov postupak nema taj problem
  - ☐ B Za razliku od Newtonovog postupka, gradijentni spust može se koristiti za "online" (pojedinačno) učenje, no može krivudati i zato sporije konvergirati od Newtonovog postupka
  - ☐ C Newtonov postupak brže konvergira, ali se može koristiti samo za konveksnu funkciju pogreške, dok gradijentni spust nema tog ograničenja, ali može zaglaviti u lokalnom optimumu
  - ☐ D Gradijentni spust znatno je računalno jednostavniji od Newtonovog postupka, no za razliku od Newtonovog postupka kod L2-regularizirane regresije ne konvergira ako primjeri nisu linearno odvojivi

2. (T) Kod Newtonovog postupka optimizacije za logističku regresiju ažuriranje težina provodi se prema sljedećem pravilu:

$$\mathbf{w} \leftarrow \mathbf{w} - \mathbf{H}^{-1} \nabla E(\mathbf{w} | \mathcal{D})$$

Očito, za provođenje ovog postupka potrebno je računati inverz Hesseove matrice, tj. matrice parcijalnih derivacija  $\mathbf{H}$ . Općenito, operacija invertiranja matrice nije uvijek izvediva, a čak i kada jest izvediva, rješenje nije uvijek numerički stabilno. **Kod logističke regresije, koji je nužan i dovoljan uvjet za izvedivost i numeričku stabilnost Newtonovog optimizacijskog postupka?**

- ☐ A Značajke moraju biti linearno zavisne
  - ☐ B Funkcija pogreške mora biti konveksna
  - ☐ C U podacima ne smije biti multikolinearnosti
  - ☐ D Broj primjera mora biti veći od broja značajki plus jedan
3. (T) Svi poopćeni linearni modeli mogu se trenirati u “online” (pojedinačnom) načinu, primjermom algoritma LMS. To vrijedi i za algoritam linearne regresije, za koji smo prvotno kao minimizaciju kvadrata provodili računajući pseudoinverz matrice dizajna. Jedna od prednosti algoritma LMS u odnosu na izračun pseudoinverza kod linearne regresije je manja računalna složenost LMS-a. Neka  $E$  označava broj epoha,  $N$  je broj primjera, a  $m$  broj značajki u prostoru značajki. **Koja je (vremenska) računalna složenost algoritma LMS, primijenjenog na linearnu regresiju?**
- ☐ A  $\mathcal{O}(ENm)$
  - ☐ B  $\mathcal{O}(E(N + m))$
  - ☐ C  $\mathcal{O}(EN^2m)$
  - ☐ D  $\mathcal{O}(ENm^2)$
4. (T) Problem višeklasne ( $K > 2$ ) klasifikacije logističkom regresijom možemo riješiti na više načina. Možemo primijeniti (1) multinomijalnu logističku regresiju (MLR), (2) binarnu logističku regresiju sa shemom OVO (BLR-OVO) ili (3) binarnu logističku regresiju sa shemom OVR (BLR-OVR). **Koja je prednost MLR nad BLR-OVO i BLR-OVR?**
- ☐ A MLR ima više parametara od BLR-OVR, ali nije osjetljiva na neuravnoteženost broja primjera po klasama
  - ☐ B MLR i BLR-OVR imaju manje parametara od BLR-OVO, no jedino za MLR vrijedi  $\sum_k P(y = k | \mathbf{x}) = 1$
  - ☐ C Za razliku od BLR-OVR i BLR-OVO, kod MLR ne postoje područja u ulaznom prostoru za koje klasifikacijska odluka nije određena
  - ☐ D Za razliku od BLR-OVO i BLR-OVR, MLR koristi funkciju softmax, pa minimizacija L1-regularizirane pogreške ima rješenje u zatvorenoj formi
5. (T) Kod logističke regresije optimizaciju tipično provodimo gradijentnim spustom. Međutim, kod linearne regresije optimizaciju smo provodili izračunom pseudoinverza matrice dizajna. **Zašto optimizaciju kod logističke regresije također ne provodimo izračunom pseudoinverza matrice dizajna?**
- ☐ A Optimizaciju parametara linearne regresije također možemo napraviti gradijentnim spustom po empirijskoj pogrešci, ali to ne radimo jer izračun pseudoinverza ima manju računalnu složenost
  - ☐ B Zbog nelinearnosti logističke funkcije, kod logističke regresije izračun pseudoinverza matrice dizajna nije moguće napraviti u zatvorenoj formi
  - ☐ C Maksimizacija log-izglednosti oznaka logističke regresije kao rješenje za parametre ne daje izraz u zatvorenoj formi koji sadržava pseudoinverz matrice dizajna
  - ☐ D Optimizaciju možemo provesti izračunom pseudoinverza matrice dizajna, međutim, za razliku od gradijentnog spusta, taj postupak ne funkcionira kada je matrica dizajna singularna
6. (T) Poopćeni linearni modeli (linearna regresija, logistička regresija i multinomijalna regresija) probabilistički su algoritmi strojnog učenja. Njihova probabilistička priroda dolazi do izražaja kako

kod modela tako i kod optimizacijskog postupka. **Koji je probabilistički princip ugrađen u optimizacijski postupak tih algoritama?**

- ☐ A Minimizirati  $\sum_{i=1}^N \ln h(\mathbf{x}^{(i)}; \mathbf{w})$ , gdje je  $h(\mathbf{x}; \mathbf{w}) = f(\mathbf{w}^T \mathbf{x})$
- ☐ B Minimizirati  $-\sum_{i=1}^N \ln y^{(i)} h(\mathbf{x}^{(i)}; \mathbf{w})$ , gdje je  $h(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \mathbf{x}$
- ☐ C Maksimizirati  $\sum_{i=1}^N \ln p(y^{(i)}|\mathbf{x}^{(i)})$ , gdje je  $\mathbb{E}[p(y^{(i)}|\mathbf{x}^{(i)})] = f(\mathbf{w}^T \mathbf{x})$
- ☐ D Maksimizirati  $\prod_{i=1}^N \ln p(y^{(i)}|\mathbf{x}^{(i)})$ , gdje je  $\mathbb{E}[p(y^{(i)}|\mathbf{x}^{(i)})] = \mathbf{w}^T \mathbf{x}$

7. (T) Postoji poveznica između algoritma logističke regresije (LR) i algoritma neuronske mreže sa sigmoidnim prijenosnim funkcijama (NN). **Koja je točno poveznica između ova dva algoritma?**

- ☐ A NN i LR imaju istu funkciju pogreške, ali se samo LR može optimirati Newtonovim postupkom jer funkcija gubitka NN nije konveksna
- ☐ B Jezgreni stroj s Gaussovom jezgrenom funkcijom istovjetan je NN sa  $L_2$ -regulariziranom funkcijom pogreške
- ☐ C Model LR istovjetan je modelu dvoslojne NN sa sigmoidnim prijenosnim funkcijama i pogreškom unakrsne entropije
- ☐ D Model dvoslojne NN istovjetan je modelu LR s poopcenim linearnim modelima sa sigmoidnim funkcijama kao baznim funkcijama

8. (T) Za optimizaciju parametara poopcenih linearnih modela može se koristiti stohastički gradijentni spust, odnosno pravilo LMS. Neka je  $(\mathbf{x}, y)$  označeni primjer za koji radimo ažuriranje težina pomoću pravila LMS. **Što možemo reći o razlici između novih (ažuriranih) i starih težina (težina prije ažuriranja)?**

- ☐ A Razlika je to veća što je stopa učenja  $\eta$  bliža jedinici
- ☐ B Razlika je to manja što je vektor  $\phi(\mathbf{x})$  bliži ishodištu
- ☐ C Razlika je to veća što je izlaz modela  $h(\mathbf{x})$  bliži nuli
- ☐ D Razlika je to manja što je oznaka  $y$  bliže jedinici

9. (T) Postoji veza između logističke regresije i modela neuronske mreže. **Koja je veza između ta dva modela?**

- ☐ A Multinomijalna logistička regresija s aktivacijskom funkcijom softmax istovjetna je dvoslojnoj neuronskoj mreži sa više neurona u izlaznom sloju
- ☐ B Logistička regresija koja kao adaptivne bazne funkcije koristi logističku regresiju istovjetna je neuronskoj mreži sa sigmoidnom aktivacijskom funkcijom
- ☐ C Neuronska mreža optimirana algoritmom propagacije pogreške unazad istovjetna je logističkoj regresiji optimiranoj stohastičkim gradijentnim spustom
- ☐ D Logistička regresija s linearnim jezgrenim funkcijama istovjetna je neuronskoj mreži sa linearnom aktivacijskom funkcijom i kvadratnom funkcijom pogreške

10. (T) Multinomijalna logistička regresija primjere klasificira u jednu od  $K$  klasa. Neka je  $\mathbf{y}^{(i)} = (y_1^i, \dots, y_K^i)$  vektor indikatorskih varijabli koji odgovara oznaci primjera  $\mathbf{x}^{(i)}$ . Neka je  $h_k(\mathbf{x}; \mathbf{W})$  vrijednost funkcije *softmax* za klasu  $k$ . Optimizacija parametara  $\mathbf{W}$  odgovara maksimizaciji logaritma vjerojatnost oznaka. **Čemu je jednaka ta vjerojatnost?**

- ☐ A  $\sum_{i=1}^N \sum_{k=1}^K \ln y_k^i h_k(\mathbf{x}^{(i)}; \mathbf{W})$
- ☐ B  $\prod_{i=1}^N \sum_{k=1}^K \ln h_k(\mathbf{x}^{(i)}; \mathbf{W}) y_k^i$
- ☐ C  $\sum_{i=1}^N \ln \prod_{k=1}^K h_k(\mathbf{x}^{(i)}; \mathbf{W}) y_k^i$
- ☐ D  $\ln \prod_{i=1}^N \prod_{k=1}^K y_k^i h_k(\mathbf{x}^{(i)}; \mathbf{W})$

