

# Strojno učenje - ishodi učenja 2019./20.

Materijal za pripremu međuispita i završnog ispita

Hvala FER2.netovcima koji su sudjelovali u izradi ovog materijala!

Poglavlja - dostupno i preko outlinea sa strane

1. [Uvod u strojno učenje](#)
2. [Osnovni koncepti](#)
3. [Regresija](#)
4. [Regresija II](#)
5. [Linearni diskriminativni modeli](#)
6. [Logistička regresija](#)
7. [Logistička regresija II](#)
8. [Stroj potpornih vektora](#)
9. [Stroj potpornih vektora II](#)
10. [Jezgrene metode](#)
11. [Neparametarske metode](#)
12. [Ansambl](#)
13. [Procjena parametara](#)
14. [Procjena parametara II](#)
15. [Bayesov klasifikator](#)
16. [Bayesov klasifikator II](#)
17. [Probabilistički grafički modeli I](#)
18. [Probabilistički grafički modeli II](#)
19. [Grupiranje I](#)
20. [Grupiranje II](#)
21. [Vrednovanje modela I](#)
22. [Vrednovanje modela II](#)
23. [Odabir značajki](#)

Ishodi označeni zvjezdicom (\*) nisu obavezni i ne provjeravaju se, ali mogu pomoći boljem razumijevanju gradiva.

TODO - do meduispita je sve lijepo formatirano, a za zavrsni je grozno

\*\*\*PRIVREMENO\*\*\*

Ažurirani ishodi učenja -> 19.1.2020: [Strojno učenje: Ishodi učenja predmeta](#)

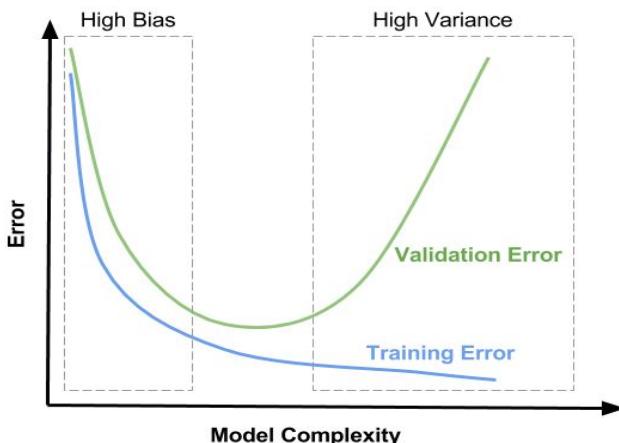
# 1 Uvod u strojno učenje

1. Definirati strojno učenje
  - a. Strojno učenje grana je umjetne inteligencije koja se bavi oblikovanjem algoritama koji svoju učinkovitost poboljšavaju na temelju empirijskih podataka.
  - b. programiranje računala na način da optimiziraju neki kriterij uspješnosti temeljem podatkovnih primjera ili prethodnog iskustva (1.prez)
2. Navesti tipične slučajeve primjene strojnog učenja
  - a. Raspoznavanje uzorka
  - b. Dubinska analiza podataka
  - c. Robotika
  - d. Računalni vid...
3. Razlikovati između nadziranog i nenadziranog strojnog učenja te dati primjere
  - a. **Nadzirano učenje**
    - i. podaci su parovi (ulaz,izlaz) =  $(x,y)$
    - ii. treba pronaći  $y = f(x)$
  - b. **Nenadzirano učenje**
    - i. dani su podaci bez ciljne vrijednosti
    - ii. treba naći pravilnost u podacima
4. Razlikovati između klasifikacije i regresije te dati primjere
  - a. **Klasifikacija** - želimo dobiti jednu od n klasa (primjer: filtriranje spama),  $Y = \text{diskretni skup (npr. } \{-1, +1\}, \{0, \dots, K - 1\} \text{)}$
  - b. **Regresija** - želimo dobiti izlaz koji je realan broj (primjer: predviđanje kretanja dionica),  $Y = \text{skup realnih brojeva (npr. } [0, 1], \text{ ili cijeli R)}$

# 2 Osnovni koncepti

1. Definirati pojmove hipoteza, model, funkcija gubitka/pogreške te dati primjere
  - a. **Hipoteza:** funkcija koja mapira ulazni prostor  $X$  u prostor oznaka  $Y$  (npr. pravac, ravnina ili hiperbola, ...)
  - b. **Model:** skup svih mogućih hipoteza (npr. svi pravci kroz ishodište ili ravnine s fiksiranom normalom)
  - c. **Funkcija gubitka:** funkcija koja nam govori koliko model loše predviđa podatke iz skupa za učenje  $D$ . Što je model lošije naučen, funkcija gubitka će imati veću vrijednost. Primjer funkcije gubitka je MSE (mean squared error).
2. Definirati pojam i vrste induktivne pristranosti te dati primjere
  - a. Induktivna pristranost je skup pretpostavki koje, uz skup za učenje, odabiremo kako bismo omogućili našem modelu da može generalizirati.  
Formalno: *Neka je  $\mathcal{L}$  algoritam za učenje, neka je  $h_{\mathcal{L}}$  hipoteza inducirana pomoću  $\mathcal{L}$  na skupu primjera  $D$  i neka je  $h_{\mathcal{L}}(x)$  klasifikacija primjera  $x \in D$  temeljem te hipoteze. Induktivna pristranost od  $\mathcal{L}$  je bilo koji skup minimalnih pretpostavki  $B$  takvih da:*
$$\forall D, \forall x \in D, ((B \wedge D \wedge x) \vdash h_{\mathcal{L}}(x))$$
  - b. Vrste induktivne pristranosti su
    - i. pristranosti **ograničavanjem**/jezika (npr. pravci vs polinomi)
    - ii. pristranost **preferencijom**/pretraživanja (neke hipoteze su bolje od drugih)
3. Objasniti komponente algoritma učenja i povezati ih s induktivnom pristranošću
  - a. **Model**
    - i. skup hipoteza (koje su definirane do na parametre), indeksiran parametrima
    - ii. Odabir modela je vrsta pristranosti **ograničavanjem**.
  - b. **Funkcija gubitka**
    - i. mjera koliko loše hipoteza mapira točke iz skupa za učenje u pripadajuće vrijednosti iz skupa oznaka
    - ii. Različite funkcije gubitka različito rangiraju hipoteze pa je to vrsta pristranosti **preferencijom**.
  - c. **Optimizacijski postupak**
    - i. on bira konkretnu hipotezu (ili parametre) iz modela koja minimizira empirijsku pogrešku modela nad ulaznim podacima
    - ii. Također pristranost **preferencijom**.
4. Definirati prostor inaćica i njegovu vezu s induktivnom pristranošću algoritma
  - a. **Prostor inaćica** je skup svih onih hipoteza koje ispravno klasificiraju primjere iz  $D$
  - b. Primjer: linearne odvojivi skupovi podataka mogu biti odvojeni sa beskonačno mnogo pravaca. S obzirom da nam treba samo jedna hipoteza, moramo uvesti induktivnu pristranost **odabira preferencijom** (neke od tih najboljih hipoteza su nam ipak "draže", na primjer maksimalna margina)
5. Odrediti predikciju/prostор inaćica zadanog modela na skupu primjera
  - a. Stvarni život. (Primjer u službenoj skripti na str. 7)
6. Objasniti utjecaj šuma na složenost modela i navesti moguće uzroke šuma
  - a. Što je više šuma u podacima na kojima učimo model, to će nam biti teže naučiti, i model će biti složeniji. No, složeniji modeli su skloni prenaučiti se na šum u podacima i tako slabije generaliziraju.
  - b. Šum može doći iz više izvora:
    - i. nepreciznost pri mjerenuju
    - ii. pogreške u označavanju

- iii. postojanje skrivenih značajki
  - iv. nejasne granice između klasa
7. Objasniti odabir modela, prenaučenost i podnaučenost
- Odabir modela:** postupak kojim tražimo optimalni skup hipoteza. Taj odabir se radi unakrsnom provjerom, odabirom modela kod kojega je empirijska pogreška (funkcija pogreške) na skupu za testiranje najmanja.
  - Prenaučenost:** zbog šuma u podacima, model se toliko nauči na šum da više ne generalizira. Primjer, student koji prolazi stare ispite i nauči to dobro, ali ne zna razumjeti gradivo. (također, <https://pbs.twimg.com/media/DdkUUTMV4AAsohU.jpg>) Mala pogreška na skupu za učenje, velika pogreška na skupu za provjeru.
  - Podnaučenost:** model ne opisuje dobro podatke. Ili nije dovoljne složenosti, ili nije dovoljno dugo učen, ili nešto treće. Velike pogreške i na skupu za učenje, i na skupu za testiranje. Slabo generalizira.
8. Razlikovati između parametara i hiperparametara modela
- Parametre** model uči
    - npr. težine i slobodni članovi linearne regresije
  - Hiperparametre** ne uči nego su mu dani kao takvi (određujemo ih mi)
    - npr. stopa učenja  $\eta$ , ili broj susjeda  $K$  u KNN, ili stupanj polinoma  $p$ .
9. (+ 10. u jednom) Objasniti unakrsnu provjeru i graf pogrešaka u ovisnosti o složenosti modela. . Interpretirati i usporediti složenost modela na temelju pogreške učenja/ispitivanja
- Unakrsna provjera (eng. cross validation) je ideja da ne treniramo naš model na čitavom skupu podataka koji posjedujemo, već da izdvojimo poseban dio tog skupa (kojeg nikad nećemo dati modelu da "vidi") za provjeru točnosti. Prije su se skupovi dijelili na otprilike 70% dataset za treniranje i 30% dataset za provjeru, no u praksi, zbog naprosto ogromne količine podataka koja je sada dostupna, udio validation seta pada na 5% ili čak 1% ukupnog dataseta.
  - Grafovi train i test pogrešaka u ovisnosti o složenosti modela:



Isprva je model odveć jednostavan (na engl. "highly biased" model) i ne generalizira dobro. Kažemo da je model **podnaučen** (engl. *underfitted*). To se odlikuje tako da su i pogreška učenja i pogreška provjere dosta visoke. Kako se složenost povećava, model sve bolje i bolje može opisati podatke i obje greške se smanjuju. U jednoj točki složenosti pogreška za provjeru je najmanja, i to je **peak performance** modela. Kasnijem povećanjem složenosti, pogreška učenja nastavlja padati (ona uvijek pada) jer se model prilagođava kako na naše podatke, tako i na njihov šum, koji nije isti u datasetu za provjeru. Zato pogreška provjere sada raste sa složenosti modela i model je **prenaučen** (engl. *overfitted*). To se na eng zove "high variance" jer male promjene u podacima će imati velike promjene u modelu (zbog šuma).

# 3 Regresija

1. Definirati model linearne regresije i jednostavne regresije
  - a.  $h(\mathbf{x}; \mathbf{w}, w_0) = \mathbf{w}^T \mathbf{x} + w_0$
  - b. Jednostavna regresija je kad su vektori  $\mathbf{x}$  i  $w$  jednodimenzionalni (dakle, skalari).
2. Navesti vrste regresija i dati primjere
  - a. Prema **ulaznim** varijablama
    - i. **jednostavna** - vrijednost funkcije ovisi o samo jednoj varijabli
    - ii. **višestruka** - vrijednost funkcije ovisi o više ulaznih varijabli
  - b. Prema **izlaznim** varijablama
    - i. **univariatna** - istovremeno predviđamo samo jednu varijablu
    - ii. **multivariatna** - istovremeno predviđamo više varijabli odjednom
3. Objasniti postupak najmanjih kvadrata i motivaciju za njegovu primjenu
  - a. Želimo da nam izlazi modela (na datasetu) budu što bliže ciljnog vektoru. "Što bliže" definiramo kao prosječno kvadratno odstupanje komponenti, odnosno, oduzmemmo dva vektora, kvadriramo svaku komponentu i sve pozbrajamo (opcionalno podijelimo sa  $N$ , nebitno). Dalje ide raspisivanje toga u neki oblik s kojim je lakše baratati, zatim dobiveni izraz deriviramo po težinama i izjednačimo s 0 jer tražimo ekstrem, i koristeći nekoliko pravila koja nas nitko nije naučio na mat2 dobijemo nešto što se zove lijevi pseudoinverz, koji se tada koristi za odabir hipoteze formulom u zatvorenom obliku.
4. Izvesti rješenje najmanjih kvadrata u matričnom obliku
  - Funkcija pogreške u matričnom obliku:

$$\begin{aligned} E(\mathbf{w} | \mathcal{D}) &= \frac{1}{2} (\mathbf{Xw} - \mathbf{y})^T (\mathbf{Xw} - \mathbf{y}) \\ &= \frac{1}{2} (\mathbf{w}^T \mathbf{X}^T \mathbf{Xw} - \mathbf{w}^T \mathbf{X}^T \mathbf{y} - \mathbf{y}^T \mathbf{Xw} + \mathbf{y}^T \mathbf{y}) \\ &= \frac{1}{2} (\mathbf{w}^T \mathbf{X}^T \mathbf{Xw} - 2\mathbf{y}^T \mathbf{Xw} + \mathbf{y}^T \mathbf{y}) \end{aligned}$$

uz  $(A^T)^T = A$  i  $(AB)^T = B^T A^T$

- Minimizacija:

$$\begin{aligned} \nabla_{\mathbf{w}} E &= \frac{1}{2} \left( \mathbf{w}^T (\mathbf{X}^T \mathbf{X} + (\mathbf{X}^T \mathbf{X})^T) - 2\mathbf{y}^T \mathbf{X} \right) = \mathbf{w}^T \mathbf{X}^T \mathbf{X} - \mathbf{y}^T \mathbf{X} = \mathbf{0} \\ \mathbf{w}^T &= \mathbf{y}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \\ \mathbf{w} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \mathbf{X}^+ \mathbf{y} \end{aligned}$$

uz  $\frac{d}{dx} Ax = A$  i  $\frac{d}{dx} x^T Ax = x^T (A + A^T)$

5. Objasniti probabilističku interpretaciju linearne regresije i izvesti kvadratni gubitak
  - a. Premda je odabir kvadratnog odstupanja bio poprilično proizvoljan, ispada da on prirodno slijedi iz probabilističke interpretacije. Ideja je da u naš model uvedemo šum. Taj šum modeliramo normalnom (Gaussovom) distribucijom s očekivanjem 0 i nekom nebitnom varijancom (disperzijom) koju cemo samo označiti sa  $\sigma^2$ . Tako, svaku točku u datasetu možemo prikazati kao zbroj vrijednosti hipoteze u toj točki i šuma, odnosno  $N(h(x), \sigma^2)$ . Zatim možemo za svaku točku izračunati vjerojatnost da se pojavila njen oznaka. Pod pretpostavkom da su sve točke u datasetu nezavisne, ukupna vjerojatnost je umnožak svih vjerojatnosti točaka. Tu ukupnu vjerojatnost onda maksimiziramo. Nakon malo algebarske manipulacije dođe se do rješenja kako je najveća vjerojatnost onda kada je najmanje prosječno kvadratno odstupanje između vrijednosti točke i vrijednosti hipoteze u toj točki.  
b. Za više detalja pogledati [ovdje, strana 3](#) ili na [videu](#).
6. Objasniti vezu između log-izglednosti oznaka pod modelom i pogreške modela
  - a. Što je pogreška manja, ukupna vjerojatnost svih točaka (**izglednost**) u datasetu je veća. Za one parametre za koje MSE postiže svoj minimum, s tim parametrima će ukupna vjerojatnost imati svoj maksimum. Drugim riječima, možemo pisati  $E \propto -\log(p)$
7. Analizirati komponente i induktivnu pristranost algoritma linearne regresije
  - a. **Model:** linearna kombinacija značajki. To je velika induktivna pristranost ograničavanjem jer je u praksi malo vjerojatno da podaci zaista linearno ovise o značajkama.
  - b. **Funkcija gubitka:** MSE
  - c. **Optimizacijski postupak:** formula u zatvorenom obliku koja nam izravno daje najbolje parametre. ( $w^* = X^+ \hat{y}$ )

# 4 Regresija II

1. Objasniti bazne funkcije, funkciju preslikavanja i motivaciju za njenu primjenu
  - a. Bazne funkcije su nelinearne funkcije ulaznih varijabli.

$$\{\phi_0, \phi_1, \phi_2, \dots, \phi_m\}, \quad \phi_j : \mathbb{R}^n \rightarrow \mathbb{R}, \quad \phi_0(\mathbf{x}) = 1$$

- b. Funkcija preslikavanja preslikava u prostor značajki.

$$\begin{aligned}\phi : \mathbb{R}^n &\rightarrow \mathbb{R}^{m+1} : \\ \phi(\mathbf{x}) &= (\phi_0(\mathbf{x}), \dots, \phi_m(\mathbf{x}))\end{aligned}$$

- i. Iz jednog vektora konstruira nove vektore. Ti novi vektori su značajke koje (možda) bolje opisuju naš skup podataka. Preslikava primjere iz  $n$ -dim u  $(m+1)$ -dim prostor. Tipično  $m > n$ : veće šanse da će u višoj dimenziji biti linearno odvojivo.
- ii. Preslikavanje u prostor značajki - umjesto da mijenjamo model, mijenjamo podatke.

2. Povezati složenost modela s funkcijom preslikavanja

- a. Složenost modela ovisi o funkciji preslikavanja (proporcionalno):
  - i. npr. polinomijalno preslikavanje  $\rightarrow$  model je istog stupnja kao i preslikavanja.
  - ii. Nelinearan model je složeniji od linearног  $\rightarrow$  sklonost prenaučenosti.

3. Objasniti regularizaciju te objasniti prednosti i nedostatke L1- i L2-regularizacije

- a. Regularizacija ograničava rast parametara (kažnjava velike težine) pri učenju (**sprječava prenaučenost**)
  - i. Što je model više prenaučen, to su veće magnitude težina
    1. Kod regularizacija kažnjavaju se hipoteze s visokim vrijednostima parametara.
    2. Cilj je što više težina pritegnuti na 0 čime se dobivaju **rijetki** modeli.
  - ii. Kompromis između točnosti i jednostavnosti modela pri učenju modela.
- b. L1 - regularizacija
  - i. + daje rijetke modele
  - ii. - nema rješenje u zatvorenoj formi
- c. L2 - regularizacija
  - i. + ima rješenje u zatvorenoj formi
  - ii. - kažnjava težine proporcionalno kvadratu njihovog iznosa (velike  $w$  više, manje  $w$  manje), ne daje rijetke modele

4. Objasniti zašto L1-regularizacija rezultira rijetkim modelima, a L2-regularizacija ne.

- a. L2 ne daje rijetke modele jer kažnjava težine proporcionalno kvadratu njihovog iznosa (velike težine više kažnjava, a manje težine manje) pa će teško parametri biti pritegnuti baš na 0 što znači da ne rezultira rijetkim modelima.

- L2-norma ( $p = 2$ ):  $\|\mathbf{w}\|_2 = \sqrt{\sum_{j=1}^m w_j^2} = \sqrt{\mathbf{w}^T \mathbf{w}}$
- L1-norma ( $p = 1$ ):  $\|\mathbf{w}\|_1 = \sum_{j=1}^m |w_j|$
- L0-norma ( $p = 0$ ):  $\|\mathbf{w}\|_0 = \sum_{j=1}^m \mathbf{1}\{w_j \neq 0\}$

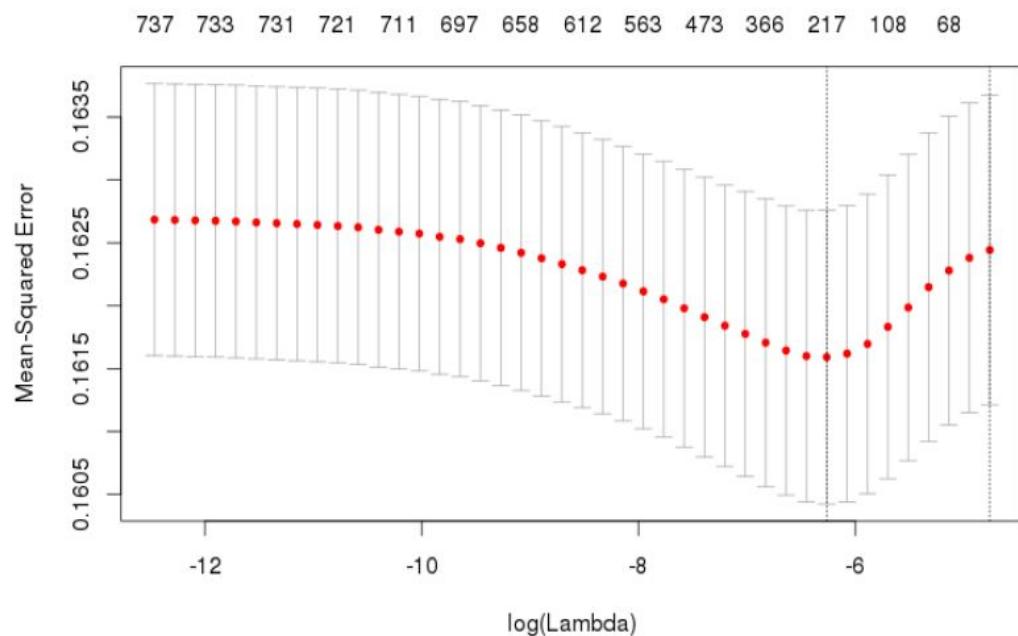
5. Izvesti rješenje L2-regulariziranih najmanjih kvadrata u matričnom obliku

$$\begin{aligned}
 E_R(\mathbf{w}|\mathcal{D}) &= \frac{1}{2}(\Phi\mathbf{w} - \mathbf{y})^T(\Phi\mathbf{w} - \mathbf{y}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \\
 &= \frac{1}{2}(\mathbf{w}^T\Phi^T\Phi\mathbf{w} - 2\mathbf{y}^T\Phi\mathbf{w} + \mathbf{y}^T\mathbf{y} + \lambda\mathbf{w}^T\mathbf{w}) \\
 \nabla_{\mathbf{w}}E_R &= \Phi^T\Phi\mathbf{w} - \Phi^T\mathbf{y} + \lambda\mathbf{w} \\
 &= (\Phi^T\Phi + \lambda\mathbf{I})\mathbf{w} - \Phi^T\mathbf{y} = 0 \\
 \mathbf{w} &= (\Phi^T\Phi + \lambda\mathbf{I})^{-1}\Phi^T\mathbf{y}
 \end{aligned}$$

gdje  $\lambda\mathbf{I} = \text{diag}(0, \lambda, \dots, \lambda)$  (težinu  $w_0$  ne regulariziramo)

6. Povezati regularizaciju, pogrešku učenja/ispitivanja i složenost modela

- a. **Regularizacija** faktorom lambda ( $\lambda$ ) **smanjuje složenost modela** što znači da ga je teže prenaučiti. Manje dopuštamo modelu da se prilagodi šumu na skupu za učenje tako što se manjim značajkama smanji težina čime se postigne i bolja generalizacija na ispitnom skupu.
- b. pogreška **učenja**
  - i. **raste s većim** lambda - lambda povećava generalizaciju
  - ii. **smanjuje se sa složenosti** modela - složenost smanjuje generalizaciju
- c. pogreška **ispitivanja**
  - i. do neke vrijednosti lambde pogreška **pada, ali** u nekom trenutku ponovno **počne rasti**
    - 1. jer prevelika lambda previše priguši model pa onda model **previše** generalizira i više ništa dobro ne predviđa (underfit / podnaučenost)
  - ii. **povećava se sa složenosti** modela zbog smanjenja generalizacije (overfit na train)



Isprva, kad ne regulariziramo, greška na skupu za provjeru je velika jer se model prenaučio na skup za učenje i ne generalizira dobro. Što više povećavamo regularizacijski parametar, greška na skupu za provjeru pada jer model sve vise i vise generalizira. U jednom trenutku greska padne na minimum i to je **peak performance** modela. Daljnje povecanje parametra regularizacije previše penalizira model tijekom učenja te on gubi moć da uopće nauči išta o našem datasetu.

7. Povezati rang matrice dizajna, multikolinearnost značajki i L2-regularizaciju
- Rang** matrice određuje broj linearne nezavisnih stupca ili redaka (uzima se manja vrijednost od tih dviju)
    - Matrica **punog ranga** je matrica kojoj su svi stupci i redci linearne **nezavisne**
  - Multikolinearnost**
    - dvije ili više značajki matrice (stupci) su linearne **zavisne**
    - Multikolinearnost se može smanjiti L2-regularizacijom na način da se u matricu doda lambda na dijagonalu

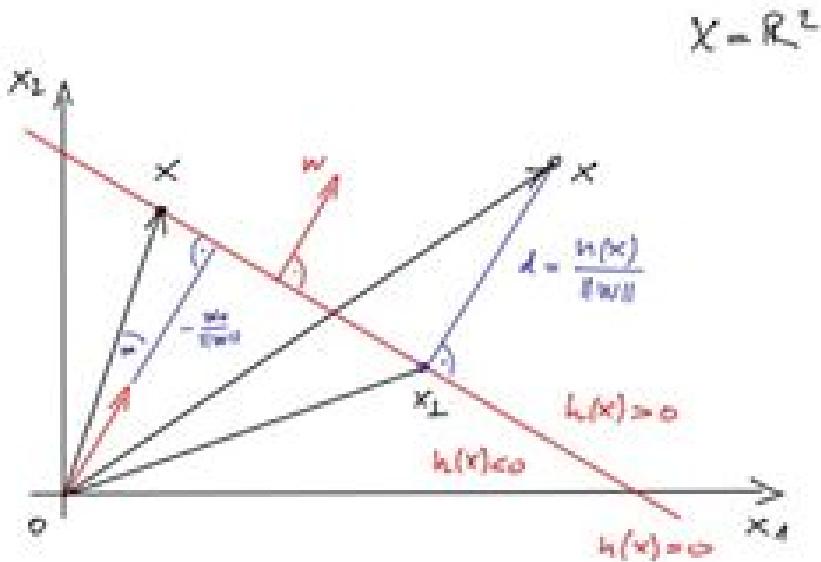
$$\mathbf{w} = (\Phi^T \Phi + \lambda \mathbf{I})^{-1} \Phi^T \mathbf{y}$$

1. to nazivamo **rekondicioniranjem matrice dizajna**.