## 8 Stroj potpornih vektora

1. (T) Kod SVM-a, problem maksimalne margine sveo se na problem minimizacije izraza  $\frac{1}{2}\|\mathbf{w}\|^2$  uz određena ograničenja. **Zašto minimizacija ovog izraza daje maksimalnu marginu?**
  - A Što je vektor  $\mathbf{w}$  kraći, to je manja vrijednost  $h(\mathbf{x})$ , pa primjeri moraju biti što dalje da bi vrijedilo  $h(\mathbf{x}) = \pm 1$ , a to znači da je margina to šira
  - B Što je vektor  $\mathbf{w}$  kraći, to je manja udaljenost  $d$  primjera od hiperravnine, a to efektivno znači da je margina to šira jer je margina fiksna a udaljenosti  $d$  se smanjuju
  - C Što je vektor  $\mathbf{w}$  kraći, to je manja vrijednost  $h(\mathbf{x})$ , ali je težina  $w_0$  konstantna, pa se udaljenosti između hiperravnine i primjera povećavaju, što znači da se margina širi
  - D Što je vektor  $\mathbf{w}$  kraći, to je veća udaljenost  $d$  primjera od hiperravnine, što znači da se potporni vektori udaljavaju od hiperravnine, a to znači da margina postaje šira
2. (T) Kod optimizacije SVM-a iskoristili smo Lagrangeovu dualnost kako bismo se iz primarnog optimizacijskog problema prebacili u dualni optimizacijski problem. To smo učinili tako da smo na temelju Lagrangeove funkcije  $L$  definirali dualnu Lagrangeovu funkciju  $\tilde{L}$  i uveli nova ograničenja, što nam je opet dalo kvadratni program. **Kako onda u konačnici glasi optimizacijski problem tvrde margine u dualnoj formulaciji (ako zanemarimo ograničenja)?**
  - A  $\operatorname{argmin}_{\alpha} \max_{\mathbf{w}, w_0} L(\mathbf{w}, w_0, \alpha)$
  - B  $\operatorname{argmax}_{\mathbf{w}, w_0} \min_{\alpha} L(\mathbf{w}, w_0, \alpha)$
  - C  $\operatorname{argmin}_{\mathbf{w}, w_0} \max_{\alpha} L(\mathbf{w}, w_0, \alpha)$
  - D  $\operatorname{argmax}_{\alpha} \min_{\mathbf{w}, w_0} L(\mathbf{w}, w_0, \alpha)$
3. (T) Svaki algoritam strojnog učenja ima neku induktivnu pristranost. Bez induktivne pristranosti nije moguće naučiti model koji bi generalizirao. **Po čemu se induktivna pristranost algoritma SVM (tvrda margina) razlikuje od induktivne pristranosti algoritma perceptrona?**
  - A SVM ima pristranost preferencijom kojom maksimizira marginu, dok perceptron nema induktivnu pristranost preferencijom već samo pristranost jezika
  - B Imaju istu pristranost preferencijom, a to je da primjeri moraju biti linearno odvojivi, no SVM ima dodatnu pristranost ograničenjem u vidu optimizacijskih ograničenja
  - C Razlikuju se po pristranost preferencijom, jer perceptron ne maksimizira marginu, premda se može dogoditi da pronađe rješenje koje maksimizira marginu
  - D Imaju istu pristranost jezika, a pristranost preferencijom također će biti ista ako se oba optimiraju gradijentnim spustom s istim početnim težinama i istom stopom učenja
4. (T) Kod izvoda algoritma SVM s tvrdom marginom, pretpostavili smo da za primjere  $\mathbf{x} \in \mathbb{R}^n$  vrijedi sljedeći uvjet linearne odvojivosti:

$$\forall (\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}. y^{(i)} h(\mathbf{x}^{(i)}) \geq 0$$

**Koliko hipoteza zadovoljava ovaj uvjet, i kako algoritam SVM odabire jednu od njih?**

- A Uvjet zadovoljava beskonačno mnogo hipoteza, međutim samo za jednu vrijedi  $yh(\mathbf{x}) = 1$  za najbliže primjere, i to je hipoteza koju odabire SVM
- B Uvjet zadovoljava konačan broj hipoteza koje su linearno odvojive, a SVM između njih odabire onu jednu koja minimizira kvadrat vektora težina
- C Uvjet zadovoljava beskonačno mnogo hipoteza, a SVM odabire onu jednu koja minimizira kvadrat vektora težina te koja ispravno klasificira sve primjere, uz uvjet da  $h(\mathbf{x})$  nije u intervalu  $(-1, +1)$
- D Uvjet zadovoljava konačan broj hipoteza koje su linearno odvojive, no one se razlikuju samo po faktoru koji množi težine  $(\mathbf{w}, w_0)$ , pa SVM odabire onu jednu za koju vrijedi  $yh(\mathbf{x}) \geq 1$  za sve primjere

5. (T) Model SVM-a može se definirati i optimirati u primarnoj ili dualnoj formulaciji. **Konceptualno, kada će primjer  $\mathbf{x}$  u dualnoj formulaciji SVM-a biti klasificiran u pozitivnu klasu?**
- ☐ A Ako je linearna kombinacija značajki iz  $\mathbf{x}$  s pozitivnim težinama veća ili jednaka linearnoj kombinaciji značajki iz  $\mathbf{x}$  s negativnim težinama
  - ☐ B Ako je vektor  $\mathbf{x}$  po skalarnom umnošku sličniji potpornim vektorima s pozitivnom oznakom nego potpornim vektorima s negativnom oznakom
  - ☐ C Ako je skalarni umnožak vektora  $\mathbf{x}$  i vektora oznaka  $\mathbf{y}$  veća od praga definiranog parametrom  $w_0$
  - ☐ D Ako većina od ukupno  $\alpha$  primjera iz skupa za učenje koji su po euklidskoj udaljenosti najbliži primjeru  $\mathbf{x}$  ima pozitivnu oznaku
6. (T) Problem maksimalne margine ima svoju geometrijsku interpretaciju: maksimalna margina odgovara simetrali spojnice konveksnih ljusaka primjera iz dviju klasa. **Što je nužan i dovoljan uvjet da klasifikacijski problem bude rješiv SVM-om s tvrdom marginom?**
- ☐ A Primjeri iz obje klase trebaju činiti konveksne skupove u ulaznom prostoru
  - ☐ B Najbliži primjeri iz suprotnih klasa moraju biti u vrhovima konveksnih ljusaka
  - ☐ C Barem jedna spojica između primjera jedne klase treba biti unutar konveksne ljuske druge klase
  - ☐ D Konveksne ljuske dviju klasa ne smiju se preklapati (trebaju biti disjunktne)

## 9 Stroj potpornih vektora II

1. (T) Problem meke margine SVM-a formulirali smo kao problema optimizacije uz ograničenja, preciznije kao problem kvadratnog programiranja. Neka je  $n$  broj značajki, a  $N$  broj primjera. **Koliko primarni optimizacijski problem meke margine ima ukupno ograničenja, a koliko varijabli po kojima optimiramo?**
- ☐ A  $2N$  ograničenja i  $2n$  varijabli
  - ☐ B  $N$  ograničenja i  $2N + 1$  varijabli
  - ☐ C  $N$  ograničenja i  $n + 1$  varijabli
  - ☐ D  $2N$  ograničenja i  $N + n + 1$  varijabli
2. (T) Kod optimizacijskog problema meke margine jedan od uvjeta KKT koji vrijede u točki rješenja je sljedeći uvjet komplementarne labavosti:

$$\alpha_i (y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + w_0) - 1 + \xi_i) = 0$$

**Što možemo zaključiti na temelju ovog uvjeta?**

- ☐ A Da se primjeri koji nisu potporni vektori sigurno nalaze izvan margine
  - ☐ B Da se potporni vektori nalaze na margini ili izvan nje, a na pravoj strani granice
  - ☐ C Da se primjeri koji nisu potporni vektori nalaze na margini ili unutar margine
  - ☐ D Da se potporni vektori ne nalaze izvan margine na pravoj strani granice
3. (T) Problem meke margine SVM-a s u primarnoj se formulaciji svodi na rješavanje sljedećeg optimizacijskog problema:

$$\operatorname{argmin}_{\mathbf{w}, w_0, \xi} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \right\}$$

uz određena linearna ograničenja. Ovaj optimizacijski problem odgovara optimizaciji regularizirane pogreške. Kod regularizirane pogreške u opreci su dva cilja: smanjenje vrijednosti funkcije gubitka i

smanjenje složenosti modela. **Kako se ta opreka manifestira kod optimizacijskog problema meke margine SVM-a?**

- ☐ A Što je veća vrijednost  $\|\mathbf{w}\|^2$ , to je margina uža i tim manje primjera ulazi u marginu, pa je tim manji zbroj od  $\xi_i$
  - ☐ B Što je veća vrijednost  $\|\mathbf{w}\|^4$  to je model složeniji, no tim je veća nelinearnost granice i to je veći hiperparametar  $C$
  - ☐ C Što je manji zbroj od  $\xi_i$ , to više primjera može ući u marginu i tim je veća vrijednost  $\|\mathbf{w}\|^2$  te je model manje složenosti
  - ☐ D Što je veći zbroj od  $\xi_i$ , to više primjera ulazi u marginu i tim je manja vrijednost  $\|\mathbf{w}\|^2$  te je model veće složenosti
4. (T) Optimizacijski problem algoritma SVM može se postaviti u formulaciji meke ili tvrde margine te u primarnoj ili dualnoj formulaciji. Ovisno o formulaciji, kvadratni program sadrži različit broj varijabli po kojima optimiramo (optimizacijske varijable). **Ako matrica dizajna ima više redaka nego stupaca, koja formulacija ima najmanje optimizacijskih varijabli?**
- ☐ A Primarni problem meke margine
  - ☐ B Primarni problem tvrde margine
  - ☐ C Dualni problem meke margine
  - ☐ D Dualni problem tvrde margine
5. (T) Kod algoritma SVM preporuča se napraviti skaliranje značajki. U protivnom, pri izračunu skalarnog produkta, značajke s većim rasponom (većom skalom) dominirat će nad značajkama s manjim rasponom (manjom skalom). Međutim, skaliranje značajki nije uvijek nužno. **Kada nije potrebno napraviti skaliranje značajki, i zašto?**
- ☐ A Kada se koristi RBF jezgra sa Mahalanobisovom udaljenošću, jer ta udaljenost uzima u obzir varijancu značajki
  - ☐ B Kada se koristi linearna jezgra i značajke su centrirane oko nule, jer se onda ne računa skalarni produkt između značajki
  - ☐ C Kada se koristi dualna formulacija SVM-a i algoritam SMO, jer se tada implicitno provodi L1-regularizacija
  - ☐ D Kada se koristi Gaussova jezgrena funkcija, jer ta jezgra implicitno inducira beskonačnodimenzijski prostor značajki
6. (T) Razmotrite problem maksimalne margine kao standardni problem optimizacije kriterijske funkcije  $f(\mathbf{x})$  uz ograničenja nejednakosti  $g_i(\mathbf{x}) \leq 0$  i jednakosti  $h_i(\mathbf{x}) = 0$ . Razmotrite taj problem za slučaj tvrde margine i za slučaj meke margine, oba u dualu. **Po čemu se dualni optimizacijski problem meke margine razlikuje od dualnog optimizacijskog problema tvrde margine?**
- ☐ A Ima dodatnih  $n$  ograničenja jednakosti
  - ☐ B Ima dodatnih  $N$  ograničenja nejednakosti
  - ☐ C Ima dodatnih  $n$  optimizacijskih varijabli i dodatnih  $N$  ograničenja jednakosti
  - ☐ D Ima dodatnih  $N$  optimizacijskih varijabli i dodatnih  $2N$  ograničenja nejednakosti
7. (T) Stroj potpornih vektora (SVM) može se koristiti kod linearno neodvojivih problema. Pretpostavite linearnu neodvojivost i razmotrite SVM za slučaj tvrde margine i meke margine. **Koji od sljedećih slučajeva će rezultirati najužom marginom?**
- ☐ A Meka margina s Gaussovom jezgrenom funkcijom i malim vrijednostima za  $C$  i  $\gamma$
  - ☐ B Meka margina s linearnom jezgrom i malom vrijednosti za  $C$
  - ☐ C Tvrda margina s linearnom jezgrom i velikom vrijednosti za  $C$
  - ☐ D Tvrda margina s Gaussovom jezgrenom funkcijom i velikom vrijednosti za  $\gamma$