8. (\*) Objasniti kondiciju matrice dizajna i vezu s regularizacijom i multikolinearnošću.
- Multikolinearnost
    - 2 ili više varijabli su visoko korelirane, to daje numerički nestabilno rješenje odnosno prenaučenost.
  - Kondicijski broj matrice
    - iskazuje nestabilnost rješenja; velik broj -> blizu da bude singularna matrica -> prenaučen model.
  - Regularizacija smanjuje multikolinearnost
    - u formuli  $\lambda I$  - učini ih linearne nezavisne odnosno rekondicionira matricu dizajna.
9. Izračunati matricu dizajna za zadane primjere i zadanu funkciju preslikavanja.  
g

# 5 Linearni diskriminativni modeli

1. Definirati pojам linearog diskriminativnog modela.
  - a. Linearni diskriminativni model je model za klasifikaciju kojemu je granica neka hiperravnina (u 3D je ravna, u 2D je pravac)
  - b. Matematički,  $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$ .
2. Objasniti geometriju linearog modela i izvesti udaljenost primjera od hiperravnine
  - a. Granica modela je  $\mathbf{w}^T \mathbf{x} + w_0 = 0$ . Ako nađemo dva primjera za koje hipoteza vraća istu vrijednost, matematičkim manipuliranjem možemo izvesti da je skalarni umnožak između težina modela i razlike ta dva vektora jednak vektorskoj nuli - okomiti su. S obzirom da je ta razlika primjera novi vektor koji je paralelan sa granicom (jer primjeri imaju istu vrijednost), to znači da je vektor težina okomit i na granicu, odnosno, taj vektor težina čini normalu na separacijsku hiperravninu. Udaljenost primjera od hiperravnine dobijemo tako da vrijednost hipoteze podijelimo sa duljinom vektora težina.



b.

$$\begin{aligned}
 \mathbf{x} &= \mathbf{x}_\perp + d \frac{\mathbf{w}}{\|\mathbf{w}\|} \\
 h(\mathbf{x}) &= \underbrace{\mathbf{w}^T \mathbf{x} + w_0}_{h(\mathbf{x})} = \underbrace{\mathbf{w}^T \mathbf{x}_\perp + w_0}_{=h(\mathbf{x}_\perp)=0} + d \frac{\mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|} \\
 h(\mathbf{x}) &= d \|\mathbf{w}\| \quad \Rightarrow d = \frac{h(\mathbf{x})}{\|\mathbf{w}\|}
 \end{aligned}$$

c.

3. Objasniti OVO/OVR, definirati pripadne modele i objasniti prednosti i nedostatke.
- Linearni diskriminativni modeli su binarni modeli, oni razlikuju samo dvije klase. Ako imamo više od dvije klase u datasetu, a želimo koristiti binarni klasifikator, moramo primijeniti trikove. Dva najčešća takva su **OVR** (one-vs-rest) i **OVO** (one-vs-one). OVO klasifikator uzima sve parove klasa i razlučuje između njih. Dobra strana je što radi super (gleda predznak kamo klasificirati pojedini primjer te "skuplja glasove" pripadnosti pojedinoj klasi,  $h(x)=\text{argmax}(\text{sum}(\text{sgn}(h_{ij}(x))))$ ), loša je što ima puno modela ( $n$  povrh 2, odnosno  $n(n-1)/2$ ). Drugi trik je OVR, a to je da treniramo  $n$  klasifikatora, svaki za svoju klasu kao pozitivnu i sve ostale kao negativnu. Dobra strana je malo modela, a loša to što potencira nejednakost u brojnostima podataka (dovodi do uneravnoteženosti klasa). Linearni modeli općenito imaju dosta problema ako je velika razlika između broja podataka dvaju klasa. Za OVR,  $h(x)=\text{argmax}(h_j(x))$ , granica između klasa je simetrala kutova  $h_j$  pravaca.
4. Objasniti klasifikaciju regresijom i objasniti nedostatke
- Klasifikacija regresijom je još jedan trik. Možemo podacima jedne klase pridijeliti oznaku 0, a drugoj klasi 1 i onda provući linearu regresiju kroz te pravce. Granica će nam biti  $h(x;w) = 0.5$  - ako  $h(x;w)$  vraća veću vrijednosti, pridjeti ćemo jedinicu, ako vraća manju, pridjeti ćemo 0. Ovo zvuči dlački ali ima mali milijun problema, npr izlaz nam ne daje nikakvu informaciju o vjerojatnosti pripadanja primjera klasu kojoj je dodjeljen, te užasna osjetljivost na stršeće vrijednosti (outliere) i na razliku u broju primjera za klase. Ako je primjer jako udaljen od granice klasa kako se kažnjava.
5. Postaviti optimizacijski problem klasifikacije regresijom za jedno/višeklasni problem

## 4 Klasifikacija regresijom

- Funkcija pogreške:

$$E(\mathbf{w}|\mathcal{D}) = \frac{1}{2} \sum_{i=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(i)}) - y^{(i)})^2 = \frac{1}{2} (\Phi \mathbf{w} - \mathbf{y})^\top (\Phi \mathbf{w} - \mathbf{y})$$

- Minimizator:

$$\mathbf{w}^* = (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y} = \Phi^+ \mathbf{y}$$

- Ideja: hipoteza koja predviđa  $y = 1$  i  $y = 0$  za primjere prve odnosno druge klase
- Model:  $h(\mathbf{x}; \mathbf{w}) = \mathbf{1}\{\mathbf{w}^\top \phi(\mathbf{x}) \geq 0.5\}$

6. Objasniti i skicirati funkciju gubitka perceptronu
- Gubitak perceptronu
    - ako je primjer dobro klasificiran, nemoj ništa kažnjavati
    - ako nije dobro klasificiran, kazni proporcionalno pogrešnoj klasifikaciji.  
 $L(y, h(\mathbf{x})) = \max(0, -y \mathbf{w}^\top \Phi(\mathbf{x}))$ .
    - Note - ovdje se pretpostavlja da klase više nisu 0,1 već -1,1

7. Definirati i skicirati funkciju pogreške perceptronu i usporediti je s pogreškom 0-1
- Funkcija pogreške je samo suma funkcija gubitka na datasetu

- b. pogreška 0-1 ima mnogih ravnih dijelova koji onemogućavaju primjenu gradijentnog spusta
8. Izvesti algoritam perceptronu
- S obzirom da funkcija gubitka nema lijepa svojstva za minimizator u zatvorenoj formi, takav ne postoji, pa primjenjujemo iterativni postupak nazvan Widrow-Hoffovo pravilo, koje kaže da idemo primjer po primjer, ako za neki primjer model pogrešno klasificira, ažuriraj mu težine nekim djeličem pogreške ( $w += \eta\Phi(x)y$ ), ako ne nemoj ništa.
  - $dL/dw = -\Phi(x)y$
- ```

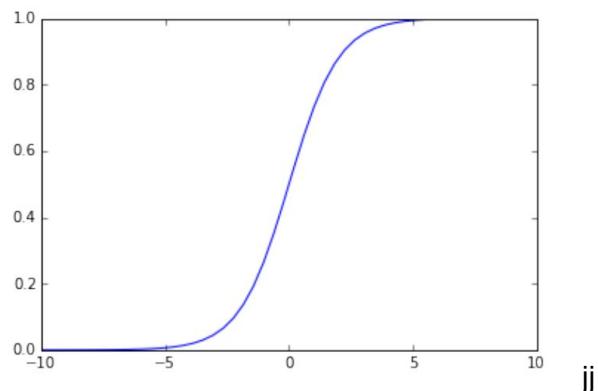
1: inicijaliziraj  $w \leftarrow (0, \dots, 0)$ 
2: ponavljam do konvergencije
3:   za  $i = 1, \dots, N$ 
4:     ako  $f(w^T \phi(x^{(i)})) \neq y^{(i)}$  onda  $w \leftarrow w + \eta \phi(x^{(i)}) y^{(i)}$ 

```
9. Navesti prednosti i nedostatke algoritma perceptronu
- Dobra stvar** je što algoritam perceptronu **ne kažnjava točno klasificirane primjere** te to da će uvijek naći hipotezu za linearno odvojive razrede
  - Loša stvar** je da algoritam perceptronu **neće konvergirati za linearno neodvojive razrede** (Rosenblatt je 1962. to dokazao), te rezultat ovisi o početnim težinama
    - Također, izlaz još uvijek ne daje nikakvu informaciju o tome koja je vjerojatnost pripadanja primjera klasi kojoj je dodijeljen.
10. Primijeniti algoritam perceptronu na zadane podatke
- stvarni život.

# 6 Logistička regresija

1. Definirati i skicirati logističku funkciju te objasniti njezine prednosti

$$\sigma(\alpha) = \frac{1}{1 + \exp(-\alpha)}$$



a. Prednosti:

- i. aktivacijska funkcija slična funkciji praga
- ii. vrijednost gnijeći na interval  $<0,1>$  što omogućava vjerojatnosnu interpretaciju
- iii. derivabilna

$$\frac{\partial \sigma(\alpha)}{\partial \alpha} = \frac{\partial}{\partial \alpha} (1 + \exp(-\alpha))^{-1} = \sigma(\alpha)(1 - \sigma(\alpha))$$

2. Model logističke regresije:

$$h(\mathbf{x}; \mathbf{w}) = \sigma(\mathbf{w}^T \phi(\mathbf{x})) = \frac{1}{1 + \exp(-\mathbf{w}^T \phi(\mathbf{x}))} = P(y = 1 | \mathbf{x})$$

Izlaz modela možemo tumačiti kao vjerojatnost da primjer pripada klasi  $y=1$

3. Definirati pojam poopćenog linearog modela.

- a. Poopćeni linearni model (GLM) - linearan model s (nelinearnom) aktivacijskom funkcijom. Nelinearna funkcija omotana oko linearne.

4. Izvod pogreške unakrsne entropije:

- Izlaz modela je **Bernoullijeva varijabla**:

$$P(y|\mu) = \begin{cases} \mu & \text{ako } y = 1 \\ 1 - \mu & \text{inače} \end{cases} = \mu^y(1 - \mu)^{1-y}$$

- U našem slučaju,  $y$  je oznaka primjera, a  $\mu$  je izlaz modela, tj.  $\mu = h(\mathbf{x}; \mathbf{w})$ , pa:

$$P(y^{(i)}|\mathbf{x}^{(i)}) = h(\mathbf{x}; \mathbf{w})^y(1 - h(\mathbf{x}; \mathbf{w}))^{1-y}$$

- Log-izglednost oznaka iz skupa označenih primjera:

$$\begin{aligned} \ln P(\mathbf{y}|\mathbf{X}, \mathbf{w}) &= \ln \prod_{i=1}^N p(y^{(i)}|\mathbf{x}^{(i)}) = \\ &= \sum_{i=1}^N \left( y^{(i)} \ln h(\mathbf{x}^{(i)}; \mathbf{w}) + (1 - y^{(i)}) \ln (1 - h(\mathbf{x}^{(i)}; \mathbf{w})) \right) \end{aligned}$$

- Empirijska pogreška je negativna log-izglednost:

$$E(\mathbf{w}|\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \left( -y^{(i)} \ln h(\mathbf{x}^{(i)}; \mathbf{w}) - (1 - y^{(i)}) \ln (1 - h(\mathbf{x}^{(i)}; \mathbf{w})) \right)$$

⇒ **pogreška unakrsne entropije (cross-entropy error)**

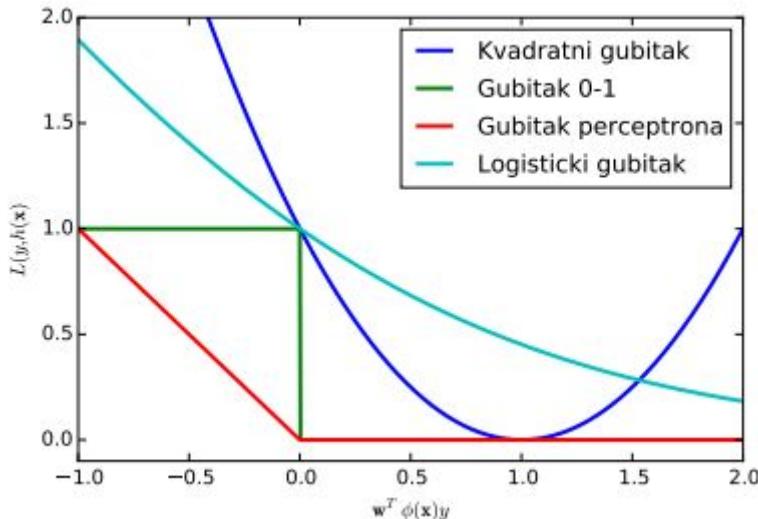
5. (\*) Izvesti logističku funkciju gubitka za oznake -1 i +1

- a. Reformulacijom funkcije gubitka unakrsne entropije definirane za  $y\{0,1\}$  na  $y\{-1,1\}$  dobijemo

$$L(y, h(\mathbf{x})) = \frac{1}{\ln 2} \ln \left( 1 + \exp(-y \mathbf{w}^T \phi(\mathbf{x})) \right)$$

(podijelili smo s  $\ln 2$  zbog tog što je gubitak upravo  $\ln 2$  kada je primjer na granici, tako da sada kroz nulu gubitak prolazi s 1)

6. Skicirati funkciju logističkog gubitka i usporediti ju s drugim funkcijama gubitka



- a. Logistički gubitak dobar je za klasifikaciju jer ne kažnjava jako ispravne primjere za razliku od kvadratnog gubitka.

7. Objasniti optimizaciju gradijentnim spustom i objasniti inačice algoritma
- Kod optimizacije želimo:  $w^* = \operatorname{argmin} E(w|D)$ 
    - ne postoji rješenje u zatvorenoj formi (zbog nelinearne sigmoidalne funkcije), zato radimo gradijentni spust (spuštamo se do minimuma).

$$\mathbf{x} \leftarrow \mathbf{x} - \eta \nabla f(\mathbf{x})$$

- pri čemu je  $\eta$  stopa učenja - eta
- Minimum nalazimo krećući se u smjeru **suprotnom od gradijenta**.

Ako je  $\eta$  prevelika dolazi do divergencije, ako je premala konvergencija je spora. Želimo globalnu konvergenciju → konveksna funkcija.

Postoje dvije inačice:

- **Batch** (grupni):  $\mathbf{w} \leftarrow \mathbf{w} - \eta \sum_{i=1}^N \nabla L(y^{(i)}, h(\mathbf{x}^{(i)}; \mathbf{w}))$
- **Stohastički (SGD)**:  $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla L(y^{(i)}, h(\mathbf{x}^{(i)}; \mathbf{w}))$

8. Izvesti algoritam grupnog/stohastičkog grad. spusta (**L2-reg.**) logističke regresije

---

**Algoritam 3.** Regularizirana logistička regresija (gradijentni spust)

---

- 1:  $\tilde{\mathbf{w}} \leftarrow (0, 0, \dots, 0)$
- 2: ponavljaj do konvergencije:
- 3:  $\Delta w_0 \leftarrow 0$
- 4:  $\Delta \mathbf{w} \leftarrow (0, 0, \dots, 0)$
- 5: za  $i = 1, \dots, N$
- 6:    $h \leftarrow \sigma(\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}^{(i)})$
- 7:    $\Delta w_0 \leftarrow \Delta w_0 + h - y^{(i)}$
- 8:    $\Delta \mathbf{w} \leftarrow \Delta \mathbf{w} + (h - y^{(i)}) \mathbf{x}^{(i)}$
- 9:    $w_0 \leftarrow w_0 - \eta \Delta w_0$
- 10:    $\mathbf{w} \leftarrow \mathbf{w}(1 - \eta \lambda) - \eta \Delta \mathbf{w}$

---

**Algoritam 4.** Regularizirana logistička regresija (stoh. gradijentni spust)

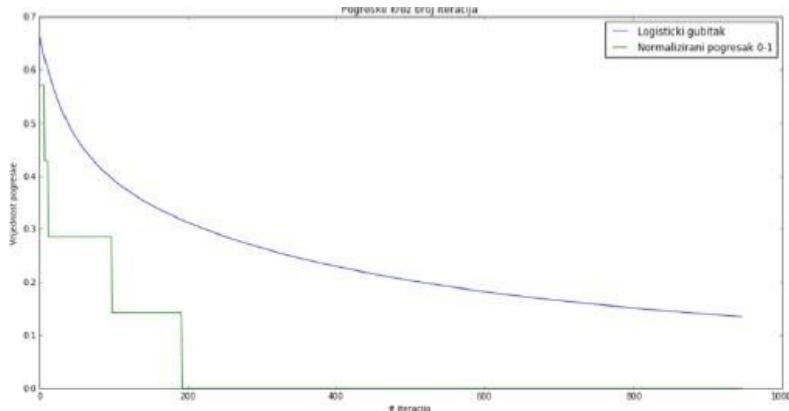
---

- 1:  $\tilde{\mathbf{w}} \leftarrow (0, 0, \dots, 0)$
- 2: ponavljaj do konvergencije:
- 3: slučajno permutiraj primjere u  $\mathcal{D}$
- 4: za  $i = 1, \dots, N$
- 5:    $h \leftarrow \sigma(\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}^{(i)})$
- 6:    $w_0 \leftarrow w_0 - \eta(h - y^{(i)})$
- 7:    $\mathbf{w} \leftarrow \mathbf{w}(1 - \eta \lambda) - \eta(h - y^{(i)}) \mathbf{x}^{(i)}$

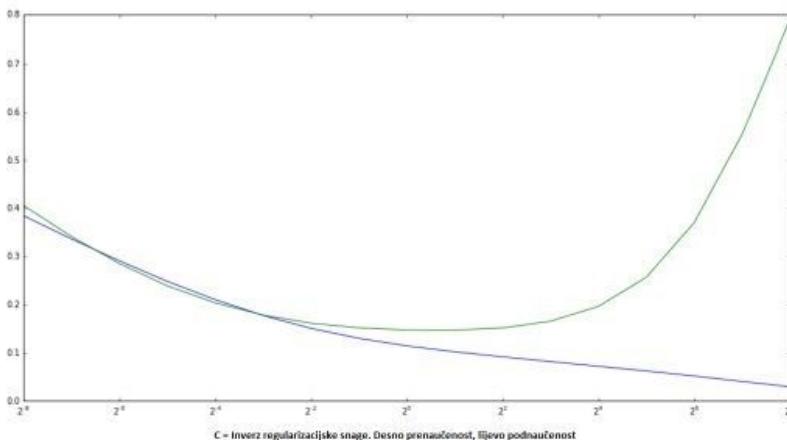
9. Objasniti prednosti regularizacija u kontekstu logističke regresije:
- sprječavanje prenaučenosti usred pretjerane nelinearnosti
  - suzbijanje nepotrebnih značajki
  - sprječavanje otvrđnjavanja sigmoide kod linearno odvojivih problema

10. Skicirati i objasniti pogrešku logističke regresije kao funkciju parametara ili iteracija

a. Pogreška logističke regresije kao funkcija **iteracija** - funkcija pada



b. Pogreška logističke regresije kao funkcija **parametara** - model je lijevo podnaučen, desno prenaučen.



11. Analizirati komponente i induktivnu pristranost algoritma logističke regresije

a. **Model**

- poopćeni linearan model s funkcijama preslikavanja i aktivacijskom funkcijom sigmoidom (za 2 klase) ili softmax (više od 2 klase)
- pristranost **ograničenja**

b. **Funkcija gubitka**

- biramo hipotezu koja minimizira negativnu pogrešku unakrsne entropije tj. maksimizira log izglednost skupa primjera za učenje
- pristranost **preferencijom** (modeliramo ulaz modela kao Bernoullijevu varijablu)

c. **Optimizacija**

- korištenjem gradijentnog spusta, ažuriramo parametre u smjeru najvećeg (najbržeg) pada gradijenta empirijske pogreške
- pristranost **preferencijom**

# 7 Logistička regresija II

PREŠAO/LA SI POLA PUTA, MOŽEŠ TI TO

1. Objasniti (kvazi-)Newtonov postupak i motivaciju za njegovu primjenu
  - a. Kako je gradijentni spust s linijskim pretraživanjem jako spor, kao alternativa se koristi optimizacija drugog reda, npr. Newtonov postupak. Vrši se kvadratna aproksimacija razvojem u Taylorov red drugog reda te se parametri ažuriraju:

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

S obzirom na to da je izračun  $H_t$  (Hesseove matrice funkcije u točki) potrebno provesti za svako ažuriranje težina što je skupo, kao alternativa se koriste kvazi-Newtonovi postupci koji pomoću gradijenta aproksimiraju  $H_t$  u svakom koraku.

2. Definirati funkciju softmax i objasniti što se njome ostvaruje
  - a. **Softmax**: popćenje sigmoide na k klase, skalira sve vrijednosti da su veće od 0 i da u zbroju daju 1, veće povećava još više, a manje smanjuje

- **Funkcija softmax**:  $\text{softmax} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , gdje za komponentu  $k$  vrijedi:

$$\text{softmax}_k(x_1, \dots, x_n) = \frac{\exp(x_k)}{\sum_j \exp(x_j)}$$

$\Rightarrow$  normalizira tako da  $\sum x_k = 1$  te smanjuje male i povećava velike vrijednosti

3. (\*) Demonstrirati da je funkcija softmax popćenje logističke funkcije
4. Napisati model multinomijalne logističke regresije i objasniti interpretaciju izlaza
  - a. Izlaz modela multinomijalne regresije možemo tumačiti kao vjerojatnost da primjer pripada klasi  $k$ .

$$h_k(\mathbf{x}; \mathbf{W}) = \frac{\exp(\mathbf{w}_k^T \phi(\mathbf{x}))}{\sum_j \exp(\mathbf{w}_j^T \phi(\mathbf{x}))} = P(y = k | \mathbf{x}, \mathbf{W})$$

5. Izvesti popćenu pogrešku unakrsne entropije
  - a. S obzirom na to da je izlaz modela multinulijeva varijabla  $y$  s distribucijom

$$P(\mathbf{y} | \boldsymbol{\mu}) = \prod_{k=1}^K \mu_k^{y_k}$$

tražimo njenu log izglednost:

$$\begin{aligned} \ln P(\mathbf{y} | \mathbf{X}, \mathbf{W}) &= \ln \prod_{i=1}^N P(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}) = \ln \prod_{i=1}^N \prod_{k=1}^K \mu_k^{y_k^{(i)}} = \ln \prod_{i=1}^N \prod_{k=1}^K h_k(\mathbf{x}^{(i)}; \mathbf{W})^{y_k^{(i)}} \\ &= \sum_{i=1}^N \sum_{k=1}^K y_k^{(i)} \ln h_k(\mathbf{x}^{(i)}; \mathbf{W}) \end{aligned}$$

iz čega slijedi popćena pogreška unakrsne entropije kao negativna log izglednost.

$$E(\mathbf{W} | \mathcal{D}) = - \sum_{i=1}^N \sum_{k=1}^K y_k^{(i)} \ln h_k(\mathbf{x}^{(i)}; \mathbf{W})$$

6. Izvesti i objasniti algoritam LMS.

a. Izvod je kompliciran, ali svodi se na uvrštavanje  $h(\mathbf{x}; \mathbf{w})$  u izraz gore i deriviranje po težinama te izjednačavanje s nulom. Ono sto dobijemo je gradijent, pa tezine samo azuriramo djelicem tog gradijenta. Konačna formula:

$$\mathbf{w}_k \leftarrow \mathbf{w} - \eta(h(\mathbf{x}^{(i)}; \mathbf{w}) - y^{(i)}) \phi(\mathbf{x}^{(i)})$$

7. Izračunati predikciju modela multinomijalne regresije za zadane primjere

a. Stvarni život.

8. (\*) Povezati poopćene linearne modele s distribucijama iz eksponencijalne familije

a. Eksponencijalna familija distribucija je skup distribucija koje su poseban slučaj jednog generaliziranog modela. Iz tog modela, odabirom parametara na poseban nacin, mogu se dobiti normalna (Gaussova), binomna, geometrijska, Poissonova, eksponencijalna... U algoritmima iz prethodnih poglavlja, od distribucija koristili smo Gaussovou (za linearu regresiju), Bernoullijevu (za logisticku) i Multinoullijevu (za multinomijalnu logisticku). Za svaki od tih modela koristili smo posebnu funkciju aktivacije (za linearu identitet, za logisticku sigmoidu, za multi softmax). Svi ti modeli su poopćeni linearni modeli, jedino se razlikuju u aktivacijskim funkcijama, a svakoj aktivacijskoj funkciji odgovara jedna distribucija iz poopćene familije distribucija koju zovemo eksponencijalna familija.

9. Objasniti vezu između logističke regresije i umjetnih neuronskih mreža

a. Logistička regresija se koristi kao **zadnji** sloj neuronske mreže ako se mreža koristi za **klasifikaciju**

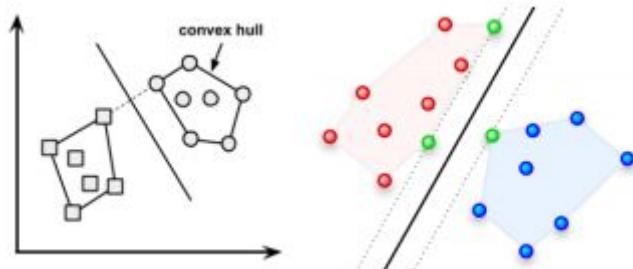
- i. ukoliko imamo više od 2 klase - afina transformacija i koristi se aktivacija softmax za klasificiranje
- ii. ukoliko razlikujemo samo 2 klase - logistička funkcija

# 8 Stroj potpornih vektora

1. Definirati maksimalnu marginu, objasniti motivaciju i dati primjere.

a. **Margina**

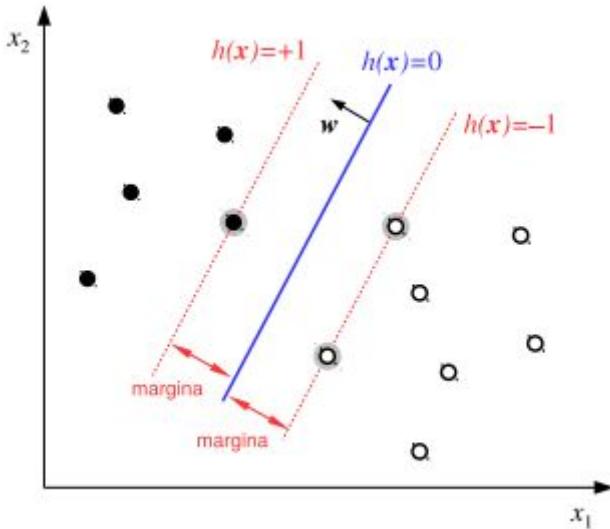
- i. udaljenost između hiperravnine i najbližih primjera obiju klasa.
- ii. Za skupove primjera postoji više mogućih pravaca koji se mogu povući da bi razdvojili klase - idealno na **polovici**, odnosno naći **što veću** marginu.
- iii. SVM pronalazi maksimalnu marginu što daje dobru generalizaciju.