## 10 Jezgrene metode

1. (T) Treniramo model SVM s nekom jezgrenom funkcijom. Nakon što smo naučili model na skupu primjera, za neki primjer  $\mathbf{x}$  želimo izračunati udaljenost tog primjera od hiperravnine u prostoru značajki. **Je li moguće izračunati tu udaljenost?**
  - ☐ A Ne, jer u dualnoj (neparametarskoj) formulaciji problema maksimalne margine nemamo vektor značajki
  - ☐ B Ne, jer granica između klasa u prostoru značajki općenito ne mora biti hiperravnina, već može biti hiperpovršina
  - ☐ C Da, ako nismo koristili Gaussovu jezgru ili neku složeniju jezgru koja koristi Gaussovu jezgru kao gradivni blok
  - ☐ D Da, ako smo koristili linearnu jezgru, odnosno ako je ulazni prostor jednak prostoru značajki
2. (T) Neke jezgrene funkcije nazivamo Mercerove jezgre ili pozitivno definitne jezgre. Takve jezgre daju pozitivno definitnu Gramovu matricu. **Zašto je dobro da je jezgrene funkcija Mercerova jezgra?**
  - ☐ A Zato što takva jezgra odgovara skalarnom produktu u nekom prostoru značajki, a to je nužno da bismo mogli primijeniti jezgreni trik
  - ☐ B Zato što takva jezgra inducira Hilbertov prostor, tj. prostor beskonačnodimenzijskih značajki, što nam daje potencijalno vrlo složene modele
  - ☐ C Zato što takva jezgra omogućava da, umjesto da vektoriziramo primjere, klasifikaciju određujemo na temelju sličnost između primjera i prototipnih primjera
  - ☐ D Zato što takva jezgra nužno daje nenegativne vrijednosti sličnosti između parova primjera, što je nužno kako gubitak ne bi bio negativan
3. (T) Gaussova jezgrene funkcija  $\kappa(\mathbf{x}_1, \mathbf{x}_2)$  nad primjerima  $\mathbf{x}_1$  i  $\mathbf{x}_2$  definirana je s parametrom preciznosti  $\gamma$ , gdje  $\gamma = 1/2\sigma^2$ . Ovaj parametar ima utjecaj na vrijednost jezgrene funkcije, ali i na složenost (nelinearnost) modela jezgrenog stroja s Gaussovom jezgrenom funkcijom. **Kakav je utjecaj parametra  $\gamma$  na vrijednost Gaussove jezgrene funkcije  $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ , gdje  $\mathbf{x}_1 \neq \mathbf{x}_2$ , te na nelinearnost modela jezgrenog stroja?**
  - ☐ A Što je vrijednost  $\gamma$  manja, to je manja vrijednost  $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ , i to je model manje nelinearan
  - ☐ B Što je vrijednost  $\gamma$  veća, to je veća vrijednost  $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ , i to je model manje nelinearan
  - ☐ C Što je vrijednost  $\gamma$  manja, to je manja vrijednost  $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ , i to je model više nelinearan
  - ☐ D Što je vrijednost  $\gamma$  veća, to je manja vrijednost  $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ , i to je model više nelinearan
4. (T) Može se dokazati da je Gaussova jezgra s hiperparametrom  $\gamma$  Mercerova jezgra. U praktičnome smislu, to znači da Gaussovu jezgru možemo koristiti za jezgreni trik umjesto da eksplicitno koristimo funkciju preslikavanja  $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^m$ . **Što to znači u matematičkome smislu?**
  - ☐ A  $\forall \mathbf{x}_1 \forall \mathbf{x}_2. \mathbf{x}_1^T \mathbf{x}_2 = \exp(-\gamma \|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\|^2)$
  - ☐ B  $\forall \mathbf{x}_1 \forall \mathbf{x}_2. \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2) = \exp(-\gamma \Delta^2)$  gdje  $\Delta^2 = (\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2)$
  - ☐ C  $\forall \mathbf{x}_1 \forall \mathbf{x}_2. \phi(\mathbf{x}_1^T \mathbf{x}_2) = \exp(-\gamma \Delta^2)$ , gdje  $\Delta^2 = (\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2)$
  - ☐ D  $\forall \mathbf{x}_1 \forall \mathbf{x}_2. (\mathbf{x}_1 \neq \mathbf{x}_2) \Rightarrow \left( \exp(-\gamma \|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\|^2) = \mathbf{x}_1^T \mathbf{x}_2 \right)$
5. (T) Važna prednost jezgrenih strojeva je mogućnost primjene jezgrenog trika. Ta je prednost pogotovo očita kada prostor ulaznih primjera  $\mathcal{X}$  nije euklidski vektorski prostor, odnosno kada primjere

nije moguće prikazati kao vektore realnih brojeva. **Koja je prednost primjene jezgrenog trika u takvom slučaju?**

- ☐ A Jezgrenim trikom implicitno se ostvaruje nelinearnost prostora značajki, što povećava kapacitet modela i povećava njegovu točnost
  - ☐ B Umjesto vektorizacije primjera, dovoljno je definirati nelinearnu funkciju preslikavanja iz ulaznog prostora u prostor značajki
  - ☐ C Jezgrena funkcija mjeri sličnost između primjera, čime se primjeri efektivno preslikavaju u beskonačnodimenzijski prostor značajki
  - ☐ D Jezgrena funkcija može biti mjera sličnosti između nevektoriziranih primjera, što implicitno inducira vektorski prostor značajki
6. (T) Stroj potpornih vektora (SVM) jedna je vrsta rijetkoga jezgrenog stroja. Jezgrena funkcija koristi jezgrene funkcije izračunate u odnosu na odabrane prototipne primjere. **Po čemu je SVM specifičan u odnosu na općeniti algoritam rijetkoga jezgrenog stroja?**
- ☐ A Zbog L1-regularizacije, mnoge težine modela bit će pritegnute na nulu
  - ☐ B Dimenzija prostora značajki ne može biti veća od broja primjera
  - ☐ C Broj parametara modela jednak je dimenziji prostora značajki
  - ☐ D Prototipni primjeri odabiru se u okviru optimizacijskog postupka
7. (T) Kažemo da Mercerove jezgrene funkcije implicitno definiraju prostor značajki. **Što to znači?**
- ☐ A Klasifikacija primjera definirana je na temelju umnoška jezgrene funkcije za taj primjer i sve druge primjere u skupu za učenje
  - ☐ B Jezgrena funkcija između primjera u prostoru značajki jednaka je skalarnom produktu tih primjera u ulaznom prostoru
  - ☐ C Broj dimenzija prostora značajki implicitno ovisi o broju klasa u ulaznom prostoru te može biti beskonačna
  - ☐ D Vrijednost jezgrene funkcije nad parom vektora jednaka je skalarnom produktu tih vektora nakon preslikavanja u prostor značajki
8. (T) Koristimo algoritam SVM s jezgrenom funkcijom  $\kappa$ . Optimizaciju provodimo u dualu. Ako je  $\kappa$  Mercerova jezgra, onda takva jezgra implicitno definira preslikavanje  $\phi$ . Međutim, u nekim slučajevima trebamo eksplicitno izraziti funkciju  $\phi$  na temelju korištene jezgre  $\kappa$ . **Kada je potrebno eksplicitno izraziti  $\phi$ ?**
- ☐ A Za izračun težine  $w_0$
  - ☐ B Za izračun funkcije gubitka
  - ☐ C Za izračun širine margine
  - ☐ D Za izračun dualnih koeficijenata

## 11 Neparametarske metode

1. (T) Algoritam SVM može biti parametarski i neparametarski, ovisno o tome provodimo li optimizaciju u primarnoj ili dualnoj formulaciji. U oba slučaja preferiramo da je model rijedak, tj. da je nakon treniranja što više parametara postavljeno na nulu. **Kako rijetkost modela ovisi o hiperparametru  $C$ ?**
- ☐ A Što je  $C$  manji, to je neparametarski model rjeđi, ali to nema utjecaja na rijetkost parametarskog modela jer on nema potporne vektore
  - ☐ B Što je  $C$  veći, to je neparametarski model manje rijedak, dok je parametarski to rjeđi jer  $\lambda$  pada
  - ☐ C Što je  $C$  manji, to je neparametarski model rjeđi, a također je to rjeđi i parametarski model jer  $\lambda$  raste
  - ☐ D Što je  $C$  veći, to je neparametarski model manje rijedak, dok parametarski model nije rijedak jer ima  $L_2$ -regularizaciju a ne  $L_1$ -regularizaciju



2. (T) Problem prokletstva dimenzionalnosti (engl. *curse of dimensionality*) pojavljuje se kod algoritama koji rade u visokodimenzijskom vektorskom prostoru i manifestira se na različite načine. **Kako se problem prokletstva dimenzionalnosti u visokodimenzijskim prostorima manifestira kod algoritma  $k$ -NN?**
  - ☐ A Udaljenosti između primjera se smanjuju i model  $k$ -NN postaje sve složeniji
  - ☐ B Povećava se broj susjeda u okolini svakog primjera i model  $k$ -NN postaje sve jednostavniji
  - ☐ C Svi primjeri su međusobno vrlo udaljeni i gube se razlike u udaljenosti
  - ☐ D Broj susjeda  $k$  nekog primjera se smanjuje i gube se granice između klasa
3. (T) Algoritmi strojnog učenja mogu biti parametarski ili neparametarski. **Što je karakteristika neparametarskih algoritama strojnog učenja?**
  - ☐ A Pretpostavljaju vjerojatnosnu distribuciju podataka
  - ☐ B Broj parametara ovisi o broju primjera
  - ☐ C Eksplicitno modeliraju granicu između primjera
  - ☐ D Svaki primjer ima globalan utjecaj na izgled hipoteze
4. (T) Nalaženje najbližih susjeda kod algoritma  $k$ -NN predstavlja izazov zbog računalne složenosti. Algoritam stabla lopti (engl. *ball tree*) jedan je od pristupa za smanjenje računalne složenosti dohvaćanja susjeda u visokodimenzijskom vektorskom prostoru. **Na koji način funkcionira algoritam stabla lopti?**
  - ☐ A Koristi preslikavanje osjetljivo na lokalne promjene u vektoru kojim se bliske točke pohranjuju u iste pretince
  - ☐ B Koristi pretraživanje duž pravca u vektorskom prostoru u smjeru suprotnome od gradijenta funkcije pogreške
  - ☐ C Koristi brzo pretraživu binarnu indeksnu strukturu za particioniranje prostora primjera u preklapajuće regije
  - ☐ D Koristi jezgri trikotnik za izračun euklidske udaljenosti između točke upita i svih drugih točaka u skupu primjera

## 14 Procjena parametara II

1. (T) Funkcija izglednosti  $\mathcal{L}(\theta|\mathcal{D})$  nije isto što i vjerojatnost. **Po čemu se izglednost razlikuje od vjerojatnosti?**
  - ☐ A Funkcija izglednosti  $\mathcal{L}(\theta|\mathcal{D})$  jednaka je gustoći vjerojatnosti  $p(\theta|\mathcal{D})$ , ali, za razliku od gustoće vjerojatnosti, nije odozgo ograničena sa 1
  - ☐ B Ako su podatci diskretni (kategoričke značajke), onda je funkcija izglednosti parametara  $\theta$  isto što i zajednička vjerojatnost uzorka  $\mathcal{D}$  i parametara  $\theta$
  - ☐ C Funkcija izglednosti  $\mathcal{L}(\theta|\mathcal{D})$  jednaka je gustoći vjerojatnosti  $p(\mathcal{D}|\theta)$ , samo što je izglednost funkcija parametara  $\theta$ , dok je  $p(\mathcal{D}|\theta)$  funkcija uzorka  $\mathcal{D}$
  - ☐ D Za razliku od vjerojatnosti, funkcija izglednosti  $\mathcal{L}(\theta|\mathcal{D})$  je simetrična, u smislu da vrijedi  $\mathcal{L}(\theta|\mathcal{D}) = p(\mathcal{D}|\theta)$
2. (T) Treniranje probabilističkih modela svodi se na procjenu njihovih parametara  $\theta$  na temelju funkcije log-izglednosti,  $\mathcal{L}(\theta|\mathcal{D})$ . **Što je zapravo funkcija log-izglednosti?**
  - ☐ A Funkcija koja primjeru  $\mathbf{x}$  pridjeljuje vjerojatnost  $p(y|\mathbf{x})$ , uz pretpostavku da se ta vjerojatnost pokorava distribuciji definiranoj modelom  $h(\mathbf{x}; \theta)$
  - ☐ B Funkcija koja uzorku  $\mathcal{D}$  pridjeljuje vjerojatnost parametara  $\theta$ , uz pretpostavku da se parametri pokoravaju distribuciji definiranoj modelom  $h(\mathbf{x}; \theta)$
  - ☐ C Funkcija koja parametrima  $\theta$  pridjeljuje vjerojatnost uzorka  $\mathcal{D}$ , uz pretpostavku da se uzorak pokorava distribuciji definiranoj modelom  $h(\mathbf{x}; \theta)$
  - ☐ D Funkcija koja primjeru  $(\mathbf{x}, y)$  pridjeljuje vjerojatnost pripadanja skupu označenih primjere  $\mathcal{D}$ , uz pretpostavku da su primjeri nezavisno i identično distribuirani

3. (T) Nepristranost je jedno od svojstava statističkih procjenitelja. Procjenitelj MLE ne mora nužno biti nepristran, tj. može biti pristran. **Za koji od sljedećih parametara distribucije procjena MLE pristrana?**
- ☐ A Srednja vrijednost Gaussove distribucije
  - ☐ B Varijanca Bernoullijeve distribucije
  - ☐ C Srednja vrijednost Bernoullijeve distribucije
  - ☐ D Kovarijacijska matrica Gaussove distribucije
4. (T) MAP-procjenitelj definiramo kao  $\hat{\theta}_{\text{MAP}} = \operatorname{argmax}_{\theta} p(\mathcal{D}|\theta)p(\theta)$ . Pri odabiru apriorne distribucije  $p(\theta)$ , nastojimo da je to neka standardna teorijska distribucija i da je konjugatna distribucija za izglednost  $\mathcal{L}(\theta|\mathcal{D})$ . **Što to znači i zašto to želimo?**
- ☐ A To znači da je apriorna distribucija ista vrsta distribucije kao i vjerojatnost podataka uz dane parametre, tj. izglednost parametara, pa će njihov umnožak biti distribucija koja je proporcionalna aposteriornoj distribuciji i čiji ćemo maksimum moći izračunati Bayesovim pravilom
  - ☐ B To znači da je apriorna distribucija upravljana hiperparametrima kojima možemo ugoditi distribucija parametara koji procjenjujemo, tj. parametri apriorne distribucije i parametri izglednosti su identični, što nam omogućava da te dvije distribucije pomnožimo i zatim nađemo maksimizator
  - ☐ C To znači da je aposteriorna distribucija parametara ista kao izglednost parametara, pa primenom Bayesovog pravila možemo izračunati apriornu vjerojatnost parametara te, nakon zanemarivanja nazivnika koji je za fiksiran skup podataka konstantan, pronaći parametre koji maksimiziraju aposterionu vjerojatnost
  - ☐ D To znači da će umnožak izglednosti i apriorne distribucije dati distribuciju koja je iste vrste kao i apriorna distribucija, a ako je riječ o standardnoj teorijskoj distribuciji iz eksponencijalne familije, njezin mod (maksimizator) postoji u zatvorenoj formi, što nam omogućava da procjenitelj izračunamo analitički
5. (T) Kod MAP-procjenitelja, apriorna distribucija parametra  $p(\theta)$  tipično se odabire tako da bude konjugatna za funkciju izglednosti  $p(\mathcal{D}|\theta)$ . Pretpostavimo da MAP-procjenitelj izračunavamo heurističkom metodom (npr., gradijentnim usponom). Što se događa ako za apriornu distribuciju parametra upotrijebimo distribuciju koja *nije* konjugatna funkciji izglednosti?
- ☐ A Aposteriornu distribuciju  $p(\theta|\mathcal{D})$  ne možemo izvesti u zatvorenoj formi, ali MAP možemo izračunati heurističkom optimizacijom
  - ☐ B Zajedničku distribuciju  $p(\mathcal{D}, \theta)$  ne možemo izvesti u zatvorenoj formi, pa MAP nije definiran
  - ☐ C Ako je apriorna distribucija  $p(\theta)$  iz eksponencijalne familije, onda je aposteriona distribucija  $p(\theta|\mathcal{D})$  u zatvorenoj formi i MAP je izračunljiv
  - ☐ D Neovisno o apriornoj distribuciji parametra  $p(\theta)$ , MAP je izračunljiv optimizacijom drugog reda (npr., Newtonovim postupkom)
6. (T) Parametre modela možemo procijeniti pomoću procjenitelja najveće izglednosti (MLE) ili procjenitelja najveće aposteriorne vjerojatnosti (MAP). Općenito, MLE i MAP daju različite procjene, no u nekim slučajevima mogu dati jednake procjene. **Kada će MLE i MAP dati jednake procjene?**
- ☐ A Kada  $p(\theta)$  definiramo kao unimodalnu distribuciju
  - ☐ B Kada broj primjera  $N$  teži prema beskonačno
  - ☐ C Kada je  $p(\theta)$  konjugatna izglednosti  $p(\mathcal{D}|\theta)$
  - ☐ D Kada se u  $\mathcal{D}$  realizirala svaka vrijednost slučajne varijable

7. (T) Za Bayesov klasifikator procjenjujemo parametar  $\mu$  Bernoullijeve distribucije. Skup primjera za učenje je razmjerno malen. Naš procjenitelj  $\hat{\mu}$  parametra  $\mu$  može biti pristran ili nepristran, dok model čiji je parametar procijenjen s  $\hat{\mu}$  može biti prenaučan ili može dobro generalizirati. Razmotrite procjenitelje MLE i MAP (konkretno, Laplaceovo zaglađivanje). **Što od sljedećega općenito vrijedi u ovom slučaju?**
- ☐ A MLE procjenitelj je pristran i očekujemo da će model loše generalizirati
  - ☐ B MAP procjenitelj je pristran, ali očekujemo da će model dobro generalizirati
  - ☐ C MLE procjenitelj je nepristran i očekujemo da će model dobro generalizirati
  - ☐ D MAP procjenitelj je nepristran i očekujemo da će model dobro generalizirati
8. (T) Procjenitelj MAP kombinira funkciju izglednosti parametara s apriornom distribucijom parametara. **Što i kako maksimizira procjenitelj MAP?**
- ☐ A Zajedničku gustoću vjerojatnosti parametara i podataka, u zatvorenoj formi, ako je apriorna distribucija konjugatna za izglednost, inače iterativno
  - ☐ B Aposteriornu vjerojatnost podataka, u zatvorenoj formi ako je zajednička vjerojatnost parametara i podataka iz eksponencijalne familije, inače iterativno
  - ☐ C Zajedničku vjerojatnost parametara i podataka, iterativno ako je apriorna distribucija iz eksponencijalne familije, inače u zatvorenoj formi
  - ☐ D Aposteriornu gustoću vjerojatnosti podataka, u zatvorenoj formi ako su izglednost i apriorna distribucija konjugatne, inače iterativno

## 15 Bayesov klasifikator

1. (T) Probabilistički modeli mogu biti generativni ili diskriminativni. U praksi su diskriminativni modeli nerijetko veće klasifikacijske točnosti od generativnih modela. **Zašto je tomu tako?**
- ☐ A Kod generativnih modela parametri se procjenjuju metodom najveće izglednosti koja je pristrana, dok se diskriminativni modeli uče gradijentnim spustom koji je statistički nepristran
  - ☐ B Diskriminativni modeli s manje parametara mogu modelirati istu granicu između klasa kao i generativni modeli, pa trebaju manje primjera da ih se nauči, a i teže ih je prenaučiti
  - ☐ C Generativni modeli mogu modelirati nelinearne zavisnosti između značajki, međutim kada su značajke stohastički nezavisne, to dovodi do prenaučenosti modela
  - ☐ D Diskriminativni modeli modeliraju zajedničku distribuciju primjera i oznaka, pa u slučaju preklapajućih distribucija u ulaznome prostoru ostvaruju veću točnost od generativnih modela
2. (T) Jedan od nedostataka generativnih modela u odnosu na diskriminativne modele jest nepotrebna složenost modeliranja. **Što to zapravo znači?**
- ☐ A Zajednička distribucija  $p(\mathbf{x}, y)$  može se faktorizirati kao  $p(y|\mathbf{x})p(\mathbf{x})$ , no takva faktorizacija ima više parametara
  - ☐ B Generativni modeli modeliraju distribuciju  $p(y|\mathbf{x})$ , što iziskuje više parametara nego li modeliranje granice između klasa
  - ☐ C Za razliku od diskriminativnih modela, generativni modeli distribuciju  $p(y|\mathbf{x})$  definiraju zasebno za svaku klasu, pa stoga imaju više parametara
  - ☐ D Za klasifikaciju nam je potrebna samo distribucija  $p(y|\mathbf{x})$ , i ona se može modelirati sa manje parametara od zajedničke distribucije  $p(\mathbf{x}, y)$
3. (T) Bayesov klasifikator definirali smo na sljedeći način:  $h_j(\mathbf{x}; \theta) = P(y = j|\mathbf{x}) = \frac{p(\mathbf{x}|y)P(y)}{\sum_{y'} p(\mathbf{x}|y')P(y')}$ . Broj klasa neka je dva. Značajke neka su diskretne, i to neke s dvije, a neke s više od dvije moguće vrijednosti. **Koje teorijske distribucije ćemo koristiti za  $P(y)$  i  $P(\mathbf{x}|y)$ ?**
- ☐ A Bernoullijevu distribuciju za  $P(y)$  i multinulijevu distribuciju za  $P(\mathbf{x}|y)$
  - ☐ B Kategoričku distribuciju za  $P(y)$  i Gaussovu distribuciju za  $P(\mathbf{x}|y)$
  - ☐ C Bernoullijevu distribuciju za  $P(y)$  i Gaussovu distribuciju za  $P(\mathbf{x}|y)$
  - ☐ D Gaussovu distribuciju za  $P(y)$  i multinulijevu distribuciju za  $P(\mathbf{x}|y)$

4. (T) Bayesov klasifikator definirali smo na sljedeći način:

$$h_j(\mathbf{x}; \boldsymbol{\theta}) = P(y = j|\mathbf{x}) = \frac{p(\mathbf{x}|y)P(y)}{\sum_{y'} p(\mathbf{x}|y')P(y')}$$

Neka je broj klasa veći od dva,  $K > 2$ , a značajke neka su realni brojevi,  $\mathbf{x} \in \mathbb{R}^n$ . **Koje teorijske distribucije ćemo koristiti za  $P(y)$  i  $P(\mathbf{x}|y)$ ?**

- ☐ A Bernoullijevu distribuciju za  $P(y)$  i Gaussovu distribuciju za  $P(\mathbf{x}|y)$
- ☐ B Kategoričku distribuciju za  $P(y)$  i Gaussovu distribuciju za  $P(\mathbf{x}|y)$
- ☐ C Gaussovu distribuciju za  $P(y)$  i multinulijevu distribuciju za  $P(\mathbf{x}|y)$
- ☐ D Kategoričku distribuciju za  $P(y)$  i za  $P(\mathbf{x}|y)$

5. (T) Bayesov klasifikator definiran je kao

$$h(\mathbf{x}; \boldsymbol{\theta}) = \operatorname{argmax}_y p(\mathbf{x}|y)P(y)$$