2. Objasniti maksimalnu marginu u kontekstu konveksnih ljsaka klasa.

- a. Geometrijski - hiperravnina je simetrala spojnica konveksnih ljsaka 2 klasa (slika iznad). Napravimo konveksnu ljsku, nađemo 2 najbliža primjera i između njih povučemo pravac (simetrala konveksnih vrhova).
- b. Ako su dva skupa točaka konveksni i disjunktni, onda su oni nužno linearno odvojivi. [https://en.wikipedia.org/wiki/Hyperplane\\_separation\\_theorem](https://en.wikipedia.org/wiki/Hyperplane_separation_theorem)

3. Izvod modela SVM-a i kvadratni problem maksimalne margine.



- Uz pretpostavku **linearne odvojivosti** i uz  $y \in \{-1, +1\}$ , vrijedi:

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

- Udaljenost primjera  $\mathbf{x}^{(i)}$  od hiperravnine je  $\frac{1}{|\mathbf{w}|} y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + w_0)$
- Tražimo hiperravninu maksimalne margine:

$$\operatorname{argmax}_{\mathbf{w}, w_0} \left\{ \frac{1}{|\mathbf{w}|} \min_i \{y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + w_0)\} \right\}$$

- Vektor  $(\mathbf{w}, w_0)$  možemo skalirati tako da za primjere najbliže margini vrijedi:

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

- Onda za sve primjere vrijedi:

$$y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + w_0) \geq 1, \quad i = 1, \dots, N$$

- Optimizacijski problem svodi se na:

$$\operatorname{argmax}_{\mathbf{w}, w_0} \frac{1}{|\mathbf{w}|}$$

**uz ograničenja:**

$$y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + w_0) \geq 1, \quad i = 1, \dots, N$$

- Ekvivalentno:

$$\operatorname{argmin}_{\mathbf{w}, w_0} \frac{1}{2} |\mathbf{w}|^2$$

uz ograničenja:

$$y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + w_0) \geq 1, \quad i = 1, \dots, N$$

⇒ konveksna optimizacija uz ograničenje, preciznije **kvadratno programiranje**

4. (\*) Objasniti Lagrangeovu funkciju i uvjete KKT

- a. Lagrangeova funkcija je nova funkcija koju definiramo sa nasom funkcijom koju minimiziramo i sa ugradenim ogranicenjima. Stacionarne tocke te nove funkcije jesu minimumi pocetne funkcije uz ogranicenja.
  - b. U tim stacionarnim tockama vrijede neka svojstva/pravila (uvjeti je los prijevod od "condition" jer je to vise stanje). Ta pravila su prvi dokazali Karun, Kush i Tucker. Neki od uvjeta su da su koeficijenti ispred ogranicenja uvijek veci ili jednaki 0, te je ili ogranicenje 0, ili je koeficijent 0.
    - i. Ovo potonje KKT pravilo je krivo zasto su (u dualnoj formulaciji SVM) koeficijenti ispred samo nekih vektora razliciti od 0, te su upravo ti vektori potporni vektori.
5. Napisati Lagrangeovu funkciju i uvjete KKT za SVM za tvrdom marginom.

- a. Lagrangeova funkcija (*//ovo je općenita langrangeova funkcije ne za SVM*)

$$L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^m \alpha_i g_i(\mathbf{x}) + \sum_{i=1}^p \beta_i h_i(\mathbf{x}) \quad \text{gdje } \alpha_i \geq 0$$

- i. matematicki, alfa može biti 0, ako je onda je suma isključena, a ako nije onda imamo ovo ograničenje

$$\alpha g(\mathbf{x}) = 0$$

- b. Uvjeti (čine ih izvorna ograničenja i dva uvjeta za alfu):

$$g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m$$

$$h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p$$

- i. u svakom slučaju za točku rješenja vrijedi

$$\alpha g(\mathbf{x}) = 0$$

c.  $L(w, w_0, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i [y^i (\mathbf{w}^T \mathbf{x}^{(i)} + w_0) - 1]$

- d. Lagrangeova funkcija za SVM sa tvrdom marginom: (*//ovo je langrangeova funkcije za SVM*)

- i. Lagrangeova funkcija

$$L(\widehat{\mathbf{w}}, w_0, \widehat{\boldsymbol{\alpha}}) = \frac{1}{2} \|\widehat{\mathbf{w}}\|^2 - \sum_{i=1}^N \alpha_i [y^i (\widehat{\mathbf{w}}^T \widehat{\mathbf{x}}^{(i)} + w_0) - 1]$$

- ii. Dual

$$\widehat{\mathbf{w}} = \sum_{i=1}^N \alpha_i y^{(i)} \widehat{\mathbf{x}}^{(i)}$$

- iii. dualna Lagrangeova funkcija (maksimiziramo)

$$\bar{L}(\widehat{\boldsymbol{\alpha}}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^{(i)} y^{(j)} (\widehat{\mathbf{x}}^{(i)})^T \widehat{\mathbf{x}}^{(j)}$$

- iv. uvjeti KKT u točki rješenja:

$$y^{(i)} (\widehat{\mathbf{w}}^T \widehat{\mathbf{x}}^{(i)} + w_0) \geq 1, \quad i = 1, \dots, N$$

$$\alpha_i \geq 0, \quad i = 1, \dots, N$$

$$\alpha_i (y^{(i)} h(\widehat{\mathbf{x}}^{(i)}) - 1) = 0, \quad i = 1, \dots, N$$

6. (\*) Objasniti načelo dualnosti i Lagrangeovu dualnu funkciju.

- a. Rješenje originalnog problema je stacionarna točka od  $L$  u kojoj je gradijent od  $L$  0. Tražimo stacionarnu točku Lagrangeove fje koja ima i  $\mathbf{x}$  i alfu, a pitanje kako ju naći nas dovodi upravo do načela dualnosti. Načelo dualnosti nam govori da je svaki optimizacijski problem moguce izraziti primarno ili dualno. Izmedu ta dva pogleda postoji odredni "duality gap" (koji je u primjeru sa SVM zapravo 0). Primjerice, rjesavanjem primarnog problema deriviranjem i

izjednacavanjem s 0 možda ne isčeznu svi potrebni koeficijenti. Ta funkcija je onda dualna funkcija problema, koja je u svim svojim tockama minimum po primarnim varijablama. Da bismo smanjili duality gap, nuzno je tu dualnu funkciju maksimizirati po sekundarnim varijablama. Ta maksimizacija zatim daje optimum pocetne funkcije, u odnosu na ogranicenja.

7. Izvesti dualni kvadratni problem maksimalne margine i dualni model SVM-a.

a. Lagrangeova funkcija za problem maksimalne margine:

$$L(\mathbf{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i \left\{ y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + w_0) - 1 \right\} \quad \text{gdje } \boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N), \alpha_i \geq 0.$$

i. Ona se derivira po težinama kako bismo prešli u dual:

$$\begin{aligned} \frac{\partial L(\mathbf{w}, w_0, \boldsymbol{\alpha})}{\partial \mathbf{w}} &= 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^N \alpha_i y^{(i)} \mathbf{x}^{(i)} \\ \frac{\partial L(\mathbf{w}, w_0, \boldsymbol{\alpha})}{\partial w_0} &= 0 \quad \Rightarrow \quad \sum_{i=1}^N \alpha_i y^{(i)} = 0 \end{aligned}$$

ii. Što uvrštavamo u Lagrangeovu funkciju, čime dobivamo dualnu Lagrangeovu funkciju:

$$\begin{aligned} \tilde{L}(\boldsymbol{\alpha}) &= \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i \left\{ y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + w_0) - 1 \right\} \\ &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^T \mathbf{x}^{(j)} \end{aligned}$$

iii. A dualni kvadratni problem maksimalne margine upravo je **maksimizirati funkciju gubitka** tako da zadovoljava KTT uvjete

$$\begin{aligned} y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + w_0) &\geq 1, \quad i = 1, \dots, N \\ \alpha_i &\geq 0, \quad i = 1, \dots, N \\ \alpha_i (y^{(i)} h(\mathbf{x}^{(i)}) - 1) &= 0, \quad i = 1, \dots, N \end{aligned}$$

b. Dualni model SVM-a

- i. umjesto sa težinama, radimo sa primjerima, "trčimo" kroz sve te gledamo koliko koji utječe na izlaz.
- ii. Skalarni umnožak između ulaznog vektora i primjera zapravo mjeri sličnost između ta dva vektora.
- iii. Doduše, neće biti potrebno trčati kroz sve primjere, nego samo one za koje je Lagrangeov multiplikator alpha razlicit od 0 - potporni vektori

$$h(\mathbf{x}) = \underbrace{\mathbf{w}^T \mathbf{x} + w_0}_{\text{Primarno}} = \underbrace{\sum_{i=1}^N \alpha_i y^{(i)} \mathbf{x}^T \mathbf{x}^{(i)}}_{\text{Dualno}} + w_0$$

8. Usporediti primarnu i dualnu formulaciju SVM-a te navesti prednosti svake od njih
  - a. Prednosti dualnog
    - i. reducirali smo broj varijabli na N
    - ii. složenost  $O(N^3)$  ili  $O(N^2)$  uz SMO. ....
9. Definirati pojam potpornog vektora
  - a. **Potporni vektori** su vektori koji leže na ravninama maksimalne margine (s njezine jedne ili druge strane)
    - i. Svi ostali vektori, za koje je  $\alpha_i = 0$  (KKT uvjeti), uopće ne utječu na izlaz modela i možemo ih posve zanemariti
  - b. U praksi znači da, nakon učenja modela, možemo zadržati samo potporne vektore te je granična hiperravnina definirana linearnom kombinacijom tih vektora. To vrijedi samo za predikciju pomoću modela. Za učenje modela trebaju nam svi primjeri iz skupa za učenje.
10. Izračunati predikciju i dualne/primarne parametre iz primarnih/dualnih parametara.
  - a. Stvarni život
11. Analizirati komponente i induktivnu pristranost algoritma SVM.
  - a. **Model:** razdvajajuća hiperravnina s najvećom mogućom udaljenosti do najbližeg primjera
  - b. **Funkcija gubitka:** Hinge loss, odnosno  $\max(0, 1 - h(\mathbf{x}; \mathbf{w})y)$
  - c. **Optimizacija:** algoritmi kvadratnog programiranja (najčešće SMO, ali može i SGD, SGP, koordinatni spust, hooke jeeves, simpleks...)

## 9 Stroj potpornih vektora II

1. Definirati problem meke maksimalne margine i objasniti njenu motivaciju.
  - a. Govorilo se o tome kako su primjeri linearno odvojivi, ali u stvarnom svijetu zapravo nije tako. Stoga se nameće pitanje kako riješiti taj problem? Jednostavno, dozvoljavanjem "uleta" u marginu, ali jednako tako i kažnjavanjem tih istih primjera koji su uletjeli, što upravo i opisuje pojam meke margine. (Čim dublji ulet rezultirat će većom kaznom)
2. Izvesti dualni kvadratni problem meke maksimalne margine
  - a. Kada želimo prijeći u dual tražimo minimum po primarnim parametrima

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{w}} &= 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^N \alpha_i y^{(i)} \mathbf{x}^{(i)} \\ \frac{\partial L}{\partial w_0} &= 0 \quad \Rightarrow \quad \sum_{i=1}^N \alpha_i y^{(i)} = 0 \\ \frac{\partial L}{\partial \xi_i} &= 0 \quad \Rightarrow \quad \alpha_i = C - \beta_i\end{aligned}$$

te dobiveno uvrštavamo u L da bismo dobili **dualnu Lagrangeovu funkciju**

$$\tilde{L}(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\top \mathbf{x}^{(j)}$$

gdje je pripadni dualni optimizacijski problem

$$\underset{\alpha}{\operatorname{argmax}} \left( \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\top \mathbf{x}^{(j)} \right)$$

uz ograničenja

$$\sum_{i=1}^N \alpha_i y^{(i)} = 0, \quad i = 1, \dots, N$$

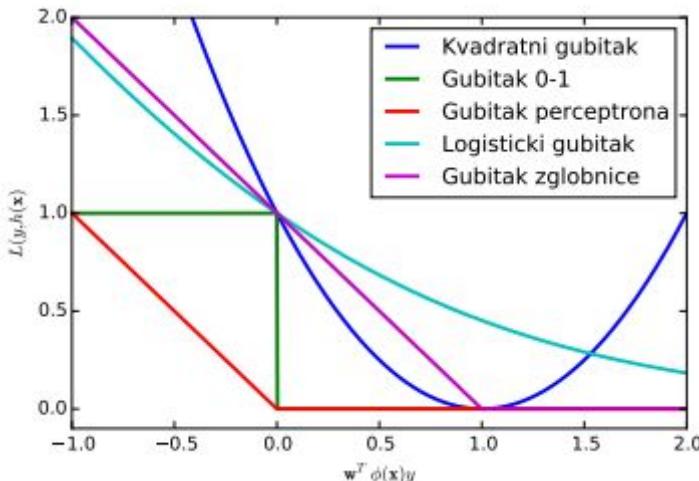
pri čemu iz uvjeta  $\beta \geq 0$ ,  $\alpha \geq 0$ ,  $\beta = C - \alpha$  slijedi ograničenje

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, N$$

3. Napisati Lagrangeovu funkciju i uvjete KKT za SVM za tvrdom marginom.
  - a. Pogledati 8.5.
4. Definirati i skicirati funkciju gubitka zglobnice.
  - a. graf gubitka zglobnice je, među ostalim, prikazan u idućem pitanju (boja - ljubičasta)
  - b. Formula glasi:

$$L(y, h(\mathbf{x})) = \max(0, 1 - yh(\mathbf{x}))$$

5. Skicirati i usporediti funkcije gubitka SVM-a i drugih linearnih modela.



6. Definirati funkciju pogreške SVM-a i objasniti ulogu hiper parametra C.

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

dobivamo:

$$E(\mathbf{w} | \mathcal{D}) = \sum_{i=1}^N \max(0, 1 - y^{(i)} h(\mathbf{x}^{(i)})) + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

gdje  $\lambda = 1/C$

- a. C je kao "anti regulacijski" parametar (oprez, nestandardna terminologija!)

- i. Što je C **manji**, SVM model je **više** regulariziran
  - ii. odnosno što je C **veći**, SVM **manje** regulariziran i skloniji overfittanju
  - iii. C je hiper parametar koji određuje složenost modela, veliki C daje složenije model, mali C jednostavnije
7. Izračunati gubitak/pogrešku zadano modela SVM-a na zadanoj primjeru/skupu.
- vjerojatno kao i ona druga pitanja di je odg "stvarni život"
8. Objasniti potrebu za skaliranjem značajki.
- Zbog predikcije SVM modela u dualnoj formi, gdje skalarnim produktom usporedujemo ulazni vektor sa potpornim vektorima, u skalarnom produktu će daleko više dominirati značajke sa većom skalom (primjerice prihod vs dob). Zbog toga je uputno skalirati sve značajke na isti interval (normalizacija) ili zero-mean-unit-variance (standardizacija).
9. (\*) Objasniti Plattovu metodu probabilističke kalibracije SVM-a.
- Jednostavno provući izlaz SVM modela kroz sigmoidu sa parametrima a i b koji se uče.

## 10 Jezgrena metode

1. Definirati jezgenu funkciju, objasniti motivaciju i navesti primjere uporabe.
- Kad ne znamo izdvojiti/prepoznati značajke možemo računati sličnost između primjera. Umjesto težina uz vektor značajki  $\mathbf{x}$ , izračunamo sličnost dvaju primjera. Naročito prikladno kada se primjeri teško vektoriziraju (npr. jer imaju strukturu). Za računanje sličnosti koriste se jezgrena funkcija

$$\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$

- b. Jezgrena funkcija je mjera sličnosti ako zadovoljava:

- $\kappa(\mathbf{x}, \mathbf{x}) = 1$
- $0 \leq \kappa(\mathbf{x}, \mathbf{x}') \leq 1$
- $\kappa(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x}', \mathbf{x})$

- c. Primjene: string kernels, tree kernels, graph kernels

2. Definirati tipične jezgrena funkcije u vektorskome prostoru:

- **Linearna jezgra**:  $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$
- **Radijalna bazna funkcija (RBF)**: općenito jezgra tipa  $\kappa(\mathbf{x}, \mathbf{x}') = f(\|\mathbf{x} - \mathbf{x}'\|)$
- **Gaussova RBF-jezgra**:

$$\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

gdje je  $\sigma^2$  je širina pojasa (*bandwidth*),  $\gamma = 1/2\sigma^2$  je preciznost  
(manja  $\sigma^2 \Leftrightarrow$  veća  $\gamma \Leftrightarrow$  primjeri su međusobno sve različitiji)

- **Ekponencijalna jezgra**:  $\kappa(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|)$
- **Inverzna kvadratna jezgra**:  $\kappa(\mathbf{x}, \mathbf{x}') = \frac{1}{1 + \|\mathbf{x} - \mathbf{x}'\|^2}$

3. Parametri Gaussove jezgrena funkcije (slika iznad objašnjeno)

- $\sigma^2$  je širina pojasa - što je veća to je Gauss 'spljošten', manja daje uži Gauss

- b.  $\gamma$  preciznost - inverz varijance - što je veća preciznost to primjeri moraju biti bliže da bi bili slični.

### Mahalanobisova udaljenosst

4. Definirati i usporediti jezgreni stroj i rijetki jezgreni stroj

- a. **Jezgreni stroj** - poopćeni linearni model s preslikavanjem  $\phi$  koje za bazne funkcije  $\phi_j$  koristi jezgrene funkcije.

$$\phi(\mathbf{x}) = (1, \kappa(\mathbf{x}, \boldsymbol{\mu}_1), \kappa(\mathbf{x}, \boldsymbol{\mu}_2), \dots, \kappa(\mathbf{x}, \boldsymbol{\mu}_m))$$

gdje su  $\boldsymbol{\mu}_j$  centroidi u prostoru primjera. To je parametarski algoritam.

- b. **Rijetki jezgreni strojevi** - umjesto centroida koriste primjere za učenje. (L1 regularizacija)

$$\phi(\mathbf{x}) = (1, \kappa(\mathbf{x}, \mathbf{x}_1), \kappa(\mathbf{x}, \mathbf{x}_2), \dots, \kappa(\mathbf{x}, \mathbf{x}_N))$$

Provjeravaju sličnost ispitnog primjera sa primjerima za učenje. To je neparametarski algoritam.

5. Objasniti jezgredni trik i prednosti inverznog oblikovanja.

- a. **Jezgredni trik** - umnožak dvaju primjera  $\mathbf{x}$  i  $\mathbf{x}'$  u prostoru značajki možemo zamijeniti funkcijom

$$\kappa(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$$

koju nazivamo jezgrenom funkcijom (objašnjenje 1.pitanje). Jedna od mogućnosti za definiranje jezgrene funkcije je inverzno oblikovanje.

- b. **Prednost inverznog oblikovanja** je to što je lakše definirati jezgrenu funkciju nego preslikavanje  $\phi$  (preslikavanje je nepoznato), odabiremo jezgrenu funkciju izravno pa treniramo model s tom funkcijom (ona mora odgovarati skalarnom umnošku u nekom prostoru značajki).

6. Objasniti Mercerovu jezgru i demonstrirati to svojstvo za linearu i Gaussovou jezgru.

- a. **Mercerove jezgre** (pozitivno definitne jezgre) definirane su uvjetima:

Prema **Mercerovom teoremu**, ako je Gram-matrica  $\mathbf{K}$  pozitivno semidefinitna, tj.  $\forall \mathbf{x} \neq 0. \mathbf{x}^T \mathbf{K} \mathbf{x} \geq 0$  za svaki skup  $\mathcal{D}$ , onda je jezgrena funkciju  $\kappa$  uvijek moguće rastaviti na skalarni produkt vektora,  $\kappa(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ , gdje je prostor značajki Hilbertov prostor  $H$ ,  $\phi(\mathbf{x}) \in H$ .

pri čemu je Gram-matrica ili jezgrena matrica simetrična matrica u koju su pohranjeni svi parovi primjera iz skupa za učenje. Da bi jezgredni trik funkcionirao, jezgrena funkcija mora biti Mercerova jezgra.

Linearna jezgra:  $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$

Gaussova jezgra:  $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right\} = \exp\{-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\}$

7. Objasniti svojstva Hilbertovog prostora induciranih Gaussovom jezgrom:

- a. U tim prostoru svi primjeri su linearne nezavisne (dimenzija tog prostora je efektivno beskonačna) i cine bazu tog podprostora
- b. skalarni produkt je uvijek između 0 i 1
- c. ortogonalnost baze ovisi o preciznosti (parametar gamma): što je preciznost veća, skalarni umnožak će vise teziti 0 (primjeri su sve razliciti), a što je preciznost manja to su svi primjeri sličniji.

8. (\*) Dokazati da Gaussova jezgra inducira beskonačno dimenzijski prostor značajki.

- a. Za one koje zele znati vise:  
<https://stats.stackexchange.com/questions/80398/how-can-svm-find-an-infinite-feature-space-where-linear-separation-is-always-p>
- 9. Izračunati parametre/predikciju zadanog jezgrenog stroja za zadane primjere. Izračunati parametre/predikciju SVM-a sa zadanim primjerima i jezgrom.
  - a. Stvarni život
- 10. Objasniti optimizaciju hiperparametara C i  $\gamma$  kod SVM-a s Gaussovom jezgrom.
  - a. C je hiperparametar koji određuje složenost. Veća vrijednost za C znači da će model više kažnjavati pogreške pa će biti složeniji. Optimiramo ga unakrsnom provjerom. Ako odaberemo visoku vrijednost za  $\gamma$ , dobit ćemo složeniji model, pa će trebati pojačati regularizaciju odabirom manje vrijednosti za C. To se radi unakrsnom provjerom 2 parametra istovremeno - iscrpno udaljeno pretraživanje u unaprijed definiranom rasponu tzv. pretraživanjem po rešetci.
  - b. Za Gaussovou jezgru:  $\gamma$  kontrolira kojom brzinom  $k(x,x')$  teži k nuli u ovisnosti o udaljenosti. Ako je  $\gamma$  malen,  $k(x,x') \rightarrow 1$  i primjeri su u prostoru značajki grupirani zajedno, što lako dovodi do podnaučenosti. Ako je  $\gamma$  velik, onda  $k(x,x') \rightarrow 0$  pa su sve točke u prostoru značajki međusobno ortogonalne što dovodi do prenaučenosti.

# 11 Neparametarske metode

1. Razlikovati parametarske i neparametarske metode.

a. **Parametarski** modeli

i. **složenost** modela **ne ovisi o broju primjera** za učenje

1. Konkretno, probabilistički parametarski postupci prepostavljaju da se podaci pokoravaju nekoj teorijskim razdiobi (npr. Gaussovoj razdiobi)

ii. Učenje se svodi na nalažanje parametara prepostavljenje distibucije, broj koji ne ovisi o broju primjera

iii. Svaki primjer neovisno gdje se nalazi ima isti utjecaj - globalni.

b. **Neparametarski** model

i. **broj parametara**, a time i **složenost** modela, **raste s brojem primjera** za učenje

ii. Ovdje ne prepostavljamo da se podaci pokoravaju nekoj teorijskoj distribuciji.

iii. Neparametarski modeli (unatoč nazivu) **imaju** parametre, ali to **nisu** parametri neke **prepostavljene** distribucije

iv. Rade se lokalne aproksimacije na temelju sličnosti primjera.

2. Navesti prednosti i nedostatke parametarskih odnosno neparametarskih metoda.

a. Parametarski modeli imaju jače prepostavke o podacima

i. Ako su te prepostavke točne, onda su u pravilu parametarski modeli bolji od neparametarskih

ii. Ako se stvarni podaci ne pokoravaju prepostavljenoj teorijskoj radiobi, pogreška klasifikacije bit će razmjerno velika.

b. **Parametarski** modeli su **bolji** kad imamo **manji broj primjera**

c. **Neparametarski** modeli su **bolji** kad **nismo sigurni u distribuciju**, a imamo **dovoljno primjera**.

3. Usporediti parametarski i neparametarski model SVM-a.

a. SVM model:

$$h(\mathbf{x}) = \underbrace{\mathbf{w}^\top \mathbf{x} + w_0}_{\text{Primarno}} = \underbrace{\sum_{i=1}^N \alpha_i y^{(i)} \mathbf{x}^\top \mathbf{x}^{(i)}}_{\text{Dualno}} + w_0$$

b. Primarna formulacija je parametarski model, a dualna neparametarski

c. Broj parametara proporcionalan je broju potpornih vektora, koji ovisi o N

i. Kada  $N \ll n$  prikladno jer algoritam SMO ima složenost  $O(N^2)$ .

4. Napisati modele i opisati algoritme k-NN i težinski k-NN.

- a. k-NN je neparametarski klasifikacijski algoritam. Radi predikciju na temelju većinske oznake k najbližih susjeda:

$$h(\mathbf{x}) = \operatorname{argmax}_{j \in \{0, \dots, K-1\}} \sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \text{NN}_k(\mathbf{x})} \mathbf{1}\{y^{(i)} = j\}$$

Skupljamo glasove za svaki. Samo nam treba model, ne i funkcija pogreške i optimizacija.

Voronojeve ćelije/dijagram: novi primjer dobije oznaku klase primjera u čiju ćeliju upadne.

Kod težinskog k-NN postoji utjecaj primjera ovisno o udaljenosti/sličnosti.

$$h(\mathbf{x}) = \operatorname{argmax}_{j \in \{0, \dots, K-1\}} \sum_{i=1}^N \kappa(\mathbf{x}^{(i)}, \mathbf{x}) \mathbf{1}\{y^{(i)} = j\}$$

Bliži susjedi - veća težina, dalji manja. Za K se može koristiti bilo koja jezgrena funkcija, time otežamo glas za sličnost.