**Po čemu se vidi da je ovo generativan, a ne diskriminativan model?**

- ☐ A Zajedničku vjerojatnost primjera i oznaka,  $p(\mathbf{x}|y)P(y)$ , faktorizira u dva faktora te zanemaruje nazivnik  $p(\mathbf{x})$ , koji je ionako konstantan za svaku klasu  $y$
  - ☐ B Parametre distribucija  $p(\mathbf{x}|y)$  i  $P(y)$ , a time indirektno i parametre aposteriorne distribucije  $P(y|\mathbf{x})$ , računa MAP-procjenitelj, čime sprječava prenaučenosť
  - ☐ C Modelira vjerojatnost primjera i oznaka, budući da je, na temelju pravila umnoška, umnožak  $p(\mathbf{x}|y)P(y)$  jednak zajedničkoj vjerojatnosti  $p(\mathbf{x}, y)$
  - ☐ D Primjer  $\mathbf{x}$  klasificira prema MAP-hipotezi, dakle u klasu koja maksimizira aposteriornu vjerojatnost oznake,  $p(y|\mathbf{x})$ , koja je proporcionalna zajedničkoj vjerojatnosti primjera i oznaka,  $p(\mathbf{x}, y)$
6. (T) Jedan od parametara Gaussovog Bayesovog klasifikatora je kovarijacijska matrica  $\boldsymbol{\Sigma}$  Gaussove multivarijatne distribucije. Broj parametara matrice  $\boldsymbol{\Sigma}$  koje treba procijeniti može se smanjiti uvođenjem pretpostavki o distribuciji primjera u ulaznome prostoru. **Uz koje minimalne pretpostavke će broj parametara za  $\boldsymbol{\Sigma}$  biti linearan u broju značajki  $n$ ?**
- ☐ A Šum je isti za sve klase
  - ☐ B Značajke nisu korelirane i šum ne ovisi o klasi
  - ☐ C Šum je isti za sve klase i sve značajke
  - ☐ D Nema linearne zavisnosti između značajki

## 16 Bayesov klasifikator II

1. (T) Gaussov Bayesov klasifikator s dijeljenom kovarijacijskom matricom (GBC) i logistička regresija (LR) su generativno-diskriminativni par modela. **Što to znači?**
- ☐ A Aposteriorna vjerojatnost klase za GBC može se izraziti kao poopćeni linearni model sa sigmoidnom aktivacijskom funkcijom
  - ☐ B GBC i neregularizirana LR modeliraju identične distribucije zajedničke vjerojatnosti primjera i oznaka, ali s različitim brojem parametara
  - ☐ C Izlaz modela LR jednak je zajedničkoj vjerojatnosti modela GBC, ali model LR iziskuje manje parametara
  - ☐ D Neovisno o optimizacijskom postupku, GBC i LR ostvaruju istu pogrešku na skupu za učenje, ali uz moguće različite brojeve parametara

2. (T) Gaussov Bayesov klasifikator s dijeljenom kovarijacijskom matricom (GBC) i logistička regresija bez regularizacije (LR) čine generativno-diskriminativni par modela. **Što je od sljedećeg točno za taj generativno-diskriminativan par modela?**
- ☐ A GBC i LR optimiraju različite funkcije pogreške, pa mogu dati različite granice između klasa
  - ☐ B Učenje modela GBC općenito je računalno složenije od učenja modela LR
  - ☐ C Oba modela daju linearnu granicu između klasa, ali GBC općenito ima više parametara od LR
  - ☐ D Vjerojatnosni izlaz modela GBC odgovara komplementu vjerojatnosnog izlaza modela LR
3. (T) Umjesto naivnog Bayesov klasifikatora, ponekad je bolje koristiti polunaivan Bayesov klasifikator? **Po čemu se polunaivan Bayesov klasifikator (SNBC) razlikuje od naivnog Bayesovog klasifikatora (NBC)?**
- ☐ A SNBC pretpostavlja zavisnosti između značajki i klase, ali ima manje parametara od NBC
  - ☐ B SNBC ima više parametara nego NBC, ali manje bridova kada ga se prikaže kao Bayesovu mrežu
  - ☐ C SNBC zajedničku vjerojatnost faktorizira u manje faktora, pa ima i manje parametara od NBC
  - ☐ D SNBC ima više parametara od NBC te modelira zavisnosti između značajki
4. (T) Polunaivan Bayesov klasifikator združuje u jedan faktor varijable kod kojih postoji statistička zavisnost. Za procjenu statističke zavisnosti između varijabli može se upotrijebiti Kullback-Leiblerova divergencija (KL-divergencija). **Kako se pomoću KL-divergencije može izračunati koliko su varijable zavisne?**
- ☐ A Što su varijable manje zavisne, to je manja KL-divergencija između marginalnih vjerojatnosti i uvjetne vjerojatnosti
  - ☐ B Što su varijable manje zavisne, to je veća KL-divergencija između faktorizirane vjerojatnosti i marginalne vjerojatnosti
  - ☐ C Što su varijable više zavisne, to je veća KL-divergencija između zajedničke vjerojatnosti i faktorizirane vjerojatnosti
  - ☐ D Što su varijable više zavisne, to je manja KL-divergencija između zajedničke vjerojatnosti i faktorizirane vjerojatnosti

## 17 Probabilistički grafički modeli

1. (T) Za Bayesovu mrežu kažemo da je generativni i parametarski model. **Zašto?**
- ☐ A Generativni jer definira zajedničku vjerojatnost svih varijabli, i opaženih i skrivenih, a parametarski jer se parametri modela mogu dobiti MLE-procjenom za svaki čvor Bayesove mreže zasebno, budući da se log-izglednost dekomponira po strukturi mreže
  - ☐ B Generativni jer se može koristiti za generiranje skupa primjera na temelju zajedničke distribucije, a parametarski jer su broj čvorova mreže i njihovo povezivanje (dakle graf) definirani parametrima koji se mogu ugađati na skupu za učenje, čime se mogu dobiti različite strukture mreže
  - ☐ C Generativni jer svaki čvor odgovara uvjetnoj vjerojatnosti koja je, na temelju Markovljevog uređajnog svojstva, generirana distribucijama čvorova roditelja, a parametarski jer Bayesova mreža zapravo definira zajedničku distribuciju koja je opisana skupom parametara
  - ☐ D Generativni jer opisuje postupak kojim se mogu generirati podatci koji se pokoravaju određenoj zajedničkoj vjerojatnosnoj distribuciji, a parametarski jer svaki čvor Bayesove mreže definira uvjetnu vjerojatnost preko teorijske distribucije koja je opisana svojim parametrima

2. (T) Bayesove mreže na sažet način prikazuju zajedničku distribuciju te kodiraju uvjetne stohastičke nezavisnosti između varijabli. No, kao i svaki model strojnog učenja, tako se i Bayesove mreže mogu prenaučiti. **Koja je veza između uvjetnih nezavisnosti varijabli u Bayesovoj mreži i opasnosti od prenaučnosti?**
- ☐ A Uvođenjem pretpostavki o uvjetnoj nezavisnosti povećava se broj čvorova mreže, a time i broj parametara, što model čini složenijim i time sklonijim prenaučnosti
  - ☐ B Uvođenje pretpostavki o uvjetnoj nezavisnosti pojednostavljuje strukturu Bayesove mreže i smanjuje broj parametara, čime se smanjuje i mogućnost prenaučnosti
  - ☐ C Uvjetne nezavisnosti određuju strukturu mreže na način da definiraju koji su čvorovi mreže međusobno povezani, međutim to nema utjecaja na složenost modela niti na sklonost prenaučnosti
  - ☐ D Pretpostavke o uvjetnoj nezavisnosti čine induktivnu pristranost modela, pa što je više uvjetnih nezavisnosti, to je veća pristranost i model je lako prenaučiti
3. (T) Bayesova mreža na sažet način definira zajedničku distribuciju vjerojatnosti uz određene pretpostavke uvjetne nezavisnosti. Neka je jedna takva pretpostavka  $x_1 \perp \{x_2, x_3\} | x_4$ . **Koji je efekt uvođenja ove pretpostavke na graf Bayesove mreže?**
- ☐ A Dodavanje dva brida
  - ☐ B Dodavanje tri brida
  - ☐ C Uklanjanje dva brida
  - ☐ D Uklanjanje tri brida
4. (T) Skriveni Markovljev model (HMM) posebna je vrsta Bayesove mreže. **Na što se odnosi pridjev “skriveni” u nazivu tog modela?**
- ☐ A Model opisuje prijelaze između stanja, a trenutačno stanje ovisi samo o prethodnom stanju
  - ☐ B Neke varijable modela nisu opažene u podacima, ali zavise o opaženim varijablama
  - ☐ C Opažene varijable ovise samo o trenutačnom stanju i prethodno opaženoj varijabli
  - ☐ D Stanja modela poredana su u lanac, a izlazi modela vidljivi su samo za posljednje stanje
5. (T) Bayesova mreža može se upotrijebiti za modeliranje kauzalnih odnosa između varijabli, odnosno za zaključivanje o uzrocima i posljedicama događaja. Jedan primjer takvog zaključivanja jest “efekt objašnjavanja”, koji se primjenjuje kada postoji interakcija uzorka nekog događaja. **Gdje u Bayesovoj mreži nastupa efekt objašnjavanja i kako se on manifestira?**
- ☐ A U strukturi  $x \rightarrow z \leftarrow y$ , gdje su uzroci  $x$  i  $y$  nezavisni, ali postaju zavisni ako je opažena posljedica  $z$
  - ☐ B U strukturi  $x \rightarrow z \rightarrow y$ , gdje su uzrok  $x$  i posljedica  $y$  zavisni, ali postaju nezavisni ako je opažen posredni uzrok  $z$
  - ☐ C U strukturi  $x \rightarrow z \leftarrow y$ , gdje su uzroci  $x$  i  $y$  zavisni, ali postaju nezavisni ako je opažena posljedica  $z$
  - ☐ D U strukturi  $x \rightarrow z \rightarrow y$ , gdje su uzrok  $x$  i posljedica  $y$  nezavisni, ali postaju zavisni ako je opažen posredni uzrok  $z$
6. (T) Probablistički grafički modeli na sažet način prikazuju zajedničku distribuciju  $n$  slučajnih varijabli. Ta je distribucija prikazana usmjerenim grafom. Neka graf sadrži čvor označen sa  $x$ . **Čemu odgovara taj čvor?**
- ☐ A Zajedničkoj vjerojatnosti varijable  $x$  i varijabli roditeljskih čvorova od  $x$
  - ☐ B Marginalnoj vjerojatnosti  $P(x)$  ili marginalnoj gustoći vjerojatnosti  $p(x)$
  - ☐ C Distribuciji varijable  $x$  uz opažene sve varijable koje varijabli  $x$  prethode u topološkom uređaju
  - ☐ D Distribuciji varijable  $x$  uvjetovane vrijednostima varijabli roditeljskih čvorova od  $x$