5. Objasniti ulogu hiperparametra k kod algoritma k-NN.

- a. k je hiperparametar koji određuje koliko susjeda uzimamo u obzir. Lako prenaučiti
  - i. ako gledamo samo prvi susjed, a on je šum - model loše generalizira
  - ii. Za veći k šum može biti nadglasan i dobije se bolja generalizacija
    - 1. Ne želimo ni preveliki k jer on određuje **složenost** - radi se unakrsna provjera

6. Pristupi za nalaženje najbližih susjeda:

- a. **egzaktne** metode: indeksiranje prostora primjera (ball tree)
- b. **aproksimativne** metode: locally sensitive hashing (LSH)

7. Objasniti problem prokletstva dimenzionalnosti.

- a. s porastom dimenzije n sve točke postaju međusobno vrlo udaljene
- b. udaljenosti postaju nediskriminativne
- c. općenit problem svih algoritama u visokodimenzijskim prostorima
- d. Što dimenzionalnost podataka raste to je manje primjera u polu prostora svih dimenzija
- e. S porastom dimenzija raste udaljenost između susjeda.

8. Definirati model k-NN za regresiju i jezgreno zaglađivanje.

- a. Neparametarska regresija = modeli zaglađivanja

- **k-nn smoother** - prosjek vrijednosti k najbližih susjeda:

$$h(\mathbf{x}) = \frac{1}{k} \sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \text{NN}_k(\mathbf{x})} y^{(i)}$$

- **Jezgreno zaglađivanje (kernel smoothing):**

$$h(\mathbf{x}) = \frac{\sum_{i=1}^N \kappa(\mathbf{x}^{(i)}, \mathbf{x}) y^{(i)}}{\sum_{i=1}^N \kappa(\mathbf{x}^{(i)}, \mathbf{x})}$$

9. Izračunati predikciju (težinske) k-NN klasifikacije/regresije za zadane primjere.

- a. Stvarni život



# 12 Ansamblji

1. Objasniti motivaciju za kombiniranje klasifikatora
  - a. Ovisno o tome koliko pristranost algoritma odgovara problemu kojeg imamo, dobivat ćemo različite rezultate. S obzirom na to da ne postoji algoritam koji je univerzalno najbolji, da svaki griješi na različit način, upravo njihovom kombinacijom možemo zadovoljiti naše potrebe.
2. Definirati modele glasanja i stackinga
  - a. **Glasanje:** uzmemu više modela te
    - i. uprosječimo njihove vrijednosti ako je regresija **ili**
    - ii. vratimo najčešću klasu kod klasifikatora
  - b. **Stacking:** izlaze različitim algoritama strojnog učenja kombiniramo i šaljemo novom modelu kao ulaz.
3. Objasniti postupak bagginga
  - a. Bagging uzima jedan algoritam strojnog učenja i generira mnogo modela i svaki overfitta na neki podskup skupa za ucenje. Zatim prilikom predikcije uprosječi izlaz svih. Bitno je da su svi podmodeli istog tipa i da se uce na drugom podskupu.
4. (\*) Objasniti naziv “0.632 bootstrap”.
  - a. 0.632 bootstrap algoritam radi na način da od skupa od N elemenata gradimo novi skup na način da N puta izaberemo neki nasumični element (elementi se, dakako, mogu ponavljati). Ime dolazi od sljedećeg: vjerojatnost za odabir nekog elementa je 1/N, dakle vjerojatnost da ga ne izaberemo je (1 - 1/N). Pošto radimo novi skup, tih odabira će biti (1-1/N)^N. Limes toga, kad N → +∞ je 1/e, odnosno 0.368, sto je vjerojatnost da neki element neće biti u jednom od skupova, što znači da će se neki element pojaviti u 63.2% generiranih podskupova.
5. Napisati pseudokod algoritma slučajne šume i objasniti prednosti tog algoritma
  - a. Prednosti: za svaki algoritam se dodatno radi odabir značajki
    - 1:  $\text{forest} \leftarrow \emptyset$
    - 2: Za  $j = 1 \dots L$
    - 3:  $\mathcal{D}_j \leftarrow \text{bootstrap uzorak}$
    - 4:  $\mathcal{F}_j \leftarrow \text{odabir } n' \text{ značajki}$
    - 5:  $h_j \leftarrow \text{treniraj stablo odluke na } \mathcal{D}_j \text{ sa značajkama } \mathcal{F}_j$
    - 6:  $\text{forest} \rightarrow \text{forest} \cup \{h_j\}$
6. Napisati pseudokod algoritma AdaBoost.
  - 1: inicijaliziraj težine primjera na  $w_j^i = 1/N$
  - 2: Za  $j = 1 \dots L$
  - 3:  $\mathcal{D}_j \leftarrow \text{bootstrap uzorak s težinama } \mathbf{w}_j$
  - 4:  $h_j \leftarrow \text{treniraj klasifikator na } \mathcal{D}_j$
  - 5: izračunaj pogrešku učenja  $E_j$
  - 6: pouzdanost  $\alpha_j \leftarrow \ln \frac{1-E_j}{E_j}$
  - 7: ažuriraj težine primjera:  $w_{j+1}^i \leftarrow w_j^i \exp \left( \alpha_j \mathbf{1}\{h_j(\mathbf{x}^{(i)}) \neq y^{(i)}\} \right)$
  - 8: normaliziraj vektor:  $\mathbf{w}_{j+1} \leftarrow \frac{\mathbf{w}_{j+1}}{\|\mathbf{w}_{j+1}\|}$

**Sretno na međuispitu!** 😊

# 13 Procjena parametara I

## 1. Navesti prednosti i nedostatke probabilističkih modela.

Prednosti:

- temelje se na teoriji vjerojatnosti, koja je vrlo razrađena
- modeliraju vjerojatnosti, čime dobivamo informaciju o pouzdanosti klasifikacije
- moguće je u model ugraditi apriorno znanje ukoliko ono postoji
- funkcioniра dobro i u slučaju kad nema puno podataka, ako su pretpostavke o podacima (razdiobe) točne

Probabilistički generativni modeli imaju naravno i neke nedostatke u odnosu na diskriminativne modele. Glavni nedostatci su:

- Broj primjera – modeliranje zajedničke vjerojatnosti  $P(\mathbf{x}, \mathcal{C}_j)$  iziskuje velik broj primjera, a da bi procjena bila pouzdana. To je osobit problem kada je ulazni prostor visoke dimenzije;
- Nepotrebna složenost modeliranja – ako je naš cilj klasifikacija, a ne generiranje primjera, onda je nepotrebno modelirati zajedničku vjerojatnost  $P(\mathbf{x}, \mathcal{C}_j)$ , koja može biti nepotrebno složena. U tom slučaju dovoljno je izravno modelirati samo posteriornu vjerojatnost  $P(\mathcal{C}_j | \mathbf{x})$ , kao što to čine diskriminativni modeli.

## 2. Objasniti vezu između učenja probabilističkih modela i procjene parametara.

Procjena parametara je procjena nekog stvarnog parametra populacije na temelju dostupnog uzorka. Na primjer, sredinom uzorka procjenjujemo srednju vrijednost populacije ( $\mu$ ), a varijancom uzorka procjenjujemo pravu varijancu populacije ( $\sigma$ ). Učenje modela predstavlja pronašetak optimalnih parametara koji dobro oponašaju razdiobu cijele populacije.

Pitanje - nije li to jedna te ista stvar?

// ovo je ctrl+c ctrl+v iz skripte sa stranice predmeta

Možda ćete sada uočiti da procjena parametara zapravo znači da određujemo parametre modela na temelju podataka. Nije li to zapravo učenje modela? Jest!

**3. Definirati i za zadani primjer izračunati očekivanje, (ko)varijancu i kovarijacijsku matricu.**

Primjer iz [DZ7](#):

| X \ Y    | 1    | 2    | 3   | $\Sigma$ |
|----------|------|------|-----|----------|
| 1        | 0.2  | 0.05 | 0.3 | 0.55     |
| 2        | 0.05 | 0.3  | 0.1 | 0.45     |
| $\Sigma$ | 0.25 | 0.35 | 0.4 |          |

Očekivanje:

$$E[X] = \sum_x x \cdot p(x) = \sum_x x \cdot \sum_y p(x,y) = 1 \cdot 0.55 + 2 \cdot 0.45 = 1.45$$

$$E[Y] = \sum_y y \cdot p(y) = \sum_y y \cdot \sum_x p(x,y) = 1 \cdot 0.25 + 2 \cdot 0.35 + 3 \cdot 0.4 = 2.15$$

Varijanca:

$$\text{Var}[X] = E[(X - E[X])^2] = \sum_x (x - E[X])^2 \sum_y p(x,y) = 0.2025 \cdot 0.55 + 0.3025 \cdot 0.45 = 0.2475$$

$$\text{Var}[Y] = E[(Y - E[Y])^2] = \sum_y (y - E[Y])^2 \sum_x p(x,y) = 1.3225 \cdot 0.25 + 0.0225 \cdot 0.35 + 0.7225 \cdot 0.4 = 0.6275$$

Kovarijanca:

$$\text{Cov}[X,Y] = E[(X - E[X])(Y - E[Y])] = \sum_x \sum_y (x - E[X]) \cdot (y - E[Y]) \cdot p(x,y) = -0.0175$$

Koeficijent korelacije:

$$\rho_{x,y} = \frac{\text{Cov}[X,Y]}{\sqrt{\text{Var}[X] \cdot \text{Var}[Y]}} = -0.1127$$

Kovarijacijska matrica:

$$\Sigma = \begin{bmatrix} \text{Var}[X] & \text{Cov}[X,Y] \\ \text{Cov}[Y,X] & \text{Var}[Y] \end{bmatrix} = \begin{bmatrix} 0.2475 & -0.0175 \\ -0.0175 & 0.6275 \end{bmatrix}$$

#### 4. Definirati Pearsonov koeficijent korelacije i vezu s nezavisnošću slučajnih varijabli.

Pearsonov koeficijent definiran je kao:

$$\rho_{x,y} = \frac{\text{Cov}[X, Y]}{\sqrt{\text{Var}[X] \cdot \text{Var}[Y]}}$$

Predstavlja mjeru u kojoj su slučajne varijable X i Y međusobno linearno zavisne. Za savršenu pozitivnu linearu zavisnost koeficijent postiže vrijednost 1, a za savršenu negativnu linearu zavisnost postiže vrijednost -1. Ako su dvije slučajne varijable nezavisne, njihov koeficijent korelacije će uvijek biti 0. Obrat ne vrijedi - korelacija 0 ne implicira nužno nezavisnost (jer koeficijent mjeri samo linearu zavisnost). Nekoliko primjera [ovdje](#).

#### 5. Povezati vrste varijabli/značajki s teorijskim vjerojatnosnim distribucijama.

Odabir vjerojatnosne distribucije ovisi o dimenzionalnosti podataka i o tome jesu li oni diskretni ili kontinuirani:

|                  | Diskretna                                       | Kontinuirana                     |
|------------------|-------------------------------------------------|----------------------------------|
| Jednodimenzijska | Bernoullijeva razdioba (za binarne vrijednosti) | Univarijatna normalna razdioba   |
|                  | Multinulijeva razdioba (za više vrijednosti)    |                                  |
| Višedimenzijska  | Vektor jednodimenzijskih varijabli              | Multivarijatna normalna razdioba |

#### 6. Definirati Bernoullijevu, kategoričku i uni/multivariatnu Gaussovou distribuciju.

| Naziv                      | Funkcija gustoće vjerojatnosti                                                                                      |
|----------------------------|---------------------------------------------------------------------------------------------------------------------|
| Bernoullijeva distribucija | $P(X = x \mu) = \begin{cases} \mu & \text{ako } x = 1 \\ 1 - \mu & \text{inače} \end{cases} = \mu^x(1 - \mu)^{1-x}$ |
| Kategorička distribucija   | $P(X = \mathbf{x} \boldsymbol{\mu}) = \prod_{k=1}^K \mu_k^{x_k}$                                                    |
| Gaussova distribucija      | $p(x \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$                  |

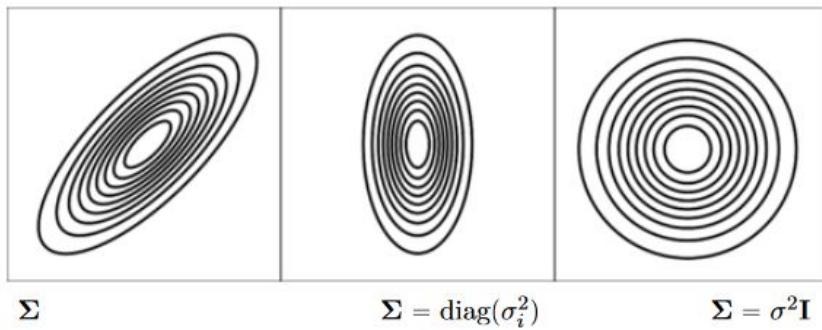
#### 7. Objasniti inačice Gaussove distribucije s ograničenjima na kovarijacijsku matricu.

Gustoća multivariatne Gaussove razdiobe definirana je kao:

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}$$

Varijante kovarijacijske matrice koje pojednostavuju izračun su:

|                                                |                                                                              |
|------------------------------------------------|------------------------------------------------------------------------------|
| dijeljena kovarijacijska matrica               | $\hat{\boldsymbol{\Sigma}} = \sum_j \hat{\mu}_j \hat{\boldsymbol{\Sigma}}_j$ |
| dijeljena i dijagonalna kovarijacijska matrica | $\boldsymbol{\Sigma} = \text{diag}(\sigma_i^2)$                              |
| izotropna kovarijacijska matrica               | $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$                                  |



Izgled Gaussove razdiobe u ovisnosti o kovarijacijskoj matrici

#### 8. Definirati statistiku, procjenitelj i pristranost procjenitelja.

**Statistika** je slučajna varijabla čija je vrijednost izračunata na temelju uzorka:

$$\Theta = g(X_1, X_2, \dots, X_N)$$

**Procjenitelj** je statistika koja pomoću uzorka procjenjuje neki nepoznat parametar populacije. Vrijednost (očekivanje) procjenitelja naziva se procjena. Razlika između očekivanja procjenitelja (on je slučajna varijabla) i pravog parametra populacije naziva se pristranost:

$$b_\theta(\Theta) = \mathbb{E}[\Theta] - \theta$$

Za procjenitelje čija je pristranost jednaka nuli kažemo da su nepristrani.

#### 9. \*Dokazati (ne)pristranost procjenitelja srednje vrijednosti i procjenitelja varijance.

[Nepristranost varijance \(wiki\)](#)

#### 10. Navesti metode za izvođenje procjenitelja.

Metoda za izvođenje procjenitelja su:

- procjenitelj najveće izglednosti (MLE)
- procjenitelj maximum a posteriori (MAP)
- Bayesovski procjenitelj (njih ne radimo)

## 14 Procjena parametara II

### 1. Definirati i objasniti funkciju izglednosti te vezu sa vjerojatnošću uzorka.

Funkcija izglednosti je funkcija:

$$\mathcal{L} : \boldsymbol{\theta} \mapsto p(\mathcal{D}|\boldsymbol{\theta})$$

koja parametrima  $\boldsymbol{\theta}$  pridjeljuje vjerojatnost da iz populacije s tim parametrima izvučemo uzorak  $D$ . Funkcija izglednosti ekvivalentna je vjerojatnosti da smo iz distribucije s parametrima  $\boldsymbol{\theta}$  izvukli uzorak  $D$ . Vjerojatnost uzorka jednaka je (uz pretpostavku nezavisnih i jednakih distribuiranih slučajnih varijabli – IID) umnošku

$$p(\mathcal{D}|\boldsymbol{\theta}) = p(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}|\boldsymbol{\theta}) = \prod_{i=1}^N p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) \equiv \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$$

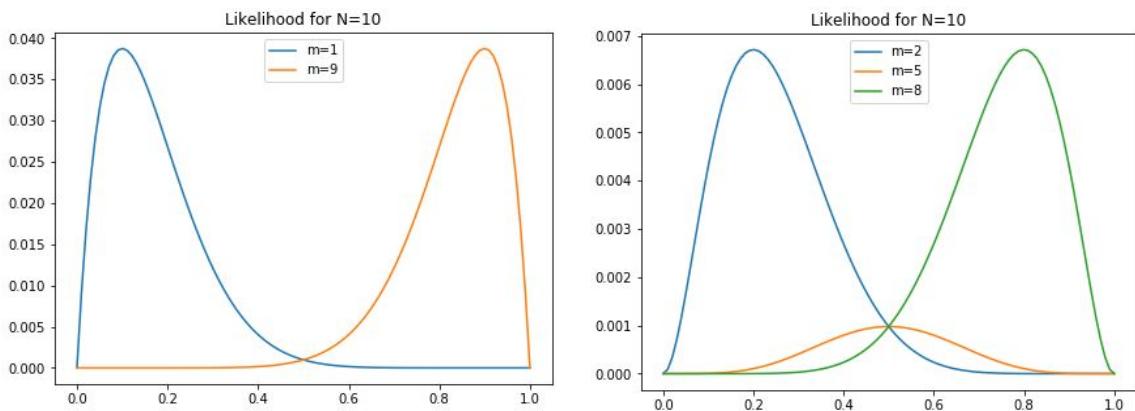
vjerojatnosti primjera: il

### 2. Napisati i skicirati funkciju izglednosti za Bernoullijevu varijablu.

Funkcija izglednosti za Bernoullijevu varijablu jednaka je:

$$\mathcal{L}(\mu|\mathcal{D}) = P(\mathcal{D}|\mu) = P(x^{(1)}, \dots, x^{(N)}|\mu) = \prod_{i=1}^N P(x^{(i)}|\mu) = \mu^m (1-\mu)^{N-m}$$

gdje je  $N$  broj događaja, a  $m$  broj pozitivnih ishoda (npr.  $N$  je broj bacanja novčića, a  $m$  koliko smo puta dobili glavu). Funkcija izglednosti za  $N = 10$  i različit broj pozitivnih ishoda:



### 3. Definirati procjenitelj MLE te navesti njegove mane i prednosti.

**Procjenitelj najveće izglednosti** (eng. *maximum likelihood estimator; MLE*) je procjenitelj koji nalazi parametre  $\theta$  koji maksimiziraju funkciju izglednosti, tj. parametre koji uzorak  $D$  koji nam je dostupan čine najvjerojatnijim:

$$\hat{\theta}_{\text{MLE}} = \underset{\theta}{\operatorname{argmax}} \mathcal{L}(\theta | D)$$

Prednosti:

- jednostavan model

Mane:

- lako ga je prenaučiti, previše se oslanja na ulazne podatke (prepostavlja da je dostupan uzorak najvjerojatniji mogući uzorak)
- ne daje dobre rezultate u slučaju da imamo mali uzorak
- nemoguće ugraditi apriorno znanje o razdiobi

#### 4. Izvesti procjenitelj MLE za Bernoullijevu i univariatnu Gaussovou razdiobu.

Log-izglednost Bernoullijeve varijable je:

$$\begin{aligned} \ln \mathcal{L}(\mu | D) &= \ln \prod_{i=1}^N P(x^{(i)} | \mu) = \ln \prod_{i=1}^N \mu^{x^{(i)}} (1-\mu)^{1-x^{(i)}} \\ &= \sum_{i=1}^N x^{(i)} \ln \mu + \left( N - \sum_{i=1}^N x^{(i)} \right) \ln (1-\mu) \end{aligned}$$

Njenom maksimizacijom dobijemo:

$$\begin{aligned} \frac{d \ln \mathcal{L}}{d \mu} &= \frac{1}{\mu} \sum_{i=1}^N x^{(i)} - \frac{1}{1-\mu} \left( N - \sum_{i=1}^N x^{(i)} \right) = 0 \\ \Rightarrow \quad \hat{\mu}_{\text{ML}} &= \frac{1}{N} \sum_{i=1}^N x^{(i)} = \frac{m}{N} \end{aligned}$$

Procjenitelj očekivanja  $\mu$  kod Bernoullijeve varijable zapravo je omjer broja realizacija i veličine uzorka, tj. **relativna frekvencija**.

Log-izglednost univariatne Gaussove varijable je:

$$\begin{aligned} \ln \mathcal{L}(\mu, \sigma^2 | D) &= \ln \prod_{i=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(x^{(i)} - \mu)^2}{2\sigma^2} \right\} \\ &= -\frac{N}{2} \ln(2\pi) - N \ln \sigma - \frac{\sum_i (x^{(i)} - \mu)^2}{2\sigma^2} \end{aligned}$$

Maksimizacijom log-izglednosti dobijemo:

$$\begin{aligned}\nabla \ln \mathcal{L}(\mu, \sigma^2 | \mathcal{D}) &= 0 \\ \Rightarrow \hat{\mu}_{\text{ML}} &= \frac{1}{N} \sum_{i=1}^N x^{(i)} \\ \Rightarrow \hat{\sigma}_{\text{MLE}}^2 &= \frac{1}{N} \sum_{i=1}^N (x^{(i)} - \hat{\mu}_{\text{MLE}})^2\end{aligned}$$

Procjenitelj varijance kod ovakvog modela nije nepristran. Može se korigirati ako izraz podijelimo sa  $N - 1$  umjesto s  $N$ .

### 5. Definirati procjenitelj MAP te navesti njegove mane i prednosti.

**Maksimum a posteriori procjenitelj (MAP)** za procjenu parametara, pored informacija koje dolaze iz podataka, koristi i pozadinsko znanje o njihovim mogućim vrijednostima. Pozadinsko znanje definira se apriornom distribucijom parametara  $p(\theta)$ , koja govori koliko su vjerojatne koje vrijednosti parametara. Aposteriornu vjerojatnost dobivamo pomoću Bayesovog pravila:

$$p(\theta | \mathcal{D}) = \frac{p(\mathcal{D}|\theta)P(\theta)}{p(\mathcal{D})}$$

MAP procjenitelj radi maksimizaciju te vjerojatnosti. Nazivnik izraza  $p(D)$  je konstantan, stoga se može zanemariti pri maksimizaciji. Model tada definiramo kao:

$$\hat{\theta}_{\text{MAP}} = \underset{\theta}{\operatorname{argmax}} \ p(\theta | \mathcal{D}) = \underset{\theta}{\operatorname{argmax}} \ p(\mathcal{D}|\theta) p(\theta)$$

Prednosti:

- moguće ugraditi pozadinsko znanje u model
- daje točnije rezultate od MLE procjenitelja u slučaju da nema puno podataka

Mane:

- češće je lakše i jeftinije skupiti više podataka nego formulirati apriornu vjerojatnost

### 6. Objasniti ideju konjugatnih distribucija i njihovu ulogu u procjeni parametara.

Kako bi aposteriornu vjerojatnost mogli analitički maksimizirati, potrebno je da distribucije  $p(D|\theta)$  i  $p(\theta)$ , kad se pomnože, daju poznatu teorijsku distribuciju, s kojom znamo raditi. Distribucija  $p(D|\theta)$  definirana je vrstom podataka s kojima radimo, dakle, potrebno je odabrati prikladnu apriornu distribuciju  $p(\theta)$ .

U slučaju kad su apriorna distribucija  $p(\theta)$  i aposteriorna distribucija  $p(\theta|D)$  iste vrste distribucija, kažemo da su one **konjugatne distribucije**.

Za svaku izglednost  $p(D|\theta)$  koja je iz eksponencijalne familije, postoji distribucija  $p(\theta)$  takva da su apriorna i aposteriorna distribucija konjugatne. Takva distribucija naziva se **konjugatna apriorna distribucija** za funkciju izglednosti.

$$p(\theta|D) \propto p(D|\theta) \cdot p(\theta)$$

konjugatne

konjugatna apriorna distribucija  
za izglednost  
 $p(D|\theta)$

Tablica konjugatnih apriorih distribucija:

| Aposteriorna $p(\theta D)$ | Izglednost $p(D \theta)$ | Apriorna $p(\theta)$ |
|----------------------------|--------------------------|----------------------|
| Beta                       | Bernoulli                | Beta                 |
| Dirichlet                  | Multinuli                | Dirichlet            |
| Normal                     | Normal                   | Normal               |
| Multivariate normal        | Multivariate normal      | Multivariate normal  |

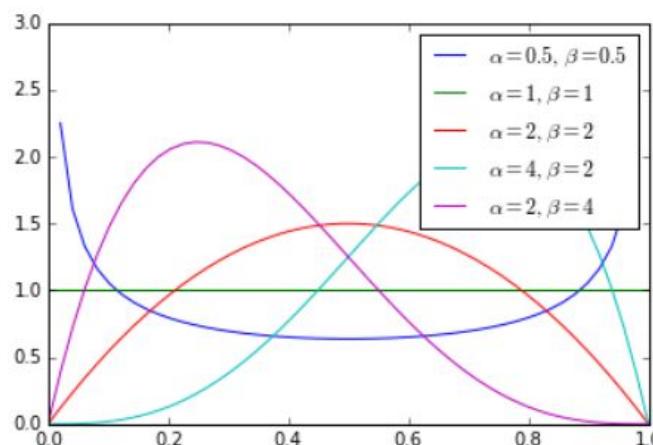
Pored analitičke maksimizacije, konjugatne distribucije omogućuju učenje s postepenim dolaskom novih podataka (**online learning**). Ako su aposteriorna i apriorna distribucija istog tipa, kad izračunamo aposteriornu distribuciju, nju u idućoj iteraciji, kad dođu novi podatci, možemo koristiti kao novu apriornu distribuciju.

## 7. Izvesti Beta-Bernoullijev model i definirati Laplaceovo zaglađivanje.

**Beta-Bernoullijev model** koristi beta distribuciju za modeliranje apiorne distribucije  $p(\mu)$ :

$$p(\mu|\alpha, \beta) = \frac{1}{B(\alpha, \beta)} \mu^{\alpha-1} (1-\mu)^{\beta-1}$$

Gdje je  $B$  beta-funkcija koja osigurava da je integral gustoće jednak 1, a  $\alpha$  i  $\beta$  parametri distribucije (oba moraju biti strogo pozitivni). Izgled distribucije u ovisnosti o parametrima:



Za parametre  $\alpha = \beta = 1$  apriorna razdioba ne sadrži nikakvo pozadinsko znanje o vrijednosti parametara.

**Maksimizator beta distribucije (mod)** računa se kao:

$$\frac{\alpha - 1}{\alpha + \beta - 2}$$

uz uvjete  $\alpha > 1$ ,  $\beta > 1$ .

Aposterioru distribuciju dobit ćemo putem Bayesovog teorema. Uz funkciju izglednosti Bernoullijeve varijable i apriorne vrijednosti definirane beta distribucijom:

$$p(\mathcal{D}|\mu) = \mu^m(1-\mu)^{N-m} \quad p(\mu|\alpha, \beta) = \frac{1}{B(\alpha, \beta)} \mu^{\alpha-1}(1-\mu)^{\beta-1}$$

**aposteriorna vjerojatnost** jednaka je umnošku  $p(D|\mu)$  i  $p(\mu|\alpha, \beta)$ :

$$p(\mu|\mathcal{D}, \alpha, \beta) = \mu^{m+\alpha-1}(1-\mu)^{N-m+\beta-1} \frac{1}{B(\alpha, \beta)p(\mathcal{D})}$$

Dobivena vjerojatnost također je beta distribucija s parametrima:

- $\alpha' = m + \alpha$
- $\beta' = N - m + \beta$
- $B(\alpha', \beta') = B(\alpha, \beta)p(\mathcal{D})$ .

**MAP procjenitelj** za Beta-Bernoullijev model definira se kao:

$$\hat{\mu}_{\text{MAP}} = \frac{\alpha' - 1}{\alpha' + \beta' - 2} = \frac{m + \alpha - 1}{m + \alpha + N - m + \beta - 2} = \frac{m + \alpha - 1}{\alpha + N + \beta - 2}$$

Za puno podataka, u nazivniku dominira  $N$  (procjenitelj je sličniji MLE), dok za malen skup podataka dominiraju  $\alpha$  i  $\beta$  (procjenitelj se više oslanja na apriorno znanje).

U strojnom učenju najčešće se koristi procjenitelj s apriornom razdiobom  $\alpha = \beta = 2$ , koji se naziva **Laplaceov procjenitelj**:

$$\hat{\mu}_{\text{MAP}} = \frac{m + 1}{N + 2}$$

Takav procjenitelj rješava problem prenaučenosti **zaglađivanjem**: u slučaju kad je  $m = 0$ ,  $\mu$  neće biti jednak nuli kao kod MLE, nego malo veći.

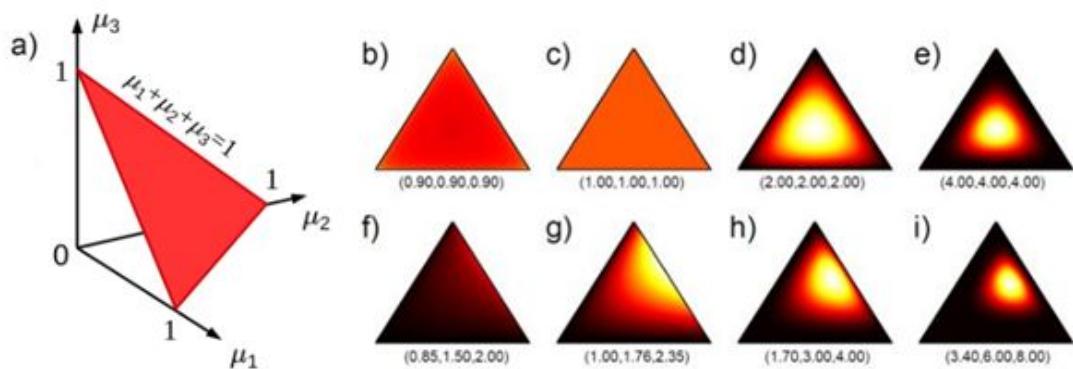
## 8. Objasniti Dirichlet-kategorički model i definirati MAP za kategoričku varijablu.

Dirichletova distribucija konjugatna je multinulijevoj distribuciji. Takav model naziva se **Dirichlet-kategorički model**.

Dirichletova distribucija definirana je kao:

$$P(\boldsymbol{\mu}|\boldsymbol{\alpha}) = P(\mu_1, \dots, \mu_K | \alpha_1, \dots, \alpha_K) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^K \mu_k^{\alpha_k - 1}$$

i predstavlja poopćenje beta distribucija na više  $\mu_k$ . Hiperparametri  $\alpha_k$  određuju vjerojatnost parametara  $\mu_k$ . Budući da suma parametara  $\mu_k$  kojih ukupno ima  $K$  mora biti jednaka 1, ti se parametri nalaze u tzv.  $(K-1)$ -dimenzijskom standardnom simpleksu. Prikaz simpleksa kombinacije  $(\mu_1, \mu_2, \mu_3)$ :



Procjenitelj MAP za parametar  $\mu_k$  jednak je:

$$\hat{\mu}_{k,\text{MAP}} = \frac{\alpha'_k - 1}{\sum_{k=1}^K \alpha'_k - K}$$

Gdje je  $\alpha'_k = N_k + \alpha_k$ , a  $N_k$  predstavlja broj nastupanja  $k$ -te vrijednosti.

## 9. Izračunati MLE/MAP Bernoullieve/kategoričke p{\distribucije za zadani uzorak

Pogledati primjer u domaćoj zadaći. **NEMA PRIMJERA**

# 15 Bayesov klasifikator I

## 1. Napisati pravilo zbroja i umnoška te izvesti Bayesovo pravilo i pravilo lanca.

- **Pravilo zbroja:**

$$P(x) = \sum_y P(x, y)$$

⇒ marginalna vjerojatnost iz zajedničke vjerojatnosti (*joint*)

- **Pravilo umnoška:**

$$P(x, y) = P(y|x)P(x) = P(x|y)P(y)$$

- Dva pravila izvedena iz pravila umnoška:

- **Bayesovo pravilo:**

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

- **Pravilo lanca** (*chain rule*):

$$\begin{aligned} P(x_1, \dots, x_n) &= P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \cdots P(x_n|x_1, \dots, x_{n-1}) \\ &= \prod_{k=1}^n P(x_k|x_1, \dots, x_{k-1}) \end{aligned}$$

⇒ faktorizacija zajedničke vjerojatnosti na umnožak faktora

## 2. Definirati model Bayesovog klasifikatora i navesti nazine pojedinih distribucija.

- Model Bayesovog klasifikatora:

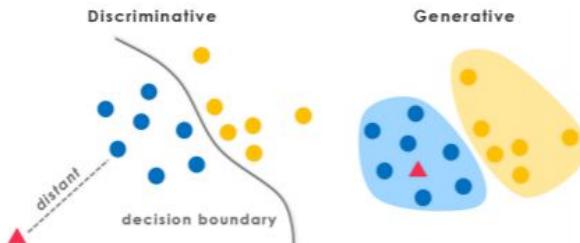
$$h_j(\mathbf{x}; \boldsymbol{\theta}) = P(y = j|\mathbf{x}) = \frac{p(\mathbf{x}|y = j)P(y = j)}{\sum_k p(\mathbf{x}|y = k)P(y = k)}$$

$P(y|X)$  - aposteriorna distribucija. Primjer, kolika je vjerojatnost da pacijent ima rak pluća ako kašlje

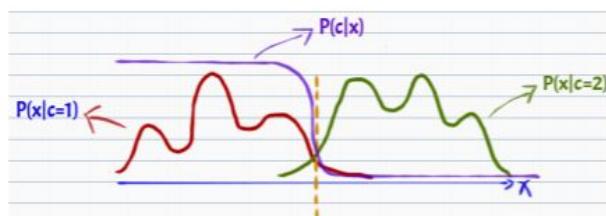
$P(X|y)$  - izglednost klase. Mjeri koliko je dani primjer X karakterističan za klasu y. Primjer, koliko je karakteristično da netko s rakom pluća kašlje.

$P(y)$  - apriori vjerojatnost klase. Primjer, vjerojatnost da netko uopće ima rak pluća.

## 3. Razlikovati generativne i diskriminativne modele te navesti prednosti i nedostatke.



- Prednosti: laka ugradnja stručnog znanja, interpretabilnost/analiza rezultata
- Nedostatci: iziskuju mnogo primjera za učenje, nepotrebna složenost modeliranja
- Primjer: nepotrebna složenost modeliranja zajedničke vjerojatnosti:



Pojašnjenje primjera, nama samo treba granica za odvajanje dvaju klasa i modeliranje cijele vjerojatnosti je overkill

#### 4. Napisati generativnu priču za model Bayesovog klasifikatora.

- Modeliraju **nastajanje podataka**  $\{(\mathbf{x}^{(i)}, y^{(i)})\}_i$  – tzv. **generativna priča**
- Generativna priča Bayesovog klasifikatora:

$$P(\mathbf{x}, y) = p(\mathbf{x}|y)P(y)$$

$\Rightarrow$  odabir oznake prema  $P(y)$ , zatim odabir primjera prema  $P(\mathbf{x}|y)$

#### 5. Definirati uni/multivarijatni Gaussov Bayesov klasifikator.

**Univarijatni Gaussov Bayesov klasifikator:**

$$h_j(x) = -\ln \sigma_j - \frac{(x - \mu_j)^2}{2\sigma_j^2} + \ln P(y = j)$$

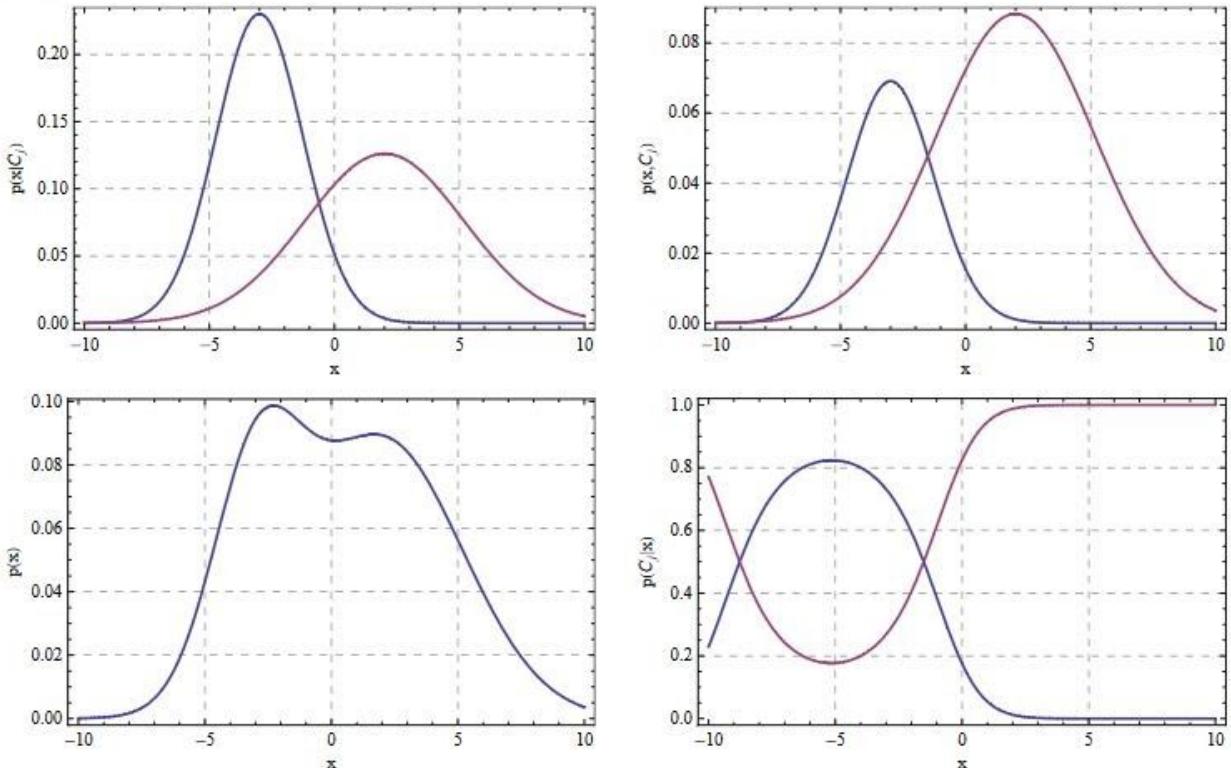
**Multivarijatni Gaussov Bayesov klasifikator:**

$$\begin{aligned} h_j(\mathbf{x}) &= \ln p(\mathbf{x}|y = j) + \ln P(y = j) \\ &= -\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_j| - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j) + \ln P(y = j) \\ &\Rightarrow -\frac{1}{2} \ln |\Sigma_j| - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j) + \ln P(y = j) \end{aligned}$$

#### 6. Skicirati gustoće vjerojatnosti univarijatnog Gaussovog Bayesovog klasifikatora.

$$p(x|y = 1) \sim \mathcal{N}(-3, 3), P(y = 1) = 0.3$$

$$p(x|y = 2) \sim \mathcal{N}(2, 10), P(y = 2) = 0.7$$



## 7. Izvesti i skicirati tri pojednostavljene inačice Gaussovog Bayesovog klasifikatora.

→ Dijeljena kovarijacijska matrica

$$\hat{\Sigma} = \sum_j \hat{\mu}_j \hat{\Sigma}_j$$

$$\begin{aligned} h_j(\mathbf{x}) &= \ln p(\mathbf{x}|y=j) + \ln P(y=j) \\ &= -\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} (\mathbf{x}^T \Sigma^{-1} \mathbf{x} - 2\mathbf{x}^T \Sigma^{-1} \boldsymbol{\mu}_j + \boldsymbol{\mu}_j^T \Sigma^{-1} \boldsymbol{\mu}_j) + \ln P(y=j) \end{aligned}$$

→ Dijeljena i dijagonalna kovarijacijska matrica

$$\Sigma = \text{diag}(\sigma_i^2)$$

- Vrijedi  $|\Sigma| = \prod_i \sigma_i^2$  i  $\Sigma^{-1} = \text{diag}(1/\sigma_i^2)$
- Izglednost klase:

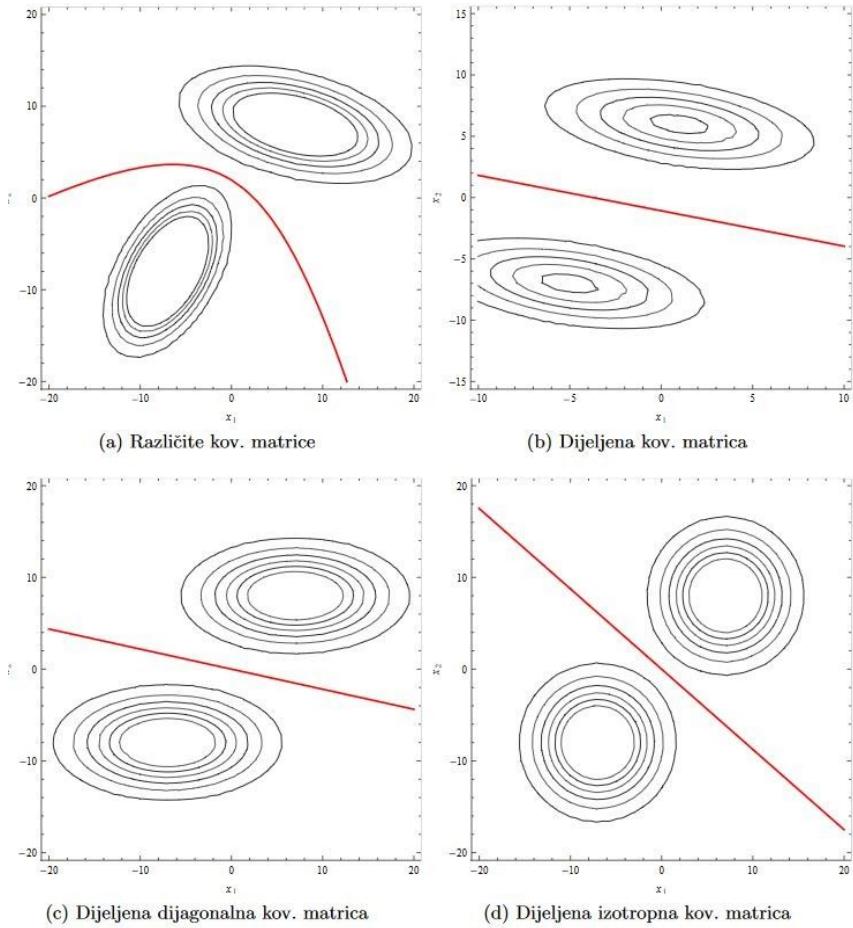
$$\begin{aligned} p(\mathbf{x}|y=j) &= \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_j)^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_j) \right) \\ &= \frac{1}{(2\pi)^{n/2} \prod_{i=1}^n \sigma_i} \exp \left( -\frac{1}{2} \sum_{i=1}^n \left( \frac{x_i - \mu_{ij}}{\sigma_i} \right)^2 \right) \\ &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_i} \exp \left\{ -\frac{1}{2} \left( \frac{x_i - \mu_{ij}}{\sigma_i} \right)^2 \right\} \\ &= \prod_{i=1}^n \mathcal{N}(\mu_{ij}, \sigma_i^2) = \prod_{i=1}^N p(x_i|y) \end{aligned}$$

$$\begin{aligned} h_j(\mathbf{x}) &= \ln p(\mathbf{x}|y=j) + \ln P(y=j) \\ &= \sum_{i=1}^n \ln \frac{1}{\sqrt{2\pi}\sigma_i} + \sum_{i=1}^n \left( -\frac{1}{2} \left( \frac{x_i - \mu_{ij}}{\sigma_i} \right)^2 \right) + \ln P(y=j) \end{aligned}$$

→ Izotropna kovarijacijska matrica

$$\Sigma = \sigma^2 \mathbf{I}$$

$$h_j(\mathbf{x}) = -\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu_{ij})^2 + \ln P(y=j)$$



## 8. \*Izvesti (ne)linearnost granice za danu inačicu Gaussovog Bayesovog klasifikatora.

$$p(\mathbf{x}|y=j) = \frac{1}{(2\pi)^{n/2} |\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)\right)$$

Nelinearnost:

- Granica izmedu dviju klasa:  $h_1(\mathbf{x}) - h_2(\mathbf{x}) = 0$ :

$$\begin{aligned} h_{12}(\mathbf{x}) &= h_1(\mathbf{x}) - h_2(\mathbf{x}) \\ &= -\frac{1}{2} \ln |\Sigma_1| - \frac{1}{2} (\mathbf{x}^T \Sigma_1^{-1} \mathbf{x} - 2\mathbf{x}^T \Sigma_1^{-1} \boldsymbol{\mu}_1 + \boldsymbol{\mu}_1^T \Sigma_1^{-1} \boldsymbol{\mu}_1) + \ln P(y=1) \\ &\quad - \left( -\frac{1}{2} \ln |\Sigma_2| - \frac{1}{2} (\mathbf{x}^T \Sigma_2^{-1} \mathbf{x} - 2\mathbf{x}^T \Sigma_2^{-1} \boldsymbol{\mu}_2 + \boldsymbol{\mu}_2^T \Sigma_2^{-1} \boldsymbol{\mu}_2) + \ln P(y=2) \right) \\ &\quad \dots \mathbf{x}^T (\Sigma_1^{-1} - \Sigma_2^{-1}) \mathbf{x} \dots \end{aligned}$$

$\Rightarrow$  član koji kvadratno ovisi o  $\mathbf{x} \Leftrightarrow$  nelinearna granica

## 9. Izračunati broj parametara za danu inačicu Gaussovog Bayesovog klasifikatora.

Multivarijatni:

Broj parametara:  $\frac{n}{2}(n+1)K + K \cdot n + K - 1 \Rightarrow \mathcal{O}(n^2)$

Dijeljena kovarijacijska matrica:

Broj parametara:  $\frac{n}{2}(n+1) + nK + K - 1 \Rightarrow \mathcal{O}(n^2)$

**Dijeljena i dijagonalna kovarijacijska matrica:**

Broj parametara:  $n + n \cdot K + K - 1 \Rightarrow \mathcal{O}(n)$

**Izotropna kovarijacijska matrica:**

Broj parametara:  $1 + Kn + K - 1 \Rightarrow \mathcal{O}(n)$

## 10. Procijeniti parametre Gaussovog Bayesovog klasifikatora na skupu podataka.

**Multivariatni:**

- MLE procjene parametara:

$$\begin{aligned}\hat{\mu}_j &= \frac{1}{N_j} \sum_{i=1}^N \mathbf{1}\{y^{(i)} = j\} \mathbf{x}^{(i)} \\ \hat{\Sigma}_j &= \frac{1}{N_j} \sum_{i=1}^N \mathbf{1}\{y^{(i)} = j\} (\mathbf{x}^{(i)} - \hat{\mu}_j)(\mathbf{x}^{(i)} - \hat{\mu}_j)^T \\ \hat{\mu}_j &= \frac{1}{N} \sum_{j=1}^N \mathbf{1}\{y^{(i)} = j\} = \frac{N_j}{N}\end{aligned}$$

**Univariatni:**

- MLE procjene parametara:

$$\begin{aligned}\hat{\mu}_j &= \frac{1}{N_j} \sum_{i=1}^N \mathbf{1}\{y^{(i)} = j\} x^{(i)} \\ \hat{\sigma}_j^2 &= \frac{1}{N_j} \sum_{i=1}^N \mathbf{1}\{y^{(i)} = j\} (x^{(i)} - \hat{\mu}_j)^2 \\ P(y = j) &= \hat{\mu}'_j = \frac{1}{N} \sum_{j=1}^N \mathbf{1}\{y^{(i)} = j\} = \frac{N_j}{N}\end{aligned}$$

# 16 Bayesov klasifikator II

## 1. Objasniti i izvesti vezu između modela logističke regresije i Bayesovog klasifikatora.

### 1 Bayesov klasifikator vs. logistička regresija

- Ideja: pokazati da logistička regresija i Bayesov klasifikator izračunavaju isti  $P(y|\mathbf{x})$

- Model **logističke regresije**:

$$h(\mathbf{x}; \mathbf{w}) = P(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$$

- Aposteriorna vjerojatnost za **kontinuirani Bayesov klasifikator** (za dvije klase):

$$\begin{aligned} P(y = 1|\mathbf{x}) &= \frac{p(\mathbf{x}|y = 1)P(y = 1)}{p(\mathbf{x}|y = 1)P(y = 1) + p(\mathbf{x}|y = 2)P(y = 2)} = \frac{1}{1 + \frac{p(\mathbf{x}|y = 2)P(y = 2)}{p(\mathbf{x}|y = 1)P(y = 1)}} = \\ &= \frac{1}{1 + \exp\left(\ln \frac{p(\mathbf{x}|y = 2)P(y = 2)}{p(\mathbf{x}|y = 1)P(y = 1)}\right)} = \frac{1}{1 + \exp(-\alpha)} = \sigma(\alpha) \end{aligned}$$

gdje

$$\alpha = \ln \frac{p(\mathbf{x}|y = 1)P(y = 1)}{p(\mathbf{x}|y = 2)P(y = 2)} = \underbrace{\ln p(\mathbf{x}|y = 1)P(y = 1)}_{h_1(\mathbf{x})} - \underbrace{\ln p(\mathbf{x}|y = 2)P(y = 2)}_{h_2(\mathbf{x})}$$

- Možemo li  $\alpha$  prikazati kao linearnu kombinaciju težina,  $\alpha = \mathbf{w}^T \mathbf{x}$ ?
- Da, ako prepostavimo **dijeljenu kovarijacijsku matricu**:

$$\begin{aligned} \alpha &= h_1(\mathbf{x}) - h_2(\mathbf{x}) \\ &= \mathbf{x}^T \underbrace{\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)}_{\mathbf{w}} - \underbrace{\frac{1}{2}\boldsymbol{\mu}_1^T \Sigma^{-1} \boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T \Sigma^{-1} \boldsymbol{\mu}_2 + \ln \frac{P(y = 1)}{P(y = 2)}}_{w_0} = \mathbf{w}^T \mathbf{x} + w_0 \end{aligned}$$

⇒ logistička regresija istovjetna je Bayesovom klasifikatoru s dijeljenom  $\Sigma$

## 2. Izvesti model naivnog Bayesovog klasifikatora počevši od nefaktorizirane izglednosti.

- Prepostavka: u svakoj klasi, svaka značajka uvjetno je nezavisna od svih drugih:

$$x_k \perp (x_1, \dots, x_{k-1})|y \Leftrightarrow P(x_k|x_1, \dots, x_{k-1}, y) = P(x_k|y)$$

- Faktorizacija uz tu prepostavku:

$$P(x_1, \dots, x_n|y) = \prod_{k=1}^n P(x_k|x_1, \dots, x_{k-1}, y) = \prod_{k=1}^n P(x_k|y)$$

- Naivan Bayesov klasifikator** (*Naïve Bayes classifier*):

$$h(x_1, \dots, x_n) = \operatorname{argmax}_j P(y = j) \prod_{k=1}^n P(x_k|y = j)$$

Induktivna pristranost

## 3. Definirati uvjetnu nezavisnost slučajnih varijabli i objasniti ju na primjeru.

- Uvjetna nezavisnost**  $X$  i  $Y$  uz dani  $Z$  – notacija:  $X \perp Y | Z$ :

$$P(X, Y | Z) = P(X|Z)P(Y|Z)$$

$$P(X|Y, Z) = P(X|Z)$$

$$P(Y|X, Z) = P(Y|Z)$$

#### **4. Objasniti potrebu za uvođenjem pretpostavke o uvjetnoj nezavisnosti značajki.**

Previše parametara i manjak generalizacije (“prone to overfitting”)

#### **5. Objasniti polunaivan Bayesov klasifikator i objasniti potrebu za takvim modelom.**

Ponekad je pretpostavka o uvjetnoj nezavisnosti između svih varijabli zaista kriva i onda te varijable ne isfaktoriziramo nego ih modeliramo zajednički. No takav model je onda opet složeniji, ima vise parametara, a i broj mogućih združivanja baš jako raste s brojem varijabli.

#### **6. Definirati KL-divergenciju i uzajamnu informaciju.**

$$D_{\text{KL}}(P||Q) = - \sum_x P(x) \ln \frac{P(x)}{Q(x)}$$

KL divergencija je mjera udaljenosti između dvije distribucije. Mjeri koliko smo prosječno sfulali ako pretpostavimo da je nešto napravljeno po distribuciji P, a zapravo je po Q (koliko cemo informacije potrošiti više). Tehnički, nije prava mjera udaljenosti jer  $D(P, Q) \neq D(Q, P)$  ali svejedno mjeri pogrešku između distribucija.

$$I(x, y) = D_{\text{KL}}(P(x, y) || P(x)P(y)) = \sum_{x,y} P(x, y) \ln \frac{P(x, y)}{P(x)P(y)}$$

Uzajamna informacija (transinformacija) je količina informacije koja jedna slučajna varijabla daje o drugoj. Računa se kao KL udaljenost zajedničke distribucije  $P(x, y)$  od nezavisne distribucije  $P(x)P(y)$ . Što je udaljenost veća, veća je pogreška koju radimo pretpostavljajući da su te dvije distribucije nezavisne, odnosno drugim riječima, stupanj zavisnosti je sve veći.