## 18 Probabilistički grafički modeli II

1. (T) Čest način probabilističkog zaključivanja kod Bayesovih mreža jest izračunavanje “aposteriornog upita”. Kod te vrste upita zanima nas distribucija nekih varijabli (varijable upita) na temelju zadanih varijabli (opažene varijable). Međutim, Bayesova mreža kodira zajedničku vjerojatnost svih varijabli mreže, uključivo i varijabli koje nisu niti varijable upita niti opažene varijable (varijable smetnje). **Na koji način izračunavamo aposteriorni upit?**
  - ☐ A Kao omjer zajedničke vjerojatnosti marginalizirane po varijablama smetnje i zajedničke vjerojatnosti marginalizirane po varijablama smetnje i varijablama upita
  - ☐ B Kao omjer zajedničke vjerojatnosti marginalizirane po varijablama upita i zajedničke vjerojatnosti marginalizirane po opaženim varijablama
  - ☐ C Kao umnožak zajedničke vjerojatnosti marginalizirane po opaženim varijablama i zajedničke vjerojatnosti marginalizirane po varijablama smetnje
  - ☐ D Kao umnožak vjerojatnosti varijable upita uvjetovane na opažene varijable i zajedničke vjerojatnosti marginalizirane po varijablama smetnje
2. (T) Glavna svrha probabilističkih grafičkih modela (PGM) jest provođenje probabilističkih upita. Jedna vrsta upita su MAP-upiti (upiti najvjerojatnijeg objašnjenja). Neka su  $\mathbf{x}_q$ ,  $\mathbf{x}_o$  i  $\mathbf{x}_n$  skupovi varijabli upita, opaženih varijabli odnosno varijabli smetnje. **Kako je definiran rezultat MAP-upita?**
  - ☐ A  $\operatorname{argmax}_{\mathbf{x}_o} \sum_{\mathbf{x}_n} p(\mathbf{x}_q, \mathbf{x}_o, \mathbf{x}_n)$
  - ☐ B  $\operatorname{argmax}_{\mathbf{x}_o} \sum_{\mathbf{x}_q} p(\mathbf{x}_q, \mathbf{x}_o, \mathbf{x}_n)$
  - ☐ C  $\operatorname{argmax}_{\mathbf{x}_q} \sum_{\mathbf{x}_o} p(\mathbf{x}_q, \mathbf{x}_o, \mathbf{x}_n)$
  - ☐ D  $\operatorname{argmax}_{\mathbf{x}_q} \sum_{\mathbf{x}_n} p(\mathbf{x}_q, \mathbf{x}_o, \mathbf{x}_n)$
3. (T) Približno zaključivanje kod Bayesovih mreža može se provesti metodama uzorkovanja. Te metode generiraju slučajan uzorak primjera iz distribucije opisane Bayesovom mrežom, iz kojega se onda mogu procijeniti parametri distribucije. Međutim, da bismo mogli uzorkovati iz Bayesove mreže, već trebamo znati parametre svih uvjetnih distribucija u čvorovima. **Zašto radimo uzorkovanje iz Bayesove mreže da bismo procijenili parametre distribucije, kad te parametre već imamo?**
  - ☐ A Ne trebamo znati parametre svih čvorova, već samo čvorova roditelja onih distribucija za koje želimo procijeniti parametre
  - ☐ B Vrijednosti parametara svih distribucija su procjene koje inicijaliziramo i zatim iterativno ažuriramo u svakom koraku uzorkovanja
  - ☐ C Uzorkovanje koristimo da bismo procijenili parametre bilo koje distribucije, a ne samo uvjetnih distribucija u čvorovima mreže
  - ☐ D Uzorkovanjem procjenjujemo parametre latentnih varijabli, čije vrijednosti ne opažamo, pa nam ti parametri nisu poznati prije uzorkovanja
4. (T) Bayesove mreže možemo upotrijebiti za procjenu aposteriorne vjerojatnosti  $P(\mathbf{x}_q|\mathbf{x}_o)$ , gdje je  $\mathbf{x}_q$  vektor varijabli upita a  $\mathbf{x}_o$  vektor opaženih varijabli. U tu se svrhu često koriste metode približnog zaključivanja. Najjednostavnija takva metoda je uzorkovanje s odbacivanjem, međutim ta metoda ima nedostatak zbog koje se u praksi ne koristi. **Koji je nedostatak metode uzorkovanja s odbacivanjem?**
  - ☐ A Ako je vjerojatnost  $P(\mathbf{x}_o)$  mala, treba generirati mnogo vektora da bi uzorak bio velik i procjena pouzdana
  - ☐ B Ako je vjerojatnost  $P(\mathbf{x}_o)$  velika, mnogo će generiranih vektora biti odbačeno i procjena će biti pristrana
  - ☐ C Ako je vjerojatnost  $P(\mathbf{x}_o)$  mala, generiramo vektore iz apriorne distribucije  $P(\mathbf{x}_q)$  i procjena je pristrana
  - ☐ D Ako je vjerojatnost  $P(\mathbf{x}_o)$  velika, generirani vektori nisu iz aposteriorne distribucije i procjena je netočna

5. (T) Unaprijedno uzorkovanje jedna je od metoda za uzorkovanje iz distribucije opisane Bayesovom mrežom. **Koji je problem kod primjene unaprijednog uzorkovanja za izračun aposteriornih upita nad Bayesovom mrežom?**
- ☐ A Mnogo uzorkovanih vektora morat će biti odbačeni, pa procjena neće biti pouzdana
  - ☐ B Dobiveni vektori bit će uzorkovani iz apriorne, a ne aposteriorne distribucije
  - ☐ C Kod marginalizacije varijabli smetnje dolazi do kombinatorne eksplozije
  - ☐ D Skrivenne varijable nisu opažene, pa se log-izglednost ne dekomponira po varijablama mreže
6. (T) Procjena parametara Bayesove mreže temelji se na maksimizaciji log-izglednosti parametara pod modelom. Procjena parametara može biti bitno drugačija za slučaj potpunih podataka, gdje su sve varijable opažene, u odnosu na slučaj nepotpunih podataka, gdje u model trebamo uključiti skrivene ili latentne varijable. **Što je prednost procjene parametara kod potpunih podataka (modela bez skrivenih varijabli) u odnosu na nepotpune podatke (modela sa skrivenim varijablama)?**
- ☐ A Kod potpunih podataka minimizacija funkcije log-izglednosti ima rješenje u zatvorenoj formi, ali funkcija nije konkavna, pa može imati više lokalnih optimuma, za razliku od modela sa skrivenim varijablama koji ima više parametara, ali konkavnu funkciju log-izglednosti
  - ☐ B Kod potpunih podataka maksimizacija log-izglednosti ima rješenje u zatvorenoj formi, ali samo ako su opažene varijable na početku niza po topološkom uređaju čvorova, za razliku od modela sa skrivenim varijablama kod kojega MLE procjenitelj ne postoji u zatvorenoj formi
  - ☐ C Kod potpunih podataka MLE procjena parametara ima rješenje u zatvorenoj formi, dok MAP procjena nema, za razliku od modela sa skrivenim varijablama kod kojeg je situacija obrnuta, a k tome taj model ima još više parametara od modela bez skrivenih varijabli
  - ☐ D Kod potpunih podataka log-izglednost se dekomponira po strukturi mreže, pa parametre svake uvjetne distribucije možemo procijeniti nezavisno od drugih čvorova i u zatvorenoj formi, međutim parametara može biti više nego kod modela sa skrivenim varijablama
7. (T) Parametre probabilističkih grafičkih modela, uključivo Bayesove mreže, možemo procjenjivati iz potpunih podataka ili nepotpunih podataka. **Zašto i kako Bayesovu mrežu učimo nad nepotpunim podacima?**
- ☐ A Jer se log-izglednost dekomponira po čvorovima mreže, pa MLE ili MAP procjenjujemo u zatvorenoj formi
  - ☐ B Jer MLE nema rješenje u zatvorenoj formi, pa umjesto MLE koristimo gradijentni uspon ili EM-algoritam
  - ☐ C Jer mreža ima skrivene varijable koje ne opažamo, pa moramo koristiti iterativne metode za MAP ili MLE
  - ☐ D Jer mreža ima manje čvorova nego što je opaženih varijabli, pa koristimo eliminaciju varijabli
8. (T) Za izvođenje probabilističkih upita nad PGM-ovima potrebno je uzorkovati iz uvjetne distribucije  $P(\mathbf{x}_q|\mathbf{x}_o)$ . Naivan pristup bio bi fiksirati vrijednosti svih varijabli  $\mathbf{x}_o$  na opažene vrijednosti, a za sve ostale varijable primijeniti unaprijedno uzorkovanje. **Zbog čega takvo uzorkovanje općenito nije dobro?**
- ☐ A Uzorkujemo iz distribucije koja je različita od  $P(\mathbf{x}_q|\mathbf{x}_o)$
  - ☐ B Ako je vjerojatnost  $P(\mathbf{x}_o)$  mala, uzorak će biti malen
  - ☐ C Ako čvorovi iz  $\mathbf{x}_o$  nemaju roditelje, uzorak će biti pristran
  - ☐ D Postupak nije primjenjiv ako čvorovi iz  $\mathbf{x}_q$  imaju roditelje



## 19 Grupiranje

1. (T) Algoritam K-sredina je iterativan algoritam za nalaženje parametara  $b_k^{(i)}$  (pripadnosti primjera grupama) i  $\mu_k$  (centroidi grupa) koji minimiziraju kriterijsku funkciju  $J$  za  $N$  primjera i  $K$  grupa. Algoritam funkciju  $J$  optimizira iterativno, jer rješenje u zatvorenoj formi ne postoji. **Zbog čega za problem minimizacije funkcije  $J$  ne postoji rješenje u zatvorenoj formi?**
  - ☐ A Jer za  $b_k^{(i)}$  mora vrijediti ograničenje  $\sum_k b_k^{(i)} = 1$  i  $b_k^{(i)} \geq 0$
  - ☐ B Jer  $J$  ovisi o  $K$  i inicijalnom odabiru za  $\mu_k$ , pa rješenje nije jedinstveno
  - ☐ C Jer  $b_k^{(i)}$  ovisi o vektorima  $\mu_k$ , a vektor  $\mu_k$  ovisi o vrijednostima  $b_k^{(i)}$
  - ☐ D Jer  $b_k^{(i)}$  ovisi o  $N$ , broj vektora  $\mu_k$  ovisi o  $K$ , a  $K$  je odozgo ograničen sa  $N$
2. (T) Konvergencija je poželjno svojstvo algoritma grupiranja. **Je li točno da algoritam k-sredina uvijek konvergira?**
  - ☐ A Da, algoritam uvijek konvergira zato što je broj particija  $N$  primjera u  $K$  skupova ograničen, a optimizacijski postupak definiran je tako da se  $J$  u svakoj iteraciji smanjuje
  - ☐ B Algoritam konvergira samo ako su početna središta dobro odabrana, inače se može dogoditi da algoritam oscilira između dva rješenja
  - ☐ C Kako se radi o algoritmu koji grupira primjere u vektorskom prostoru, broj rješenja je neograničen, stoga algoritam ne mora konvergirati
  - ☐ D Algoritam uvijek konvergira zato što je broj primjera  $N$  uvijek veći ili jednak broju grupa  $K$ , a kao mjera udaljenosti koristi se euklidska udaljenost, koja je nužno nenegativna
3. (T) Algoritam K-means++ proširenje je algoritma K-sredina heurističkim odabirom početnih središta grupa. **Koja je glavna ideja odabira početnih središta grupa kod algoritma K-means++?**
  - ☐ A Vjerojatnost da je neki primjer središte grupe proporcionalna je s brojem primjera u toj grupi i udaljenosti tih primjera od centroida skupa podataka
  - ☐ B Središta grupe su srednje vrijednosti grupa projiciranih na pravac u smjeru prve komponente rastava skupa podataka na glavne komponente (PCA)
  - ☐ C Središta grupa su mjesta na kojima graf kriterijske funkcije u ovisnosti o broju grupa naglo opada pa stagnira
  - ☐ D Najvjerojatnije središte grupe jest primjer koji je najviše udaljen od njemu najbližeg središta
4. (T) Algoritmi grupiranja k-sredina i k-medoida razlikuju se, između ostaloga, i po vremenskoj računalnoj složenosti. Naime, algoritam k-medoida računalno je složeniji od algoritma k-sredina. **Zašto je algoritam k-medoida računalno složeniji od algoritma k-sredina?**
  - ☐ A Za razliku od algoritma k-sredina koji se zasniva na euklidskoj udaljenosti, čiji je izračun računalno nezahtjevan, algoritam k-medoida koristi funkcije sličnosti čije računanje iziskuje mnogo računalnih operacija
  - ☐ B Budući da algoritam k-medoida ne koristi centroide, nego medoide, na kraju svake iteracije mora kombinatoričkom provjerom po primjerima pronaći medoide koje minimiziraju kriterijsku funkciju  $J$
  - ☐ C Za razliku od algoritma k-sredina, algoritam k-medoida je algoritam mekog grupiranja, što iziskuje provođenje dodatnih koraka unutar algoritma
  - ☐ D Kriterijska funkcija algoritma k-medoida jest mnogo složenija od one k-sredina, upravo zato što algoritam k-medoide koristi medoide, a ne centroide
5. (T) Algoritam K-medoida općenitiji je od algoritma K-sredina budući da se može koristiti za primjere koji nisu prikazani kao vektori. Međutim, razmotrite slučaj kada primjeri jesu prikazani

kao vektori, ali ih želimo grupirati na temelju mjere udaljenosti koja nije euklidska. **Koji bismo algoritam koristili u tom slučaju i zašto?**