#### **7. Navesti moguće kriterije za odabir modela kod polunaivnog Bayesovog klasifikatora.**

FSSJ - uzima varijablu po varijablu te gleda točnost modela ako a) ju dodamo uvjetno nezavisno b) združimo ju s nekom od prethodnih varijabli, pa uzmemo ono što daje točniji model.

TAN, k-DB - procjena zavisnosti varijabli (nije opisano).

q

# 17 Probabilistički grafički modeli I

## 1. Navesti tri aspekta probabilističkih grafičkih modelova.

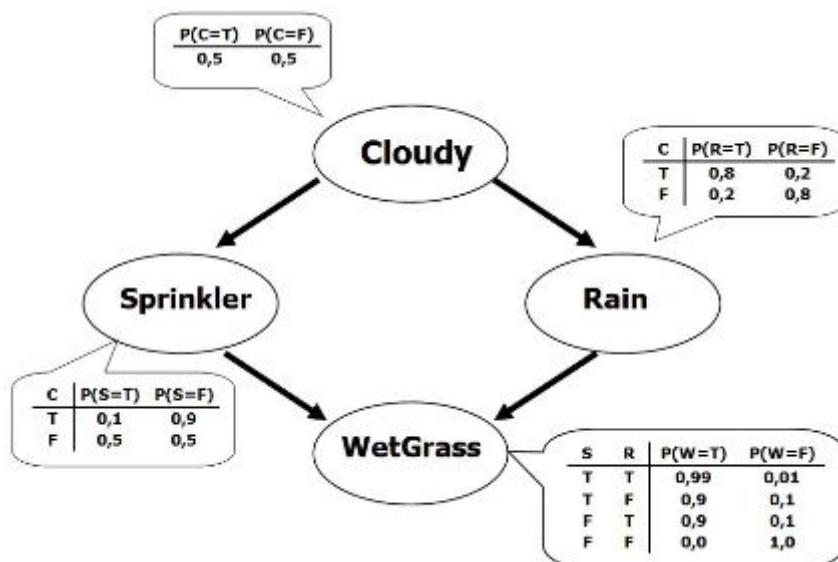
- Tri aspekta PGM-a: (1) **reprezentacija**, (2) **zaključivanje** i (3) **učenje**
- Reprezentacija:
  - usmjereni aciklički graf  $\Rightarrow$  **Bayesove mreže**
  - neusmjereni graf  $\Rightarrow$  **Markovljeve mreže**
- Zaključivanje – određivanje vrijednosti nepažanih varijabli na temelju opažanih
- Učenje – procjena parametara ili učenje strukture mreže na temelju podatka

## 2. Definirati Bayesovu mrežu.

Usmjereni aciklički graf. Čvorovi su slučajne varijable. Bridovi su uvjetne zavisnosti.

## 3. Napisati faktorizaciju zajedničke distribucije na temelju Bayesove mreže i obratno.

Najčešći primjer ikad:



$$p(c, s, r, w) = p(c)p(s|c)p(r|c)p(w|s, r)$$

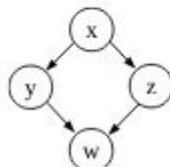
## 4. Definirati uređajno Markovljevo svojstvo i kako ono uvjetuje bridove u mreži.

- **Uredajno Markovljevo svojstvo** (UMS): svaki čvor  $x_k$  ovisi samo o roditeljima:

$$x_k \perp \text{pred}(x_k) \setminus \text{pa}(x_k) \mid \text{pa}(x_k)$$

gdje je  $\text{pred}(x_k)$  skup prethodnika čvora  $x_k$  po topološkom uređaju

- Primjer:  $p(x, y, z, w) = p(x)p(y|x)p(z|x)p(w|y, z)$



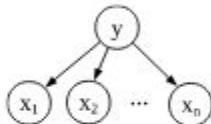
## 5. Izvesti uvjetne nezavisnosti iz dane faktorizacije zajedničke distribucije.

Ima u zadaći primjera

## 6. Nacrtati Bayesovu mrežu polu/naivnog Bayesovog klasifikatora.

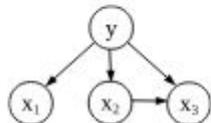
- Naivan Bayesov klasifikator:

$$P(\mathbf{x}, y) = P(y) \prod_{i=1}^n P(x_i|y)$$



- Polunaivan Bayesov klasifikator. Npr.:

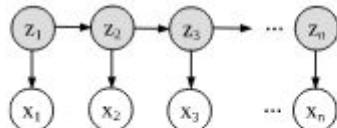
$$P(x_1, x_2, x_3, y) = P(x_1|y)P(x_2|x_3|y)P(y) = P(x_1|y)P(x_2|y)P(x_3|x_2,y)P(y)$$



## 7. Objasniti, definirati i skicirati skriveni Markovljev model (HMM).

- Skriveni Markovljev model (*Hidden Markov Model, HMM*):

$$p(\mathbf{x}, \mathbf{z}) = p(z_1)p(x_1|z_1) \prod_{k=2}^n p(z_k|z_{k-1})p(x_k|z_k)$$



⇒ indirektno modelira dulje zavisnosti preko skrivenih varijabli  $\mathbf{z}$

Latentne varijable  $z_i$  su skrivene i njih ne možemo izravno opaziti. To skriveno stanje je zatim odgovorno za varijable koje možemo opaziti.

## 8. \*Izvesti tri pravila d-odvojivosti za stazu od tri čvora.

1. Račvanje: ako z uvjetuje  $x$  i  $y$ , i opazili smo  $z$ , onda su  $x$  i  $y$  nezavisni. U suprotnom, zavisni su.
2. Lanac. Ako  $x$  uvjetuje  $z$ , i  $z$  uvjetuje  $y$ , te smo opazili  $z$ , onda su  $x$  i  $y$  nezavisni. U suprotnom, zavisni su.
3. Sraz. Ako je  $z$  uvjetovan od  $x$  i od  $y$ , te smo opazili  $z$ , onda su  $x$  i  $y$  zavisni. U suprotnom, nezavisni su.

## 9. \*Iščitati uvjetne nezavisnosti varijabli u Bayesovoj mreži korištenjem d-odvojivosti.

Stvarni život.

## 10. Objasniti efekt objašnjavanja te ga ilustrirati na primjeru.

Efekt objasnjavanja je situacija u kojoj se dvije varijable u srazu natječu da objasne treću varijablu. Kao primjer treba uzeti  $P(S|W, R)$  i  $P(R|W, S)$  sa primjera iz 3. ishoda i vidjeti kako se mijenjaju vjerojatnosti za  $S$  ako je  $R=0,1$  ili za  $R$  ako je  $S=0,1$ .  $W$  postaviti proizvoljno. Ispast će da će smanjenje vjerojatnosti jedne varijable uzrokovati povećanje druge varijable.

# 18 Probabilistički grafički modeli II

## 1. Razlikovati opažene i skrivene varijable te varijable upita i varijable smetnje.

- Opažene varijable su one kojima znamo vrijednost.
- Skrivene varijable su one kojima ne znamo vrijednost.
- Varijable upita su varijable koje su skrivene, ali zelimo znati njihovu distribuciju
- Varijable smetnje su skrivene varijable za koje nas uopće nije briga (samo smetaju)

## 2. Razlikovati posteriorne i MAP-upite.

- Posteriorni upiti vraćaju nazad razdiobu neke slučajne varijable. Pitanje koje se pitamo je "koja je distribucija vjerojatnosti nekog  $X$  ako znamo neki  $Y$ ".
- MAP (maximum a posteriori; a.k.a. MPE, most probable explanation) vraća vrijednost koja ima najveću vjerojatnost. Nas nije briga koja je vjerojatnost toga - bitno nam je samo za koju vrijednost distribucija ima maksimum.

## 3. Egzaktnim zaključivanjem izračunati posteriorni/MAP-upit na danoj Bayesovoj mreži.

Stvarni život. Primjer u zadaćama.

## 4. \*Objasniti ideju postupka eliminacije varijabli.

Nekoliko je vrsta eliminacija. Ili se prvo generira zajednicka distribucija (koja se zatim marginalizira) ili se tijekom racunanja "cacheaju" rezultati koje onda ne moramo ponovo racunat. Eliminacija varijabli u HMM zove se forward-backward algoritam, a varijanta za MAP je Viterbijev algoritam.

## 5. Objasniti zaključivanje unaprijednim uzorkovanjem i uzorkovanjem s odbijanjem.

Egzaktno zaključivanje je naporno - velik je broj svih kombinacija koje se mogu napraviti. Ali, PGMovi su generativni modeli. Njih možemo natjerati da generiraju vrijednosti pomoću slučajnog procesa i onda gledati koja je distribucija generiranih vrijednosti (kao da smo umjetno generirali dataset). To je puno komputacijski lakše.

Unaprijedno uzorkovanje ide redom po mreži i uzorkuje prvo roditelje, a onda varijable koje ovise o tim roditeljima, i tako dalje dok ne uzorkujemo sve. Ovo ponovimo puno puta i imamo uzorak kojem onda možemo izračunati vjerojatnost.

Uzorkovanje s odbijanjem je slično unaprijednom, samo trebamo odbiti sve one vektore koji ne odgovaraju uvjetima. To je malo problem ako je vjerojatnost uvjeta dost mala.

## 6. Objasniti zaključivanje uzorkovanjem i Gibbsovo uzorkovanje.

Gibbsovo uzorkovanje krene sa slučajnim vektorom koji predstavlja varijable i onda svaku komponentu tog vektora uzorkujemo u ovisnosti o svim drugim i tako dođemo do finalnog nasumičnog vektora. Ako to ponovimo puno puta, dobit ćemo uzorak.

## 7. Izvesti log-izglednost Bayesove mreže te MLE/MAP procjenitelje.

- Log-izglednost za općenitu Bayesovu mrežu:

$$\begin{aligned}
 \ln \mathcal{L}(\boldsymbol{\theta} | \mathcal{D}) &= \ln p(\mathcal{D} | \boldsymbol{\theta}) = \ln p(\mathbf{x}_1, \dots, \mathbf{x}_n) = \ln \prod_{i=1}^N p(\mathbf{x}_i | \boldsymbol{\theta}) \\
 &= \ln \prod_{i=1}^N \prod_{k=1}^n p(x_k | \text{pa}(x_k), \boldsymbol{\theta}_k) = \ln \prod_{k=1}^n \prod_{i=1}^N p(x_k | \text{pa}(x_k), \boldsymbol{\theta}_k) \\
 &= \sum_{k=1}^n \sum_{i=1}^N \ln p(x_k | \text{pa}(x_k), \boldsymbol{\theta}_k)
 \end{aligned}$$

- MLE procjena za  $k$ -ti čvor:

$$\boldsymbol{\theta}_k^* = \underset{\boldsymbol{\theta}_k}{\operatorname{argmax}} \sum_{i=1}^N \ln p(x_k | \text{pa}(x_k), \boldsymbol{\theta}_k)$$

- MAP procjena za  $k$ -ti čvor:

$$\boldsymbol{\theta}_k^* = \underset{\boldsymbol{\theta}_k}{\operatorname{argmax}} \left( \sum_{i=1}^N \ln p(x_k | \text{pa}(x_k), \boldsymbol{\theta}_k) + \ln p(\boldsymbol{\theta}_k) \right)$$

- MAP-procjena za kategorijsku razdiobu (Dirichlet-kategorijski model uz  $\alpha = 2$ ):

$$\begin{aligned}
 \hat{\mu}_{k,j,l} &= \frac{N_{kjl} + 1}{N_{kj} + K_k} \\
 N_{kjl} &= \sum_{i=1}^N \mathbf{1}\{\mathbf{x}_{\text{pa}(x_k)}^{(i)} = j \wedge x_k^{(i)} = l\} \\
 N_{kj} &= \sum_l N_{kjl}
 \end{aligned}$$

gdje je  $K_k$  broj mogućih vrijednosti varijable  $x_k$

## 8. Procjeniti parametre Bayesove mreže na danom skupu podataka.

Stvarni život. Primjer možda postoji u zadaćama.

# 19 Grupiranje I

## 1. Definirati problem grupiranja.

Razdjeljivanje primjera u grupe (klastere) tako da su slični primjeri skupa.

## 2. Razlikovati partijsko i hijerarhijsko grupiranje te čvrsto i meko grupiranje.

- Čvrsto grupiranje je kad jedan primjer može biti u samo jednoj grupi.
- Meko grupiranje je kad jedan primjer može biti u više grupa.
- Hijerarhijsko grupiranje je kad se grupe mogu razložiti na podgrupe itd.
- Particijsko grupiranje je kad nemamo podgrupe

## 3. Napisati kriterijsku funkciju algoritma k-sredina i izvesti iterativnu optimizaciju.

- **Funkcija pogreške** (kriterijska funkcija):

$$J = \sum_{k=1}^K \sum_{i=1}^N b_k^{(i)} \| \mathbf{x}^{(i)} - \boldsymbol{\mu}_k \|^2$$

gdje je  $\boldsymbol{\mu}_k$  centroid  $k$ -te grupe, a  $b_k^{(i)}$  indikatorska varijabla pripadnosti  $\mathbf{x}^{(i)}$  grupe  $k$

- Svaki primjer  $\mathbf{x}^{(i)}$  svrstavamo u grupu s njemu najbližim centroidom  $\boldsymbol{\mu}_k$ :

$$b_k^{(i)} = \begin{cases} 1 & \text{ako } k = \operatorname{argmin}_j \| \mathbf{x}^{(i)} - \boldsymbol{\mu}_j \|^2 \\ 0 & \text{inače} \end{cases}$$

- Tražimo grupiranje  $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K$  koje minimizira pogrešku:  $\operatorname{argmin}_{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K} J$
- Analitička minimizacija nije moguća jer su  $b_k^{(i)}$  i  $\boldsymbol{\mu}_k$  međuvisni
- Alternativa: **iterativna optimizacija**

- Fiksiramo  $\boldsymbol{\mu}_k$  na neke inicijalne vrijednosti
- Pridružimo primjere grupama (izračunamo  $b_k^{(i)}$  za  $i = 1, \dots, N$ )
- Uz fiksne  $b_k^{(i)}$ , minimizacija  $J$  daje formulu za ažuriranje centroida:

$$\nabla_{\boldsymbol{\mu}_k} J = \mathbf{0} \quad \Rightarrow \quad 2 \sum_{i=1}^N b_k^{(i)} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_k) = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{\mu}_k = \frac{\sum_i b_k^{(i)} \mathbf{x}^{(i)}}{\sum_i b_k^{(i)}}$$

- Ponavljamo do konvergencije  $\boldsymbol{\mu}_k$  odnosno  $b_k^{(i)}$

## 4. Napisati pseudokod algoritma k-sredina i primijeniti ga na dane podatke.

### Algoritam K-sredina (k-means algorithm)

```
1: inicijaliziraj centroide  $\boldsymbol{\mu}_k$ ,  $k = 1, \dots, K$ 
2: ponavljam
3:   za svaki  $\mathbf{x}^{(i)} \in \mathcal{D}$ 
4:      $b_k^{(i)} \leftarrow \begin{cases} 1 & \text{ako } k = \operatorname{argmin}_j \| \mathbf{x}^{(i)} - \boldsymbol{\mu}_j \|^2 \\ 0 & \text{inače} \end{cases}$ 
5:   za svaki  $\boldsymbol{\mu}_k$ ,  $k = 1, \dots, K$ 
6:      $\boldsymbol{\mu}_k \leftarrow \sum_{i=1}^N b_k^{(i)} \mathbf{x}^{(i)} / \sum_{i=1}^N b_k^{(i)}$ 
7: dok  $\boldsymbol{\mu}_k$  ne konvergiraju
```

## 5. Objasniti konvergenciju i neoptimalnost algoritma k-sredina.

Algoritam konvergira - broj konfiguracija je konačan, a  $J$  nužno pada kroz iteracije, dakle algoritam će nužno naći (neki) minimum.

Optimalnost - algoritam pohlepno pretražuje te nalazi lokalni minimum. Optimalnost ovisi o početnim središtima.

## 6. Objasniti najvažnije načine odabira početnih središta.

- Nasumičan odabir iz cijelog vektorskog prostora.
- Nasumičan odabir nekih primjera
- K slučajnih vektora nadodanih centroidu cijelog dataseta.
- nesto sjeban sa PCA nemam pojma
- kmeans++ - vjerojatnost odabira nekog primjera kao novog središta raste s kvadratom udaljenosti od već odabranih središta. (dosta cool)

## 7. Napisati pseudokod algoritma k-medoida i primijeniti ga na dane podatke.

- Prototipi grupa nisu centroidi nego **medoidi** (odabrani primjeri u svakoj grupi)
- Funkcija pogreške:

$$J = \sum_{i=1}^N \sum_{k=1}^K b_k^{(i)} \nu(\mathbf{x}^{(i)}, \boldsymbol{\mu}_k)$$

gdje je  $\nu : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  općenita **mjera različitosti** dvaju primjera

- Tipična izvedba je **algoritam PAM** (*partitioning around medoids*)

### Algoritam PAM

```
1:  inicijaliziraj medoide  $\mathcal{M} = \{\boldsymbol{\mu}_k\}_{k=1}^K$  na odabране  $\mathbf{x}^{(i)}$ 
2:  ponavljaj
3:    za svaki  $\mathbf{x}^{(i)} \in \mathcal{D} \setminus \mathcal{M}$ 
4:       $b_k^{(i)} \leftarrow \begin{cases} 1 & \text{ako } k = \operatorname{argmin}_j \nu(\mathbf{x}^{(i)}, \boldsymbol{\mu}_j) \\ 0 & \text{inače} \end{cases}$ 
5:    za svaki  $\boldsymbol{\mu}_k \in \mathcal{M}$ 
6:       $\boldsymbol{\mu}_k \leftarrow \operatorname{argmin}_{\boldsymbol{\mu}_j \in \mathcal{D} \setminus \mathcal{M} \cup \{\boldsymbol{\mu}_k\}} \sum_i b_k^{(i)} \nu(\mathbf{x}^{(i)}, \boldsymbol{\mu}_j)$ 
7:  dok  $\boldsymbol{\mu}_k$  ne konvergiraju
```

## 8. Navesti glavne razlike algoritma k-sredina i algoritma k-medoida.

| K-sredina                               | K-medoida                                     |
|-----------------------------------------|-----------------------------------------------|
| Grupira primjere iz vektorskog prostora | Grupira primjere koji nisu nuzno iz vek.pros. |
| Udaljenost je euklidska                 | Udaljenost je bilo koja mjera različitosti    |
| Algoritam manje složenosti              | Algoritam veće složenosti                     |

## 9. Objasniti osnovne pristupe određivanju broja grupa.

Ne možemo jednostavno minimizirati kriterijsku funkciju jer ona ima trivijalni minimum 0 za  $K = \text{broj primjera}$ . Moramo ići pametnije:

Metoda laka - gleda se kad smanjenje kriterijske funkcije postane marginalno.

“Regularizirana” minimizacija - kažnjavamo velike brojeve grupa.

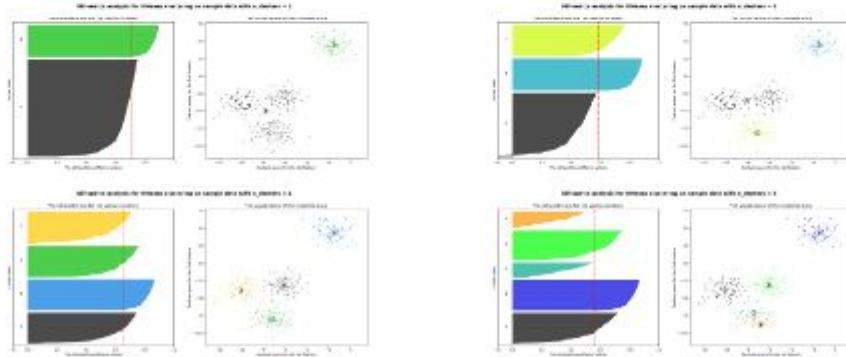
Silueta:

- **Analiza siluete** (*silhouette analysis*):

- Silueta primjera  $\mathbf{x}^{(i)}$ :

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \in [-1, +1]$$

- $a(i)$  i  $b(i)$  su prosjek udaljenost od  $\mathbf{x}^{(i)}$  do primjera iste odnosno najbliže grupe
  - Računamo i grafički prikazujemo  $s(i)$  za sve primjere svake grupe
  - Loše grupiranje: ispodprosječne siluete nekih grupa i/ili visoka varijanca silueta
  - Primjer (scikit-learn):



## 10. Definirati Randov indeks i izračunati ga na danom primjeru.

Mjera točnosti grupiranja nad parovima

- **Randov indeks** – točnost grupiranja na razini parova primjera:

$$R = \frac{a + b}{\binom{N}{2}} \in [0, 1]$$

- $a$  – broj jednakoznačenih parova u istim grupama
  - $b$  – broj različito označenih parova u različitim grupama
  - Optimalan  $K$  je onaj koji maksimizira  $R(K)$

# 20 Grupiranje II

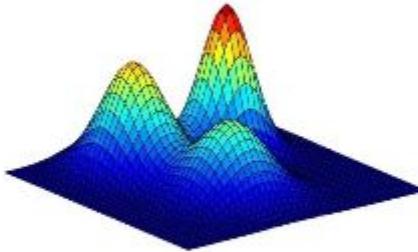
## 1. Definirati model miješane gustoće te model Gaussove mješavine (MoG).

- Model miješane gustoće je linearna kombinacija  $K$  komponenti:

$$p(\mathbf{x}) = \sum_{k=1}^K p(\mathbf{x}, y=k) = \sum_{k=1}^K P(y=k)p(\mathbf{x}|y=k) = \sum_{k=1}^K \pi_k p(\mathbf{x}|\boldsymbol{\theta}_k)$$

gdje su  $\pi_k$  koeficijenti mješavine, a  $p(\mathbf{x}|\boldsymbol{\theta}_k)$  gustoće komponenti

- Primjer: model bivarijatne Gaussove mješavine s  $K = 3$  grupe:



- Odgovornost možemo izračunati Bayesovim pravilom:

$$h_k^{(i)} = P(y=k|\mathbf{x}^{(i)}) = \frac{P(y=k)p(\mathbf{x}^{(i)}|y=k)}{\sum_j P(y=j)p(\mathbf{x}^{(i)}|y=j)} = \frac{\pi_k p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k)}{\sum_j \pi_j p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_j)}$$

- Parametri modela su  $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\theta}_k\}_{k=1}^K$ ,  $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \Sigma_k)$

## 2. Objasniti što su latentne varijable i koja je njihova uloga u modelu MoG.

Bez latentnih varijabli imamo nepotpunu log-izglednost koja se ne faktorizira po komponentama pa stoga nemamo rješenje u analitičkoj formi. Dodavanjem latentnih varijabli dobivamo potpunu log-izglednost koja se faktorizira po komponentama (jer je model definiran umnoškom a ne sumom, pa zbog svojstva logaritma da umnozak pretvara u sumu logaritam ne ostaje "zaglavljen" kao kod izraza za gradijent nepotpune log-izglednosti). Ukoliko znamo vrijednosti latentnih varijabli (što u praksi ne znamo) postoji analitičko rješenje, no obzirom da ih ne znamo koristimo algoritam maksimizacije očekivanja (koji je btw općenito algoritam koji se općenito koristi za modele s latentnim varijablama)

Latentne varijable opisuju vezu između primjera i grupa: koji primjer pripada kojoj grupi.

## 3. Definirati potpunu i nepotpunu log-izglednost parametara modela MoG.

- Log-izglednost parametara modela (tzv. nepotpuna izglednost):

$$\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \ln \prod_{i=1}^N p(\mathbf{x}^{(i)}) = \ln \prod_{i=1}^N \sum_{k=1}^K \pi_k p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) = \sum_{i=1}^N \ln \sum_{k=1}^K \pi_k p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k)$$

$\Rightarrow$  ne faktorizira se po komponentama  $\Rightarrow$  maksimizacija nema analitičko rješenje

- Log-izglednost parametara modela (tzv. potpuna izglednost):

$$\begin{aligned} \ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}, \mathbf{Z}) &= \ln \prod_{i=1}^N p(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}|\boldsymbol{\theta}) = \ln \prod_{i=1}^N \prod_{k=1}^K \pi_k^{z_k^{(i)}} p(\mathbf{x}|\boldsymbol{\theta}_k)^{z_k^{(i)}} \\ &= \sum_{i=1}^N \sum_{k=1}^K z_k^{(i)} \left( \ln \pi_k + \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) \right) \end{aligned}$$

$\Rightarrow$  ako su  $\mathbf{z}^{(i)}$  poznate, maksimizacija ove log-izglednosti ima analitičko rješenje

- $\mathbf{z}^{(i)}$  su nepoznate, no možemo izračunati očekivanje izglednosti uz fiksirane  $\pi_k$  i  $\boldsymbol{\theta}_k$

#### 4. Objasniti kako radi algoritam maksimizacije očekivanja.

##### Algoritam GMM (model GMM + EM-algoritam)

inicijaliziraj parametre  $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$   
ponavljam do konvergencije log-izglednosti ili parametara

###### E-korak:

Za svaki primjer  $\mathbf{x}^{(i)} \in \mathcal{D}$  i svaku komponentu  $k = 1, \dots, K$ :

$$h_k^{(i)} \leftarrow \frac{p(\mathbf{x}^{(i)} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \pi_k}{\sum_{j=1}^K p(\mathbf{x}^{(i)} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \pi_j}$$

###### M-korak:

Za svaku komponentu  $k = 1, \dots, K$ :

$$\boldsymbol{\mu}_k \leftarrow \frac{\sum_i h_k^{(i)} \mathbf{x}^{(i)}}{\sum_i h_k^{(i)}}, \quad \boldsymbol{\Sigma}_k \leftarrow \frac{\sum_i h_k^{(i)} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_k)(\mathbf{x}^{(i)} - \boldsymbol{\mu}_k)^T}{\sum_i h_k^{(i)}}, \quad \pi_k \leftarrow \frac{1}{N} \sum_{i=1}^N h_k^{(i)}$$

Izračunaj trenutnu vrijednost log-izglednosti

$$\ln \mathcal{L}(\boldsymbol{\theta} | \mathcal{D}) = \sum_{i=1}^N \ln \sum_{k=1}^K \pi_k p(\mathbf{x}^{(i)} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

#### 5. Napisati izraze za E-korak i M-korak algoritma maksimizacije očekivanja za MoG.

- Dva koraka algoritma: E-korak (*expectation*) i M-korak (*maximization*)
- E-korak:** Izračun očekivanja potpune izglednosti uz fiksirane parametre u iteraciji  $t$ :

$$\begin{aligned} \mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(t)}) &= \mathbb{E}_{\mathbf{Z} | \mathcal{D}, \boldsymbol{\theta}^{(t)}} \left[ \sum_{i=1}^N \sum_{k=1}^K z_k^{(i)} (\ln \pi_k + \ln p(\mathbf{x}^{(i)} | \boldsymbol{\theta}_k)) \right] \\ &= \sum_{i=1}^N \sum_{k=1}^K \underbrace{\mathbb{E}[z_k^{(i)} | \mathcal{D}, \boldsymbol{\theta}^{(t)}]}_{= h_k^{(i)}} (\ln \pi_k + \ln p(\mathbf{x}^{(i)} | \boldsymbol{\theta}_k)) \end{aligned}$$

- M-korak:** Izračun parametara za iteraciju  $(t+1)$  koji maksimiziraju očekivanje:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(t)}) &= 0 \\ \nabla_{\pi_k} \left( \sum_{i=1}^N \sum_{k=1}^K h_k^{(i)} \ln \pi_k + \lambda \left( \sum_k \pi_k - 1 \right) \right) &= 0 \quad \Rightarrow \quad \pi_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^N h_k^{(i)} \end{aligned}$$

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_k} \sum_{i=1}^N h_k^{(i)} \ln p(\mathbf{x}^{(i)} | \boldsymbol{\theta}_k) &= 0 \quad \Rightarrow \quad \boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_i h_k^{(i)} \mathbf{x}^{(i)}}{\sum_i h_k^{(i)}} \\ \Rightarrow \quad \boldsymbol{\Sigma}_k^{(t+1)} &= \frac{\sum_i h_k^{(i)} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_k^{(t+1)})(\mathbf{x}^{(i)} - \boldsymbol{\mu}_k^{(t+1)})^T}{\sum_i h_k^{(i)}} \end{aligned}$$

#### 6. Razlikovati aglomerativno i divizivno hijerarhijsko grupiranje.

Aglomerativno hijerarhijsko grupiranje počinje sa grupom za svaki primjer i sjedinjuje najbljiže grupe (po udaljenosti ili sličnosti) u veće grupe, sve dok ne ostane samo 1.