- ☐ A Algoritam K-medoida, jer kriterijska funkcija algoritma K-sredina koristi euklidsku udaljenost
  - ☐ B Algoritam K-sredina, jer za vektorizirane primjere možemo izračunati centroide grupa
  - ☐ C Algoritam K-medoida, jer vektorizirani primjeri također mogu biti medoidi
  - ☐ D Algoritam K-sredina, jer koristi mjeru udaljenosti, dok algoritam K-medoida koristi mjeru sličnosti
6. (T) Algoritam K-medoida proširenje je algoritma K-sredina na neeuklidske ulazne prostore. Kod algoritma K-medoida kao funkciju različitosti  $\nu$  možemo u načelu koristiti bilo koju funkciju. Specifično, ako je ulazni prostor vektorski, možemo koristiti euklidsku udaljenost. **Što bi se dogodilo da kao funkciju različitosti koristimo euklidsku udaljenost?**
- ☐ A Algoritam bi bio davao grupe koje bi u prosjeku bile više izdužene nego one dobivene algoritmom K-sredina
  - ☐ B Algoritam K-medoida davao bi isto grupiranje kao i algoritam K-sredina, ali s većom vremenskom složenosti
  - ☐ C Algoritam bi uz veću prostornu složenost davao grupiranje u manji broj grupa nego algoritam K-sredina
  - ☐ D Algoritam bi primjere grupirao slično kao i algoritam K-sredina, pogotovo ako u središtima grupa postoje primjeri
7. (T) Algoritam K-sredina iterativno smanjuje vrijednost funkcije pogreške  $J$ . Neovisno o broju grupa  $K$ , algoritam uvijek konvergira (završava s izvođenjem). Međutim, konačna vrijednost funkcije pogreške  $J$  može se razlikovati između dva pokretanja algoritma s istim brojem grupa  $K$ . **Zašto?**
- ☐ A Zbog nelinearnosti granice između  $K$  grupa
  - ☐ B Zbog loše odabrane vrijednosti za broj grupa  $K$
  - ☐ C Zbog razlike u početnim središtima  $\mu_k$
  - ☐ D Zbog izotropnosti kovarijacijskih matrica  $\Sigma_k$

## 20 Grupiranje II

1. (T) Model miješane gustoće sa  $K$  komponenti vjerojatnosnu distribuciju neoznačenih podataka definira kao  $p(\mathbf{x}) = \sum_{k=1}^K \pi_k p(\mathbf{x}|\theta_k)$ . Što modelira uvjetna vjerojatnost  $p(\mathbf{x}|\theta_k)$ ?
- ☐ A Gustoću vjerojatnosti primjera  $\mathbf{x}$  unutar grupe  $k$
  - ☐ B Vjerojatnost da primjer  $\mathbf{x}$  pripada grupi  $k$
  - ☐ C Stupanj pripadnosti primjera  $\mathbf{x}$  grupi  $k$
  - ☐ D Gustoću vjerojatnosti grupe  $k$  za primjer  $\mathbf{x}$
2. (T) Za meko grupiranje možemo koristiti model miješane gustoće s latentnim varijablama. Kod tog modela, svaki primjer  $\mathbf{x}^{(i)}$  ima pridruženu latentnu varijablu  $\mathbf{z}^{(i)}$ . Obje ove varijable zapravo su slučajne varijable sa svojom pretpostavljenom distribucijom. **Koju distribuciju pretpostavljamo za latentnu varijablu  $\mathbf{z}^{(i)}$  i zašto?**
- ☐ A Beta-distribuciju koja opisuje s kojom vjerojatnošću primjer  $\mathbf{x}^{(i)}$  pripada svakoj grupi
  - ☐ B Kategoričku distribuciju koja opisuje kojoj grupi primjer  $\mathbf{x}^{(i)}$  zapravo pripada
  - ☐ C Bernoullijevu distribuciju koja opisuje s kojom vjerojatnošću primjer  $\mathbf{x}^{(i)}$  pripada grupi  $\mathbf{z}^{(i)}$
  - ☐ D Gaussovu distribuciju koja opisuje vjerojatnost da je grupa generirala primjer  $\mathbf{x}^{(i)}$

3. (T) Model Gaussove mješavine često treniramo algoritmom maksimizacije očekivanja. **Na što se odnosi pojam “očekivanje” u nazivu tog algoritma?**
- ☐ A Izglednost parametara modela izračunata uz fiksiranu pripadnosti svakog primjera njemu najbližoj grupi
  - ☐ B Vjerojatnost parametara modela s fiksnim dodjeljivanjem primjera grupama izračunata na temelju maksimizacije log-izglednosti
  - ☐ C Vjerojatnost skupa podataka pod modelom s fiksnim parametrima izračunata na temelju vjerojatnosti pripadanja primjera svakoj grupi
  - ☐ D Vjerojatnost pripadanja primjera svakoj grupi izračunata Bayesovim pravilom na temelju modela Gaussove mješavine
4. (T) Algoritam maksimizacije očekivanja (EM-algoritam) maksimizira očekivanje potpune log-izglednosti, što se pokazuje da dovodi i do maksimizacije nepotpune log-izglednosti. **Koja je razlika između potpune i nepotpune log-izglednosti, i zašto maksimiziramo očekivanje potpune log-izglednosti umjesto izravno log-izglednost?**
- ☐ A Potpuna log-izglednost je izglednost izračunata na svim primjerima iz neoznačenog skupa primjera, dok je nepotpuna log-izglednost izračunata samo za označene primjere koji se koriste za evaluaciju modela, a očekivanje računamo zato jer je postupak grupiranja stohastičan
  - ☐ B Potpuna log-izglednost je log-izglednost s neopaženim varijablama, a u slučaju GMM-a to su centriodi i kovarijacijske matrice komponenata, koje procjenjujemo metodom MLE, koja maksimizira očekivanje log-izglednosti
  - ☐ C Potpuna log-izglednost računa se za označene primjere a nepotpuna log-izglednost za neoznačene primjere, a u oba slučaja kod modela GMM računamo očekivanje log-izglednosti jer postupak za različite početne centriode može dati različite log-izglednosti
  - ☐ D Potpuna log-izglednost je log-izglednost modela GMM s latentnim varijablama, koje definiraju koji primjer pripada kojoj grupi, međutim kako to zapravo ne znamo, moramo računati s očekivanjem tih varijabli
5. (T) Za procjenu parametara modela GMM tipično se koristi algoritam maksimizacije očekivanja (EM-algoritam). To je iterativan optimizacijski algoritam. **Pod kojim uvjetima EM-algoritam (primijenjen na model GMM) konvergira, i kamo?**
- ☐ A Krenuvši od nekih početnih parametara, algoritam uvijek konvergira do parametara koji maksimiziraju očekivanje log-izglednosti, međutim to ne moraju biti parametri koji maksimiziraju vjerojatnost podataka
  - ☐ B Algoritam konvergira samo ako su primjeri u ulaznom prostoru sferični, ako su zavisnosti između značajki linearne, i ako nema multikolinearnosti, jer u protivnom zavisnosti nije moguće modelirati kovarijacijskom matricom
  - ☐ C Algoritam uvijek konvergira, i to do točke u prostoru parametara koja maksimizira log-izglednost parametara, no brzina konvergencije ovisi o tome kako su inicijalizirani parametri
  - ☐ D Algoritam uvijek konvergira, međutim globalni maksimum log-izglednosti parametara doseže samo ako je broj grupa postavljen na pravi broj grupa ili tako da je broj grupa jednak broju primjera
6. (T) Algoritam GMM, odnosno model Gaussove mješavine s algoritmom maksimizacije očekivanja kao optimizacijskim postupkom, počinje je algoritma k-sredina. **Uz koje uvjete algoritam GMM degenerira u algoritam k-sredina?**
- ☐ A Umjesto maksimizacije log-izglednosti, minimizira se negativna log-izglednost, a početna središta se odabiru algoritmom k-sredina
  - ☐ B Koeficijenti mješavine su jednaki za sve komponente Gaussove mješavine, a kovarijacijske matrice su dijagonalne
  - ☐ C Kovarijacijska matrica komponenti Gaussove mješavine je dijeljena i izotropna, a odgovornosti su zaokružene na cijeli broj
  - ☐ D Kovarijacijska matrica komponenti Gaussove mješavine je jedinična matrica, a maksimizira se negativna log-izglednost