Divizivno kreće od jedne grupe koju onda razdjeljuje na podgrupe i tako dok svaki primjer ne postane svoja grupa.

#### 7. Objasniti i definirati jednostruku, potpunu i prosječnu povezanost grupe.

Sve su mjere povezanosti udaljenost/sličnosti između grupa.

Jednostruka povezanost - najmanja udaljenost između dva primjera u zasebnim grupama.

Potpuna povezanost - najveća udaljenost između dva primjera u zasebnim grupama

Prosječna povezanost - prosječna udaljenost između svih parova primjera, svaki iz svoje grupe

Centroidna povezanost - udaljenost centroida svake grupe.

#### 8. Primjeniti algoritam HAC na dan skup primjera te skicirati pripadni dendrogram.

Stvarni život.

# 21 Vrednovanje modela I

## 1. Izračunati matricu zabune za dani primjer.

Stvarni život

## 2. Izračunati točnost, preciznost, odziv i F1-mjero za danu matricu zabune.

- **Matrica zabune** (*confusion matrix*) – usporeba stvarnih oznaka i predikcija modela

|       |   | Stvarno |    |
|-------|---|---------|----|
|       |   | 1       | 0  |
| Model | 1 | TP      | FP |
|       | 0 | FN      | TN |

TP – true positives, FP – false positives, FN – false negatives, TN – true negatives

- **Točnost** (*accuracy*) je udio točno klasificiranih primjera u skupu svih primjera:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} = 1 - E(h|\mathcal{D})$$

- Ako je udio klase izrazito neuravnotežen, točnost nije indikativna mjera

- **Preciznost** (*precision*):

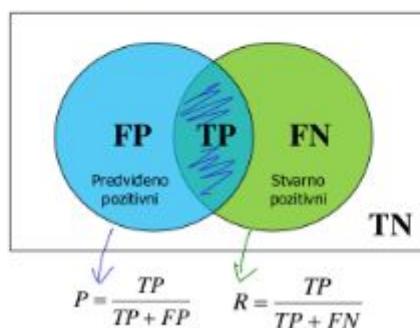
$$P = \frac{TP}{TP + FP}$$

⇒ udio pozitivno klasificiranih primjera u skupu pozitivno klasificiranih primjera

- **Odziv** (*recall, true positive rate, sensitivity*):

$$R = TPR = \frac{TP}{TP + FN}$$

⇒ udio pozitivno klasificiranih primjera u skupu svih pozitivnih primjera



## 3. Objasniti motivaciju za uvođenjem F1-mjere.

Mozda pogresno

Problem sa točnošću i preciznošću je ako se broj primjera dviju klase vrlo razlikuje. F1 mjera uzima i jedno i drugo u obzir

(pitali na labosu pa ako se netko sjeca mozda spamham primjer...)

F1 mjeru uvodimo da bismo postigli "kompromis" izmedju preciznosti i odziva. Mijenjanje klasifikacijskog praga ima suprotan učinak na preciznost (u odnosu na odziv), pa nije moguće mijenjanjem klasifikacijskog praga istovremeno maksimizirati i preciznost i odziv, već tražimo nekakav "soft spot" u kojem je relativno OK vrijednost za obje velicine.

Mjera F1 je harmonijska sredina preciznosti i odziva. U nekim slučajevima želimo više naglasiti preciznost, dok u drugim želimo više naglasiti odziv pa se zbog toga javlja motivacija za uvođenjem **F $\beta$ -mjere**.

<https://www.youtube.com/watch?v=VJBY7bVnnJo>

#### 4. Izračunati mikro- i makro-mjere za danu višeklasnu matricu zabune.

**Makro-prosjek** ( $M$ ): izračun mjere za svaku klasu pa uprosječivanje kroz klase

$$Acc^M = \frac{1}{K} \sum_{j=1}^K Acc_j, \quad P^M = \frac{1}{K} \sum_{j=1}^K P_j, \quad R^M = \frac{1}{K} \sum_{j=1}^K R_j, \quad F_1^M = \frac{1}{K} \sum_{j=1}^K F_{1,j}$$

$\Rightarrow$  jednak utjecaj svih klasa  $\Rightarrow$  loš rezultat na manjim klasama narušava mjeru

**Mikro-prosjek** ( $\mu$ ): zbrajanje matrica pojedinačnih klasa pa izračun mjere

$$TP = \sum_{j=1}^K TP_j, \quad FP = \sum_{j=1}^K FP_j, \quad FN = \sum_{j=1}^K FN_j, \quad TN = \sum_{j=1}^K TN_j$$

$\Rightarrow$  vrijedi  $FP = FN \Rightarrow$  vrijedi  $P^\mu = R^\mu = F_1^\mu$

Vrijedi  $Acc^M = Acc^\mu$

Primjer ( $N = 13$ ,  $K = 3$ ):

$$\begin{array}{c} \begin{array}{ccc} y = 1 & y = 2 & y = 3 \end{array} \\ \begin{array}{c} y = 1 \\ y = 2 \\ y = 3 \end{array} \left( \begin{array}{ccc} 1 & 1 & 0 \\ 2 & 2 & 3 \\ 0 & 0 & 4 \end{array} \right) \Rightarrow \underbrace{\begin{array}{ccc} y = 1 & y = 2 & y = 3 \end{array}}_{\text{Makro}} \Rightarrow \underbrace{\begin{array}{c} \text{zbroj} \\ \begin{array}{c} 7 & 6 \\ 6 & 20 \end{array} \end{array}}_{\text{Mikro}} \end{array}$$

$$Acc^M = \frac{1}{3} \left( \frac{10}{13} + \frac{7}{13} + \frac{10}{13} \right) = 0.69 \quad Acc^\mu = \frac{27}{39} = 0.69$$

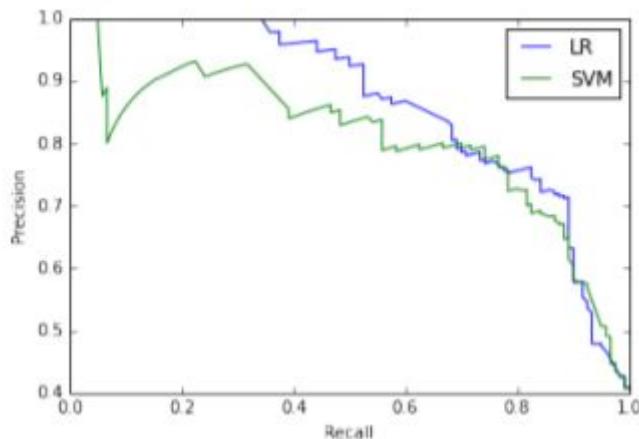
$$P^M = \frac{1}{3} \left( \frac{1}{2} + \frac{2}{7} + \frac{4}{4} \right) = 0.60 \quad P^\mu = \frac{7}{13} = 0.54$$

$$R^M = \frac{1}{3} \left( \frac{1}{3} + \frac{2}{3} + \frac{4}{7} \right) = 0.52 \quad R^\mu = \frac{7}{13} = 0.54$$

$$F_1^M = \frac{1}{3} (0.40 + 0.40 + 0.73) = 0.51 \quad F_1^\mu = \frac{2P^M R^M}{P^M + R^M} = 0.54$$

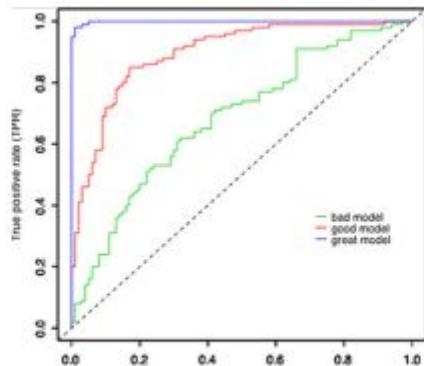
#### 5. Skicirati krivulju preciznost-odziv i definirati prosječnu preciznost.

Krivulja preciznost-odziv (P-R) – preciznost kao funkcija odziva (monotonu opada)



## 6\*. Definirati i skicirati ROC-krivulju te definirati mjeru AUC.

- **Krivulja ROC** – odziv kao funkcija od FPR (fall-out)

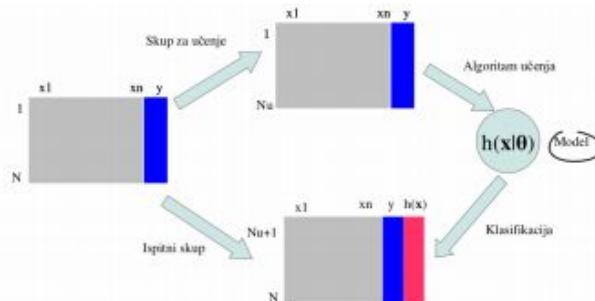


- Nasumična predikcija  $\Rightarrow TPR = FPR$ , neovisno o udjelu pozitivnih primjera
- Agregatna mjera: **površina ispod ROC krivulje (AUC)** (*area under curve*)

## 7. Definirati metodu (ponovljenog) izdvajanja i navesti njene nedostatke.

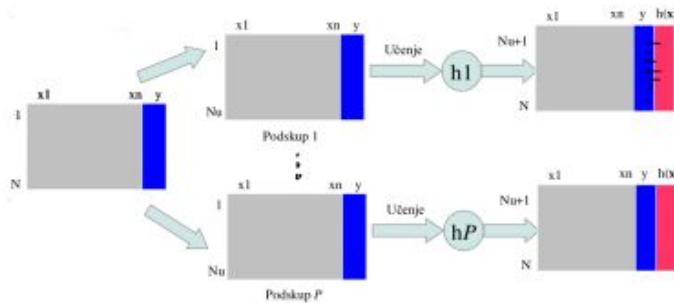
- **Metoda izdvajanja (holdout method)**

- Podjela na skup za učenje i skup za ispitivanje (npr., 70%–30%)
- Prednost: mjerimo pogrešku generalizacije
- Nedostaci: gubitak primjera za učenje, procjena na samo jednom uzorku



- **Ponovljeno izdvajanje (repeated holdout)**

- Višestruko uzorkovanje skupova za učenje/ispitivanje pa izračun prosjeka mjere
- Prednost: procjena pogreške generalizacije na više uzorka
- Nedostatak: ne kontroliramo koji su primjeri i koliko puta upotrijebljeni



## 8. Definirati metodu unakrsne provjere “izdvoji jednog” i navesti njene nedostatke.

- **Metoda izdvoji jednoga (LOOCV)** (*leave-one-out cross-validation*)

- $k$ -struka unakrsna provjera uz  $k = N$
- Prednost: gotovo svi primjeri se koriste za učenje u svakoj iteraciji
- Nedostatci: računalno skupo, visoka varijanca procjene pogreške

## 9. Napisati pseudokod ugniježđene k-struke unakrsne provjere i objasniti prednosti.

### Ugniježđena unakrsna provjera $k \times l$

```
1: podijeli  $\mathcal{D}$  na vanjske preklope  $\mathcal{D}_i$ ,  $i = 1, \dots, k$ 
2: za  $i = 1, \dots, k$  radi: vanjska petlja
3:    $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D} \setminus \mathcal{D}_i$ ,  $\mathcal{D}_{\text{test}} \leftarrow \mathcal{D}_i$ 
4:   za svaku odabrano vrijednost hiperparametra  $\alpha$  radi:
5:     podijeli  $\mathcal{D}_{\text{train}}$  na unutarnje preklope  $\mathcal{D}_j$ ,  $j = 1, \dots, l$ 
6:     za  $j = 1, \dots, l$  radi: unutarnja petlja
7:        $\mathcal{D}_{\text{train}'} \leftarrow \mathcal{D}_{\text{train}} \setminus \mathcal{D}_j$ ,  $\mathcal{D}_{\text{validate}} \leftarrow \mathcal{D}_j$ 
8:       nauči model na  $\mathcal{D}_{\text{train}'}$  i ispita ga na  $\mathcal{D}_{\text{validate}}$ 
9:       izračunaj prosjek mjere na  $l$  unutarnjih preklopa
10:      odaberite hiperparametar  $\alpha$  koji maksimizira prosjek mjere
11:      nauči odabrani model na  $\mathcal{D}_{\text{train}}$  i ispita ga na  $\mathcal{D}_{\text{test}}$ 
12:      izračunaj prosjek mjere na  $k$  vanjskih preklopa
```

PREDNOSTI:

Dobivamo točniju procjenu pogreške nego s metodom izdvajanja (engl. holdout) jer:

- Odabir modela radimo na temelju prosjeka pogreške
- Konačna pogreška modela računa se na temelju prosjeka pogreške

## 10. Objasniti postupak stratifikacije kod unakrsne provjere.

ovo je možda krivo

Iz svake klase se određeni broj primjera na kojima se trenira i iz svake klase se uzme određeni broj primjera na kojima se testira.

Sa stare prezentacije(slajd 23/33):

[https://www.fer.unizg.hr/\\_download/repository/SU-12-VrednovanjeKlasifikatora.pdf](https://www.fer.unizg.hr/_download/repository/SU-12-VrednovanjeKlasifikatora.pdf)

Podjela na skup za učenje i skup za ispitivanje može biti takva da ne zrcali pravu razdiobu primjera u skupu za učenje

- Može rezultirati s pretjerano pesimističnom procjenom

Rješenje je da se skupovi **stratificiraju**, odnosno da razdioba klasa bude sačuvana u oba skupa:

- skup primjera podijeliti u  $K$  podskupova, po jedan za svaku klasu
- svaki takav podskup podijeliti u  $k$  preklop
- združiti  $K$  preklopa (po jedan od svake klase) u jedan preklop

# 22 Vrednovanje modela II

## 1. Objasniti distribuciju uzorkovanja i njenu ulogu u statističkom zaključivanju.

Distribucija neke statistike nad uzorkom, pomoću te distribucije procjenjujemo parametre.

## 2. Definirati distribuciju uzorkovanja procjenitelja srednje vrijednosti.

Takozvani središnji granični teorem (CLT)

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{N}\right)$$

Dakle procjenitelj srednje vrijednosti se ponaša (uvijek! dost bitno) po Gaussovoj distribuciji.  
ko}

## 3. \*Definirati z-statistiku i t-statistiku i navesti pripadne distribucije uzorkovanja.

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{N}} \sim \mathcal{N}(0, 1)$$

Z-statistika se koristi kad treba računati odudara li dobivena srednja vrijednost značajno od stvarne. Varijanca populacije poznata. Ravna se po Gaussovoj distribuciji sa srednjom vrijednošću 0 i varijancom 1.

$$t = \frac{\bar{x} - \mu}{\hat{\sigma}/\sqrt{N}} \sim t(N-1)$$

T-statistika se koristi za računanje odudara li dobivena srednja vrijednost značajno od stvarne, ali sa malo primjera i nepoznata je varijanca populacije (pa ju isto moramo procijeniti). Ravna se po T distribuciji sa N-1 stupnjeva slobode.

## 4. Objasniti potrebu za t-statistikom.

Ako nam je varijanca populacije nepoznata, onda ju moramo procijeniti iz uzorka. Ta greška u procjeni uzrokuje to da nam distribucija statistike vise nije Gaussova razdioba nego studentova T razdioba.

## 5. Objasniti što su populacija, uzorak i statistika kod vrednovanja modela.

Populacija - svi mogući primjeri (moguće beskonačno)

Uzorak - vrijednosti mjere na K preklopa višestruke unakrsne provjere

Statistika - srednja vrijednost mjere kroz K preklopa

## 6. \*Izračunati interval pouzdanosti točnosti modela za zadani primjer.

Stvarni život. Paste-am neke linkove kad ih nađem.

## 7. Objasniti kritičnu vrijednost i p-vrijednost te njihovu ulogu u statističkom zaključivanju.

p-value je vjerojatnost da ćemo, u ponovljenim eksperimentima, dobiti isti ili ekstremniji uzorak, pod pretpostavkom da je H0 hipoteza točna. Dakle to je kinda vjerojatnost da su rezultati koje ste dobili čistom srećom odgovaraju onome sto ste i trebali dobit.

kritična vrijednost je maksimalna vrijednost za koju ne možemo odbaciti H0 u korist H1. Npr za test srednje vrijednosti sa  $N(0, 1)$ , kritična vrijednost za značajnost od 5% je 1.96.

## 8. Napisati korake t-testa za srednju vrijednost i navesti pretpostavke testa.

- Uvjeti: 1) uzorak je došao od populacije s gaussovom distribucijom  
2) varijanca populacije nam nije poznata
1. uzeti nivo značajnosti (najčešće 0.1, 0.05, 0.01)
  2. odabrati željenu srednju vrijednost s kojom želimo testirati
  3. procijeniti varijancu populacije pomoću uzorka.
  4. izracunati t-statistiku (formula iznad!)
  5. usporediti s kritičnom vrijednošću za studentovu T distribuciju sa (N-1) stupnjem slobode.

## 9. Objasniti postupak usporedbe točnosti modela pomoću t-testa.

Nije jasno na što se misli usporedba točnosti modela. Ako se misli na usporedbu dva modela, pogledati stavku ispod. Daljnji tekst ove stavke opisuje izracun intervala pouzdanosti za točnost jednog modela.

Raspolažemo vrijednostima za accuracy nad K preklopa i sad treba izracunati interval pouzdanosti. Prvo moramo definirati nivo znacajnosti (najčešće 0.1, 0.05 i 0.01). Tada u tablici Studentove T distribucije trazimo (dvostranu) kriticnu vrijednost za nivo znacajnosti i N=K-1 stupnjeva slobode. Tu vrijednost označit su sa  $t_{critical}$ . Nadalje, racunamo srednju vrijednost  $m_{sample}$  uzorka od K točnosti i (biased! u nazivniku je K-1) procjenu varijance  $s_{sample}$ . Sirina intervala D je definirana kao  $D = t_{critical} s_{sample} / \sqrt{K}$ . Interval pouzdanosti je zatim jednostavno  $u_{sample} +/- D$ , odnosno  $[u - D, u + D]$

## 10. Jedno/dvostranim t-testom ispitati značajnost razlike točnosti dvaju modela.

- **Upareni t-test** (*matched-pair t-test*): testiranje razlike u točnosti kroz K preklopa
- Uzorak je  $\{d_k\}_{i=1}^K$ , gdje je  $d_i = m_i^A - m_i^B$  razlika u mjeri  $m$  na preklopu  $i$
- Izračunavamo srednju vrijednost razlika,  $\bar{d} = \bar{m}^A - \bar{m}^B = \frac{1}{K} \sum_{i=1}^K d_i$
- Hipoteze:

$$H_0 : \bar{m}^A - \bar{m}^B = \bar{d} = 0 \quad \text{točnosti su iste}$$

$$H_1 : \bar{m}^A - \bar{m}^B \neq 0 \quad \text{točnosti su različite (dvostrani test)}$$

$$\text{ili } H_1 : \bar{m}^A - \bar{m}^B \leq 0 \quad \text{točnost od A je manja/veća od B (jednostrani test)}$$

- t-statistika:

$$t = \frac{\bar{d} - 0}{\hat{\sigma} / \sqrt{K}}, \quad \text{gdje } \hat{\sigma} = \sqrt{\frac{\sum_{i=1}^K (d_i - \bar{d})^2}{K - 1}}$$

- Ako je  $K < 30$ , treba provjeriti vrijedi li normalnost razlika  $d_i$

# 23 Odabir značajki

## 1. Objasniti motivaciju za odabir značajki.

Odabir značajki - odabir podskupa izvornih značajki

- Uklanjanje irrelevantnih i redundantnih značajki povećava točnost modela
- Lakše razumijevanje i objašnjavanje modela
- Pomoć u vizualizaciji podataka

## 2. Razlikovati odabir i transformaciju značajki te njihove prednosti i nedostatke.

**Odabir značajki** (*feature selection*) - odabir podskupa izvornih značajki

**Transformacija značajki** - izvođenje novih značajki iz izvornih značajki

## 3. Objasniti metodu univarijatnog filtra te njezine prednosti i nedostatke.

### Univarijatni filter

Procjena intrinsične vrijednosti (*merit*) svake značajke pa odabir po pragu ili rangu

Prednosti: dobro skalira, računalno jednostavno, nezavisno od modela

Nedostatci: nezavisno od modela, ne uzima u obzir interakciju između značajki

Ideja: značajka  $x_k$  je **relevantna**  $\Leftrightarrow$  postoji **zavisnost** između varijabli  $x_k$  i  $y$

**Uzajamna informacija** – zavisnost varijabli  $x$  i  $y$  kao odstupanje  $P(x, y)$  od  $P(x)P(y)$ :

$$I(x, y) = D_{\text{KL}}(P(x, y) \parallel P(x)P(y)) = \sum_{x,y} P(x, y) \ln \frac{P(x, y)}{P(x)P(y)}$$

$\Rightarrow$  relevantnost značajke  $x_k$  za klasu  $y$  proporcionalna je sa  $I(x_k, y)$

## 4. Objasniti odabir t-testom i izračunati relevantnost značajke za dani primjer.

**t-test** (primjenjivo za  $K = 2$ )

- Test značajnosti razlike srednje vrijednosti od  $x_k$  za klase  $y = 0$  i  $y = 1$
- Hipoteza  $H_0$ : srednje vrijednosti su jednake
- t-statistika (pod  $H_0$  distribuirana po t-distribuciji):

$$t = \frac{\bar{x}_k^0 - \bar{x}_k^1}{\hat{\sigma}_i \sqrt{\frac{1}{N_0} + \frac{1}{N_1}}} \sim t(N_0 + N_1 - 2)$$

gdje  $N_y = \sum_{i=1}^N \mathbf{1}\{y^{(i)} = y\}$  i  $\bar{x}_k^y = \frac{1}{N_y} \sum_{i=1}^N x_k^{(i)} \mathbf{1}\{y^{(i)} = y\}$

- relevantnost značajke  $x_k$  obrnuto je proporcionalna **p-vrijednosti**

## 5. Objasniti odabir $\chi^2$ -testom i izračunati relevantnost značajke za dani primjer.

**$\chi^2$ -test** (primjenjivo za kategoričke značajke)

- Hipoteza  $H_0$ : varijable  $x_k$  i  $y$  su nezavisne ( $x_k \perp y$ )
- $N$  – broj primjera,  $K$  – broj klase,  $K_k$  – broj vrijednosti varijable  $x_k$
- **Tablica kontingencije** dimenzije  $K_k \times K$  sadrži opažene frekvencije  $O_{i,j}$
- Izračun očekivanih frekvencija ( $E_{i,j}$ ) uz pretpostavku  $H_0$ :

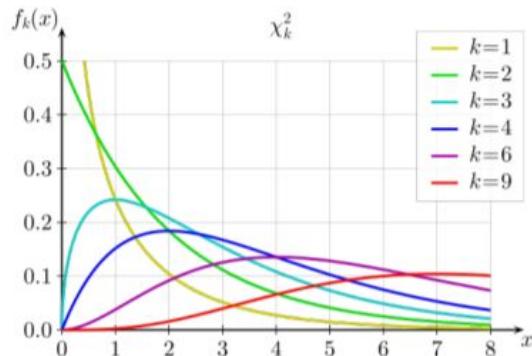
$$P(x_k = i) = \sum_j P(x_k = i, y = j)$$

$$P(y = j) = \sum_i P(x_k = i, y = j)$$

$$E_{i,j} = NP(x_k = i)P(y = j)$$

- $\chi^2$ -statistika (pod  $H_0$  distribuirana po  $\chi^2$ -distribuciji):

$$\chi^2 = \sum_{i=1}^{K_k} \sum_{j=1}^K \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \sim \chi^2((K_k - 1)(K - 1))$$



$\Rightarrow$  relevantnost značajke  $x_k$  obrnuto je proporcionalna **p-vrijednosti**

## Računanje $\chi^2$ -test na primjeru

4)  $\chi^2$ -test  $\rightarrow x_k$  binarna varijacija (multiplijeva),  $k=2$  ili  $k>2$  klase  
 Bernoullijeva

| matrica zabune |    | $y=1$ | $y=0$ |                                       |
|----------------|----|-------|-------|---------------------------------------|
| $x_k=1$        | 30 | 5     |       | $\rightarrow$ želimo „jake“ diagonale |
|                | 10 | 20    |       | (ili / ili \ )                        |
|                |    |       | N=65  |                                       |

ovo su: observed  
 $[O_{ij}]$

$$\left. \begin{array}{l} P(x_k=1) = \frac{30+5}{65} = 0,54 \\ P(x_k=0) = \frac{30}{65} = 0,46 \\ P(y=1) = \frac{40}{65} = 0,62 \\ P(y=0) = \frac{25}{65} = 0,38 \end{array} \right\}$$

Očekivane frekvencije uz pretp.  $x_k \perp y$

$$E[x_k \wedge y] = N \cdot P(x_k) \cdot P(y)$$

$$\left. \begin{array}{c} k \\ k_k \end{array} \right\} \begin{array}{c} y=1 \\ x_k=1 \\ x_k=0 \end{array} \begin{array}{c} 21,76 \\ 13,34 \\ 18,53 \\ 11,36 \end{array} \rightarrow 65 \cdot 0,54 \cdot 0,62$$

$\rightarrow$  pretp. nezavisnosti

$\rightarrow$  expected  $[E_{ij}]$

\* Ono što smo stvarno izmjenili mora biti daleko od ovog  $\rightarrow$  znači da su nekako zanimljive

$$\chi^2 = \sum_{i=1}^{K_k} \sum_{j=1}^k \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \sim \chi^2_{(K-1)(K_k-1)}$$

stupanj slobode za  $\chi^2$

$$\chi^2 = \frac{(30-21,76)^2}{21,76} + \frac{(5-13,34)^2}{13,34} + \dots = 18,83 \rightarrow$$

gleđaj u statističku tablicu za

ovu vrijednost  $\chi^2$  statistike

$\downarrow$

p-vrijednost =  $1,4 \cdot 10^{-5}$  (poprično malo!)