7. (T) Za grupiranje primjera u  $K$  grupa koristimo model Gaussove mješavine (GMM) s dijeljenom kovarijacijskom matricom. Nakon grupiranja, odgovornosti zaokružujemo na cijeli broj, čime dobivamo tvrdo grupiranje. Iste podatke grupiramo algoritmom K-medoida (KM). **Uz koje parametre ovih algoritama očekujemo dobiti najsličnije rezultate grupiranja?**
- ☐ A GMM:  $\Sigma_k = \text{diag}(\sigma_1^2, \dots, \sigma_n^2)$  i  $\pi_k = 1/K$ ; KM: euklidska udaljenost
  - ☐ B GMM:  $\Sigma_k = \sigma^2 \mathbf{I}$ ; KM: euklidska udaljenost
  - ☐ C GMM: puna  $\Sigma_k$  i  $\pi_k = 1/K$ ; KM: Mahalanobisova udaljenost
  - ☐ D GMM:  $\Sigma_k = \sigma^2 \mathbf{I}$  i  $\pi_k = 1/K$ ; KM: Mahalanobisova udaljenost
8. (T) Broj grupa  $K$  hiperparametar je mnogih algoritama grupiranja, pa tako i algoritma GMM. Optimalan broj grupa može se odrediti na razne načine, a jedan od njih je Akaikeov kriterij. **Na kojem se principu temelji odabir broja grupa Akaikeovim kriterijem?**
- ☐ A Model s optimalnim brojem grupa je onaj koji minimizira log-izglednost nepotpunih podataka, a maksimizira log-izglednost potpunih podataka
  - ☐ B Optimalan broj grupa je onaj koji maksimizira očekivanje log-izglednost modela, uz pretpostavku izotropne kovarijacijske matrice
  - ☐ C Model s optimalnim brojem grupa je onaj koji podatke čini najvjerojatnijima, ali to čini sa što manje parametara
  - ☐ D Optimalan broj grupa je onaj kod kojeg, nakon daljnjeg povećanja broja grupa, vrijednost log-izglednosti stagnira ili blago raste
9. (T) Optimizaciju parametara modela Gaussove mješavine (GMM) ne provodimo u zatvorenoj formi. S druge strane, parametre Gaussovog Bayesovog klasifikatora, koji je sličan modelu GMM, optimiramo u zatvorenoj formi. **Zašto parametre GMM-a ne optimiramo u zatvorenoj formi, dok kod Gaussovog Bayesovog klasifikatora to radimo?**
- ☐ A Za razliku od Gaussovog Bayesovog klasifikator, GMM je nenadzirani algoritam, pa log-izglednost podataka nije definirana i nije moguća maksimizacija u zatvorenoj formi
  - ☐ B Kod GMM-a, pored koeficijenata mješavine i vektora sredina, trebamo procijeniti i kovarijacijske matrice, za što ne postoji procjenitelj u zatvorenoj formi
  - ☐ C Kod GMM-a ne znamo koji primjer pripada kojoj grupi, pa je gustoća primjera jednaka zbroju gustoći komponenti, za što ne postoji maksimizator u zatvorenoj formi
  - ☐ D Parametri oba modela mogu se optimirati u zatvorenoj formi, međutim kod modela GMM računalno je jednostavnije koristiti EM-algoritam
10. (T) Algoritam k-medoida proširenje je algoritma k-sredina. **Što algoritam k-medoida i algoritam hijerarhijskog grupiranja HAC imaju zajedničko, a po čemu se razlikuju?**
- ☐ A Oba algoritma mogu grupirati na temelju mjere udaljenosti koja nije euklidska, no algoritam k-medoida ima veću vremensku složenost od algoritma HAC
  - ☐ B Oba algoritma imaju vremenska složenost veću od linearne u broju primjera, no kod k-medoida primjer može pripada u više grupa istovremeno
  - ☐ C Oba algoritma izvode se onoliko iteracija koliko ima grupa, no algoritam k-medoida može raditi s općenitom mjerom sličnosti ili udaljenosti
  - ☐ D Oba algoritma mogu grupirati primjere koji nisu vektorizirani, no algoritam HAC daje hijerarhiju dok algoritam k-medoida daje particiju grupa
11. (T) Za procjenu parametara modela GMM (središta grupa  $\mu_k$ , kovarijacijske matrice  $\Sigma_k$  i koeficijenti mješavine  $\pi_k$ ) tipično se koristi algoritam maksimizacije očekivanja (EM-algoritam). Prije

pokretanja EM-algoritma, parametre je potrebno inicijalizirati. U praksi se za inicijalizaciju parametara  $\mu_k$  koristi algoritam K-sredina. **Koji je očekivani efekt uporabe algoritma K-sredina za inicijalizaciju EM-algoritma?**

- ☐ A Veća log-izglednost početnih i konačnih parametara te manje iteracija do konvergencije
  - ☐ B Manja log-izglednost početnih parametara, ali veća log-izglednost konačnih parametara
  - ☐ C Manja log-izglednost početnih i konačnih parametara te optimalan broj grupa  $K$
  - ☐ D Veća potpuna log-izglednost konačnih parametara i veće vrijednosti gustoća komponenti
12. (T) Prije primjene algoritma maksimizacije očekivanja na model GMM, model je potrebno proširiti latentnim varijablama  $\mathbf{z}^{(i)}$ , i to jednom takvom varijablom za svaki primjer  $\mathbf{x}^{(i)}$ . Neka je slučajna varijabla  $z_k^{(i)}$   $k$ -ta komponenta slučajnog vektora  $\mathbf{z}^{(i)}$ . **Koju informaciju kodira varijabla  $z_k^{(i)}$ ?**
- ☐ A Pripadnost primjera  $\mathbf{x}^{(i)}$  grupi  $k$
  - ☐ B Vjerojatnost da primjer  $\mathbf{x}^{(i)}$  pripada grupi  $k$
  - ☐ C Vjerojatnost primjera  $\mathbf{x}^{(i)}$ , ako je njegova grupa  $k$
  - ☐ D Odgovornost primjera  $\mathbf{x}^{(i)}$  za grupu  $k$

## 21 Vrednovanje modela

1. (T)  $F_1$ -mjera računa točnost klasifikatora kao harmonijsku sredinu preciznosti (P) i odziva (R). **Zašto se za  $F_1$ -mjeru koristi harmonijska, a ne aritmetička sredina?**
- ☐ A Jer su P i R obrnuto proporcionalni, ali definirani na istom intervalu
  - ☐ B Jer niti P niti R ne uzimaju u obzir broj stvarno negativnih primjera
  - ☐ C Jer P i R nisu definirani na istoj skali, ali njihove recipročne vrijednosti jesu
  - ☐ D Jer P nije definiran za trivijalan klasifikator koji sve primjere klasificira negativno
2. (T) Mjera točnosti nije prikladna za vrednovanje klasifikatora na skupovima podataka s neuravnoteženim brojem primjera po klasama. Jedna alternativa mjeri točnosti je  $F_1$ -mjera, međutim ni ta mjera nije uvijek prikladna. Pretpostavite da vrijednost  $F_1$ -mjere postavljamo na nulu u slučajevima kada je harmonijska sredina preciznosti i odziva nedefinirana. **U kojem slučaju  $F_1$ -mjera ne bi bila prikladna mjera za vrednovanje klasifikatora jer bi bila previše optimistična?**
- ☐ A Ako je većina primjera pozitivna i klasifikator sve primjere klasificira pozitivno
  - ☐ B Ako je većina primjera pozitivna, a klasifikator sve primjere klasificira negativno
  - ☐ C Ako je većina primjera negativna, a klasifikator sve primjere klasificira pozitivno
  - ☐ D Ako je većina primjera negativna i klasifikator sve primjere klasificira negativno
3. (T) Za vrednovanje višeklasnog klasifikatora često se koriste mjere  $F_1^\mu$  (mikro F1-mjera) i  $F_1^M$  (makro F1-mjera). **Koji je očekivani odnos između vrijednosti tih mjera, i zašto?**
- ☐ A  $F_1^M < F_1^\mu$ , jer kod makro F1-mjere računamo prosjek F1-mjere kroz sve klase, a klasifikator na manjim klasama više griješi
  - ☐ B  $F_1^M > F_1^\mu$ , jer kod mikro F1-mjere zbrajamo matrice zabune kroz sve klase, a primjera iz manjih klasa ima manje, pa manje doprinose pogrešci
  - ☐ C  $F_1^M > F_1^\mu$ , jer kod makro F1-mjere zbrajamo matrice zabune kroz sve klase, a primjera iz većih klasa ima više, pa više doprinose pogrešci
  - ☐ D  $F_1^M < F_1^\mu$ , jer kod mikro F1-mjere računamo prosjek F1-mjere kroz sve klase, a klasifikator na većima klasama manje griješi

4. (T) Za vrednovanje klasifikatora na manjim skupovima podataka može se koristiti metoda unakrsne provjere “izvoji jednog” (engl. *leave-one-out cross-validation*, LOOCV). Prednost te metode je što se procjena dobiva na temelju mnogo uzoraka. No, metoda ima i nekih nedostataka. **Što je nedostatak metode LOOCV?**
- ☐ A Procjena je pristrana jer se ispitni skupovi preklapaju u  $1/N$  primjera
  - ☐ B Varijanca procjene je visoka jer klasifikatori dijele  $(N - 2)/N$  primjera za učenje
  - ☐ C Procjena može biti pesimistična, jer ispitani model može biti suboptimalne složenosti
  - ☐ D Procjena je pristrana jer se svaki primjer u skupu za ispitivanje koristi  $N$  puta
5. (T) Procjena pogreške modela metodom unakrsne provjere omogućava nam da procijenimo prediktivnu moć modela, mjerenu kao točnost modela na neviđenom skupu primjera. Daljnja razrada te ideje je ugniježdjena višestruka unakrsna provjera, koja se u praksi vrlo često koristi. **Koja je motivacija za korištenje ugniježdjene višestruke unakrsne provjere, umjesto obične unakrsne provjere?**
- ☐ A Omogućava nam da procijenimo prediktivnu moć modela optimalne složenosti te maksimalno iskoristimo raspoložive podatke za učenje i ispitivanje
  - ☐ B Provodi optimizaciju hiperparametra modela na uniji skupa za provjeru i skupa za testiranje, čime postiže bolju točnost modela jer više primjera ostaje za treniranje
  - ☐ C Razdvaja skup za učenje od skupa za ispitivanje te time osigurava da doista mjerimo prediktivnu moć modela, odnosno ispitnu pogrešku, a ne pogrešku učenja
  - ☐ D Omogućava nam da odredimo točnost modela s klasifikacijskim pragom, na način da u obzir uzimamo preciznost i odziv za različite vrijednosti klasifikacijskog praga
6. (T) Ugniježdjena  $k$ -struka unakrsna provjera često se koristi za procjenu točnosti modela. **Što je prednost ugniježdjene  $k$ -struke unakrsne provjere u odnosu na običnu unakrsnu provjeru?**
- ☐ A Procjenjujemo pogrešku generalizacije, a ne pogrešku učenja
  - ☐ B Model ispitujemo na cijelom raspoloživom skupu primjera
  - ☐ C Procjenjujemo ispitnu pogrešku modela s najmanjom pogreškom učenja
  - ☐ D Procjenjujemo očekivanu ispitnu pogrešku modela optimalne složenosti
7. (T) Kod vrednovanja algoritma strojnog učenja važno je da vrednujemo model optimalne složenosti. Zbog toga označeni skup podataka dijelimo na tri disjunktna podskupa: skup za učenje, skup za provjeru i ispitni skup. Za odabir modela (tj. optimizaciju hiperparametara) koristimo skup za učenje i skup za provjeru, dok točnost optimalnog modela ispitujemo na ispitnome skupu. Zamislite da smo zaboravili upotrijebiti skup za provjeru: umjesto toga hiperparametre modela optimirali smo na skupu za učenje, a model smo potom ispitali na ispitnom skupu. **U odnosu na to da smo pravilno proveli vrednovanje algoritma, što možemo očekivati u ovom slučaju?**
- ☐ A Model će biti prenaučen i imat će manju točnost na ispitnome skupu
  - ☐ B Model će biti podnaučen, ali imat će veću točnost na ispitnome skupu
  - ☐ C Model će biti prenaučen, ali imat će veću točnost na ispitnome skupu
  - ☐ D Model će biti optimalne složenosti i imat će veću točnost na ispitnome skupu

## Rješenja

	1	2	3	4	5	6	7	8	9	10	11	12
2. Osnovni koncepti	B	B	A	A	D	C	B	D	A	D		
3. Linearna regresija	C	C	B	C	B	A	B	D	C			
4. Linearna regresija II	A	B	B	C	B	A	A	B				
5. Linearni diskriminativni modeli	C	A	D	B	B	D	D					
6. Logistička regresija	B	B	B	C	D							
7. Logistička regresija II	B	C	A	B	C	C	D	B	B	C		
8. Stroj potpornih vektora	A	D	C	C	B	D						
9. Stroj potpornih vektora II	D	D	A	B	A	B	D					
10. Jezgrene metode	C	A	D	B	D	D	D	C				
11. Neparametarske metode	C	C	B	C								
14. Procjena parametara II	C	C	D	D	A	B	B	A				
15. Bayesov klasifikator	B	D	A	B	C	D						
16. Bayesov klasifikator II	A	C	D	C								
17. Probabilistički grafički modeli	D	B	C	B	A	D						
18. Probabilistički grafički modeli II	A	D	C	A	B	D	C	A				
19. Grupiranje	C	A	D	B	A	D	C					
20. Grupiranje II	A	B	C	D	A	C	C	C	C	D	A	A
21. Vrednovanje modela	C	A	A	B	A	D	A					