## 6. \*Objasniti postupak RELIEF i napisati pseudokod algoritma.

- **RELIEF** (Kira i Rendell, 1992) – neparametarska iterativna metoda (za  $K = 2$ )

- Iterativno ugadanje vektora relevantnosti svih  $n$  značajki (vektor  $w$ )
- Slučajan odabir pivotnog primjera i primjera iste (*hit*) i različite klase (*miss*)
- Relevantnost  $x_k$  pada ako primjeri istih klasa imaju različite vrijednosti
- Relevantnost  $x_k$  raste ako primjeri različitih klasa imaju različite vrijednosti

### Algoritam RELIEF

- 1: postavi  $w_k \leftarrow 0$  za svaku značajku  $k = 1, \dots, n$
- 2: **za**  $i = 1, \dots, m$  radi:
- 3:     nasumično odaber primjer  $\mathbf{x} \in \mathcal{D}$
- 4:     pronađi najblizi pogodak  $\mathbf{x}^h \in \mathcal{D}$  i promašaj  $\mathbf{x}^m \in \mathcal{D}$  (po L2-normi)
- 5:     **za**  $k = 1, \dots, n$  radi:
- 6:          $w_k = w_k - \frac{1}{N}(x_k - x_k^h)^2 + \frac{1}{N}(x_k - x_k^m)^2$

## 7. Objasniti metodu multivariatnog filtra te njezine prednosti i nedostatke.

Univariatne metode ocjenjuju relevantnost, neovisno o redundanciji značajki

Multivariatne metode ocjenjuju relevantnost i redundantnost skupa značajki

## 8. \*Objasniti postupak VIF za uklanjanje redundantnih značajki.

### ~~• Uklanjanje značajki faktorom inflacije varijance (VIF) (*variance inflation factor*)~~

- Ideja:  $x_k$  je redundantna  $\Leftrightarrow$  može ju se dobro predvidjeti iz drugih varijabli
- Model linearne regresije sa  $x_k$  kao zavisnom varijablom:

$$h_k(x_k; \mathbf{w}) = w_1x_1 + \cdots + w_{k-1}x_{k-1} + w_{k+1}x_{k+1} + \cdots + w_nx_n$$

- VIF varijable  $x_k$ :

$$\text{VIF}_k = \frac{1}{1 - R_k^2} \in [1, \infty)$$

gdje je  $R_k^2$  **koeficijent determinacije** za  $h_k$  (v. odjeljak 5.1.3 dodatka skripti)

- U praksi, značajke za koje  $\text{VIF} \geq 10$  smatraju se redundantnima
- Iterativno uklanjanje redundantnih značajki i ažuriranja VIF vrijednosti
- VIF uklanja isključivo redundantne značajke (ne odabire relevantne značajke)

## 9. Definirati relevantnost skupa značajki mjerom CFS.

**Correlation feature selection (CFS)** – nalazi relevantne i neredundantne značajke

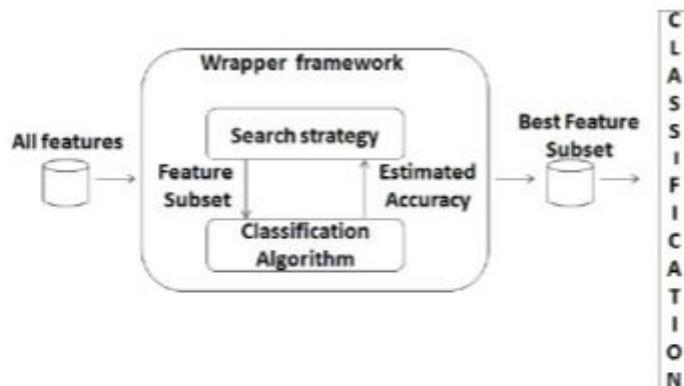
- Ocjena vrijednosti **podskupa značajki**  $S$  koji sadrži  $k$  značajki:

$$\text{Merit}_S = \frac{k\bar{r}_{x,y}}{\sqrt{k + k(k-1)\bar{r}_{x,x}}}$$

- $\bar{r}_{x,y}$  – prosječna korelacija (npr. Pearsonova) između varijabli iz  $S$  i varijable  $y$
- $\bar{r}_{x,x}$  – prosječna korelacija između svih  $k$  varijabli iz  $S$
- **Unaprijedno pretraživanje** prostora od  $2^n$  podskupova metodom **najbolji prvi**

## 10. Objasniti metodu omotača te njezine prednosti i nedostatke.

Pretraživanje prostora od  $2^n$  podskupova značajki i provjera točnosti modela

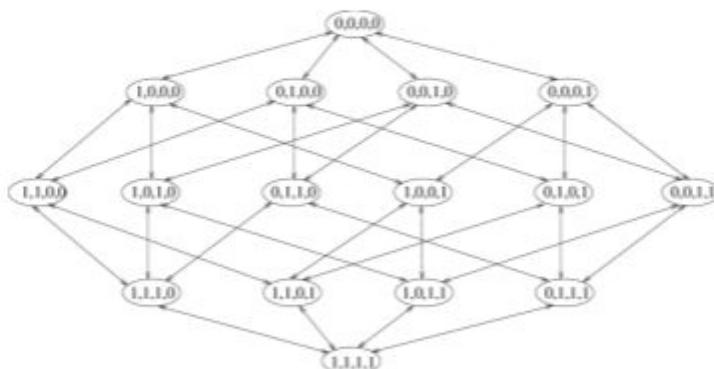


- Kriterijska funkcija:

- **Točnost modela** procijenjena unakrsnom provjerom
  - Mjera **prikladnosti modela** (*goodness of fit*) (npr. F-test)

- Pretraživanje:

- **Unaprijedni odabir** – kreće od praznog skupa i dodaje značajke
  - **Unatražni odabir** – kreće od svih značajki i uklanja značajke
  - **Stepenast odabir** (*stepwise*) – unaprijedan odabir s unatražnim uklanjanjem



- Prednost: prilagođenost konkretnom modelu; nedostatak: računalna složenost

JEL BI MOGAO NETKO TEKST ŠTO SE  
UMETNUO KAO FOTO IZOŠTRITI I MALO  
LJEPŠE STAVITI, KAO ŠTO JE U PRVOM  
DIJELU GRADIVA NAPRAVLJENO, (DA ME  
IDE U STARTU BI TAKO, A NE OVAKO  
LOŠE) HVALA!

→Dobra ideja

gradimo24 Extra...

Kako je kolega predložio :

“Na kraju dokumenta staviti objašnjenja onih zadataka za koje je Šnajder rekao da ih očekujemo u ZI, npr.

DZ10 4 (zadatak s matricom), DZ10 5c (izračun Randovog indeksa)... “

Na to bih nadodao i zadatke iz ZI

TODO...DZ\_10\_4.

4. [Svrha: Isprobati rad algoritma hijerarhijskog aglomerativnog grupiranja (HAC) na konkretnom primjeru, za slučaj kada primjeri nisu vektori. Uočiti razliku između udaljenosti i sličnosti te razliku između jednostrukih i potpunih poveza-nosti.] Jednako kao i algoritam k-medoida, algoritam hijerarhijskog algomerativnog grupiranja može se primijeniti u slučajevima kada primjeri nisu prikazani kao vektori značajki te kada umjesto mjere udaljenosti između vektora raspolažemo općenitjom mjerom sličnosti (ili različitosti). Neka je sličnost primjera iz  $\mathcal{D}$  definirana sljedećom matricom sličnosti:

$$S = \begin{pmatrix} & a & b & c & d & e \\ a & 1.00 & 0.26 & 0.15 & 0.20 & 0.17 \\ b & 0.26 & 1.00 & 0.24 & 0.31 & 0.31 \\ c & 0.15 & 0.24 & 1.00 & 0.20 & 0.50 \\ d & 0.20 & 0.31 & 0.20 & 1.00 & 0.24 \\ e & 0.17 & 0.31 & 0.50 & 0.24 & 1.00 \end{pmatrix}$$

- (a) Izgradite dendrogram uporabom jednostrukog povezivanja. Kada bi bilo potrebno napraviti particiju grupa, na kojoj biste razini presjekli taj dendrogram?
- (b) Izgradite dendrogram uporabom potpunog povezivanja. Kada bi bilo potrebno napraviti particiju grupa, na kojoj biste razini presljekli taj dendrogram?

## TODO...DZ\_10\_5.

5. [Svrha: Razumjeti kako se unutarnji kriterij algoritma grupiranja može (pokušati) upotrijebiti za provjeru grupiranja (odabir optimalnog broja grupa). Razumjeti da Akaikeov kriterij u stvari oponaša regulariziranu funkciju pogreške, koja pak aproksimira pogrešku generalizacije. Isprobati izračun Randovog indeksa na konkretnom primjeru.] Nedostatak svih algoritama grupiranja koje smo razmotrili jest što se broj grupa  $K$  mora zadati unaprijed. Osim u rijetkim slučajevima kada nam je taj broj unaprijed poznat, to predstavlja problem.

- (a) Skicirajte krivulju log-izglednosti kod EM-algoritma kao funkciju broja grupa  $K$ . Obrazložite izgled krivulje. Možete li temeljem ove krivulje odrediti optimalan broj grupa? Kako?
- (b) Optimizacija broja grupa  $K$  može se provesti nekim kriterijem koji kombinira funkciju pogreške (odnosno log-izglednost) i složenost modela. Takav kriterij odgovara strukturnome riziku modela, koji je minimalan za optimalan broj grupa. Jedan takav kriterij jest Akaikeov informacijski kriterij (AIC):

$$K^* = \operatorname{argmin}_K (-2 \ln \mathcal{L}(K) + 2q(K)) \quad (1)$$

gdje je  $-\ln \mathcal{L}(K)$  negativna log-izglednost podataka za  $K$  grupa, a  $q(K)$  je broj parametara modela s  $K$  grupa.

Prepostavite da podatci  $\mathcal{D}$  u stvarnosti dolaze iz  $K = 5$  grupa. Podatke grupiramo dvjema varijantama EM-algoritma: standardni algoritam i preinačeni algoritam s dijeljenom kovarijacijskom matricom (zajednička kovarijacijska matrica procijenjena nad čitavim skupom primjera  $\mathcal{D}$  na početku izvođenja algoritma). Skicirajte za ta dva algoritma funkciju koju minimizira Akaikeov minimizacijski kriterij.

- (c) Kada su primjeri ili podskup primjera označeni, kvaliteta grupiranja (uključivo i broj grupa  $K$ ) može se procijeniti Randovim indeksom. Randov indeks zapravo izračunava točnost s kojom ćemo par jednako označenih primjera smjestiti u istu grupu, odnosno par različito označenih primjera u različitu grupu. Izračunajte Randov indeks za sljedeću pariticiju označenih primjera (podskupovi su grupe dobivene grupiranje, a brojke su oznake klase primjera):

$$\{\{0, 0, 1, 2\}, \{1, 1\}, \{2, 2, 2, 1, 0\}\}.$$

- (d) Skicirajte vrijednost Randovog indeksa kao funkcije broja grupa  $K$ .
- (e) Randov indeks možemo koristiti samo ako su podatci označeni ili je podskup podataka označen. Međutim, čini se da to onda ujedno podrazumijeva da je unaprijed poznat broj grupa  $K$ . Imamo li koristi od Randovog indeksa čak i onda kada unaprijed znamo broj grupa? Možemo li ikako upotrijebiti Randov indeks, a da nam broj grupa nije unaprijed poznat?

## PROCJENITELJI 16/17

1. (8 bodova) Procjenitelji.

- Definirajte funkciju log-izglednosti  $\ln \mathcal{L}(\theta|\mathcal{D})$  i objasnite na kojoj se pretpostavci ona temelji. Zašto radimo s logaritmom izglednosti i zašto je to opravdano?
- Skicirajte  $\ln \mathcal{L}(\mu, \sigma^2|\mathcal{D})$  kao funkciju od  $\mu$  za skup primjera  $\mathcal{D} = \{0, 2, 4\}$  uz pretpostavku da se primjeri ravnaju po Gaussovoj razdiobi,  $p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ .
- Definirajte ML-procenitelj te izvedite  $\hat{\mu}_{ML}$ , korak po korak, za parametar  $\mu$  univariatne Gaussove razdiobe. Je li ta procjena nepristrana i što to znači?
- Definirajte MAP-procenitelj. Kada MAP-procenitelj ima rješenje u zatvorenoj formi?
- Krenuvši od MAP-procenitelja, izvedite Laplaceov procjenitelj za parametar  $\mu$  Bernoullijeve varijable. Gustoća vjerojatnosti beta-distribucije jest  $p(\mu|\alpha, \beta) = \mu^{\alpha-1}(1-\mu)^{\beta-1}/B(\alpha, \beta)$ , a mod je  $\frac{\alpha-1}{\alpha+\beta-2}$ .
- Objasnite koja je veza između MLE-procjena parametara  $w$  kod linearne i logističke regresije i minimizacije pogrešaka tih modela.

a) temelji se na iid pretpostavci koja govori da je skup podataka nezavisno i identično distribuiran (ista distribucija za sve primjere)

zbog matematičke jednostavnosti (prodot sa logaritmom postaje suma pa je lakše i "jeftinije" za računalo), max. logaritamske funkcije je jednak maximumu obične funkcije, opravdano je zato jer logaritamska funkcija monotono rastuća

b)

$$D = 0, 2, 4$$

$$L = \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x_i - \mu}{2\sigma^2}\right) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^3 \exp\left(-\frac{(0 - \mu)^2 + (2 - \mu)^2 + (4 - \mu)^2}{2\sigma^2}\right)$$

$$\ln(L) = -C(3\mu^2 - 12\mu + 20)$$

gdje je C neka konstantica koja nam nije previse bitna, a ovisi o sigma<sup>2</sup>  
ocekivano, parabolica koju smo dobili ima maksimum u u=2.

b)

c) ?!

d) Ima rješenje u zatvorenoj formi kad su apriorna i aposterorina distribucija konjugantne, tj ako koristimo neku od apirono konjugantnih parova distribucija. Tipa bernoulijeva i beta, dirchletova i multinulijeva i normalna i normalna.

e) ?!

f) ?!

2. (7 bodova) Bayesov klasifikator.

- (a) Napišite model naivnog Bayesovog klasifikatora. Napišite sve pretpostavke i opišite sve induktivne pristranosti ovog modela.
- (b) Definirajte bilo kakav polunaivan diskretan Bayesov klasifikator i napišite njegovu "generativnu priču".
- (c) Izgrađujemo Bayesov model za klasifikaciju primjera iz  $\mathcal{X} = \mathbb{R}$  u tri klase. Učenjem na skupu primjera dobili smo sljedeće parametre modela:  $P(\mathcal{C}_1) = 0.7$ ,  $P(\mathcal{C}_2) = 0.2$ ,  $\mu_1 = -2$ ,  $\mu_2 = 0$ ,  $\mu_3 = 1$ ,  $\sigma_1^2 = 2$ ,  $\sigma_2^2 = 3$ ,  $\sigma_3^2 = 1$ . Skicirajte funkcije gustoće vjerojatnosti  $p(x|\mathcal{C}_j)$ ,  $p(x)$  i  $p(\mathcal{C}_j|x)$ .
- (d) Kod multivarijatnog Bayesovog klasifikatora, izglednosti klasa definirane su gustoćom:

$$p(\mathbf{x}|y=j) = \frac{1}{(2\pi)^{n/2} |\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j)\right).$$

Izvedite općeniti izraz za model  $h(\mathbf{x}|\theta)$  s dijeljenom kovarijacijskom matricom. Skicirajte u prostoru  $\mathcal{X} = \mathbb{R}^2$  konture gustoća razdiobe  $p(x_1, x_2, y)$  i vjerojatnosti  $P(y|x_1, x_2)$  za dvije klase.

- (e) Izvedite vezu između modela logističke regresije i modela kontinuiranog Bayesovog klasifikatora.

## -VREDNOVANJE 16/17

3. (6 bodova) Vrednovanje i statističko testiranje klasifikatora.
- Izračunajte makro-točnosti i makro- $F_1$  na temelju sljedeće matrice zabune (retci odgovaraju predviđenoj, a stupci stvarnoj kategoriji):
 
$$\begin{pmatrix} 10 & 4 & 6 \\ 8 & 19 & 8 \\ 5 & 5 & 21 \end{pmatrix}.$$
  - Za vrednovanje SVM-a koristimo (naravno) ugniježđenu unakrsnu provjeru s 5 vanjskih i 5 unutarnja preklopa. Hiperparametre optimiramo pretraživanjem po rešetci. Hiperparametri su jezgra (linearna ili RBF), parametar  $C \in \{2^{-5}, 2^{-4}, \dots, 2^5\}$  i parametar  $\gamma \in \{2^{-15}, 2^{-14}, \dots, 2^5\}$ . Koliko ćemo puta ukupno trenirati model i kako biste odredili ukupno optimalne hiperparametre?
  - Trenirali smo model  $h_2$  i želimo provjeriti je li njegov  $F_1$  statistički značajno različit od  $F_1$  modela  $h_1$ . Oba modela vrednujemo desetorostrukom unakrsnom provjerom na ukupno  $N=1000$  primjera te računamo točnosti oba modela na svakom od deset preklopa (lijeva tablica).

| $i$ | $F_1(h_1)$ | $F_1(h_2)$ | $i$ | $F_1(h_1)$ | $F_1(h_2)$ | $df$ | 0.10  | 0.05  | 0.02  | 0.01  | 0.005 |
|-----|------------|------------|-----|------------|------------|------|-------|-------|-------|-------|-------|
| 1   | 0.627      | 0.595      | 6   | 0.462      | 0.677      | 7    | 1.895 | 2.365 | 2.998 | 3.499 | 4.029 |
| 2   | 0.570      | 0.581      | 7   | 0.541      | 0.613      | 8    | 1.860 | 2.306 | 2.897 | 3.355 | 3.833 |
| 3   | 0.396      | 0.630      | 8   | 0.539      | 0.631      | 9    | 1.833 | 2.262 | 2.821 | 3.250 | 3.690 |
| 4   | 0.529      | 0.691      | 9   | 0.541      | 0.613      | 10   | 1.812 | 2.228 | 2.764 | 3.169 | 3.581 |
| 5   | 0.562      | 0.518      | 10  | 0.539      | 0.631      |      |       |       |       |       |       |

Uparenim t-testom testirajte je li točnost modela  $h_2$  statistički značajno različita od točnosti  $h_1$  na razini značajnosti  $\alpha = 1\%$  te riječima formulirajte zaključak. Kritične vrijednosti za dvostrani t-test dane su u desnoj tablici (retci: stupnjevi slobode; stupci:  $\alpha$  za dvostrani test).

## GRUPIRANJE 16/17

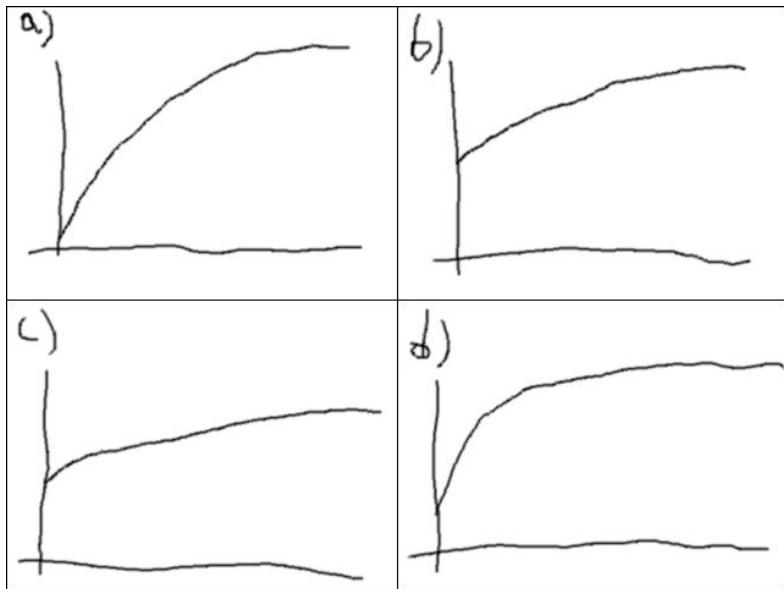
### 4. (7 bodova) Grupiranje.

- (a) Napišite funkciju pogreške algoritma k-srednjih vrijednosti i iz nje izvedite pseudokôd algoritma. Koja je vremenska a koja prostorna složenost ovog algoritma?
- (b) Sličnosti između primjera definirane su sljedećom matricom *sličnosti*:

$$S = \begin{matrix} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{pmatrix} 1.0 & 0.1 & 0.3 & 0.1 \\ 0.1 & 1.0 & 0.1 & 0.2 \\ 0.3 & 0.1 & 1.0 & 0.2 \\ 0.1 & 0.2 & 0.2 & 1.0 \end{pmatrix} \end{matrix}$$

Primijenite hijerarhijsko aglomerativno grupiranje (HAC) s jednostrukim povezivanjem te skicirajte pripadni dendrogram. Na dendrogramu naznačite sličnosti na kojima se grupe spajaju.

- (c) Raspolažemo skupom neoznačenih primjera i manjim podskupom od 8 primjera označenih u tri klase te želimo napraviti provjeru grupiranja. Za  $K = 3$  i  $K = 4$ , algoritam primjere grupira u particiju  $\{\{0, 2, 2\}, \{0, 0\}, \{1, 1, 2\}\}$  odnosno  $\{\{0, 2, 2\}, \{0, 0\}, \{1, 1\}, \{2\}\}$ . Izračunajte Randove indekse. Skicirajte krivulju Randovog indeksa kao funkciju broja grupe  $K$ .
- (d) Napišite izraz za mješavinski model s latentnim varijablama. Koja je značenje latentnih varijabli? Izvedite izraz za (potpunu) log-izglednost i ukratko objasnite na koji način dalje provodimo optimizaciju.
- (e) Raspolažemo neoznačenim podatcima  $\mathcal{D}$  koji potječu iz  $K=12$  klase i čije su značajke međusobno visoko linearne zavisne. Podatke grupiramo modelom Gaussova mješavina, i to modelom s nedijeljenim kov. matricama (GMM-full) i s dijeljenom dijagonalnom kov. matricom (GMM-diag). Skicirajte očekivani izgled log-izglednosti  $\ln \mathcal{L}(\theta | \mathcal{D})$  kao funkcije broja iteracija, i to za: (1) GMM-full s nasumično odabranim  $K=12$  središta i (2) GMM-full inicijaliziran algoritmom k-srednjih vrijednosti sa  $K=12$  središta, (3) GMM-diag inicijaliziran algoritmom k-srednjih vrijednosti sa  $K=12$  središta te (4) GMM-diag s nasumično odabranim  $K=100$  središta (ukupno četiri krivulje).
- e) nisam siguran skroz ali mislim da je ovo blizu točnog odgovora....



tri su bitne odrednice svakog graf-a:

- 1) pocetak na y osi (za nasumicno je pocetak dost los, a za sredista je pocetak kao malo bolji)
  - 2) krajnja vrijednost (za neke overfittane modele je dosta visoka, za modele koji generaliziraju je nesto niza)
  - 3) brzina porasta (za overfittane modele raste jako brzo, i ako ima puno sredista isto jako brzo raste)
- GMM-full se lako overfitta, GMM-diag kao bolje generalizira, algoritam k-sredista je super u inicijalizaciji te K=100 puno brze spusta gresku od K=12.
- po meni svi grafovi bi trebali biti strogo rastuci (dakle ako je negdje pad, to je do moje nesposobnosti crtanja, a ne do strojnog ucenja), ali opet mozda sam promasio citav nogomet

## DZ - skice rješenja s predavanja

- sorry što je neuredno, trebalo je brzo pisati

### DZ 8 - Bayesov klasifikator

1. [Svrha: Razumjeti model Bayesovog klasifikatora i njegove komponente. Razumjeti što su to generativni modeli, kako se razlikuju od diskriminativnih te koje su njihove prednosti i njihovi nedostatci.]

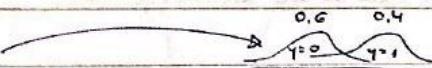
- Definirajte model Bayesovog klasifikatora i navedite sve veličine koje se pojavljuju u definiciji modela. Objasnite zašto faktoriziramo brojnik. Objasnite ulogu nazivnika i objasnite kada ga možemo zanemariti.
- Je li taj model parametarski ili neparametarski? Obrazložite odgovor.
- Objasnite zašto Bayesov model nazivamo generativnim i opišite generativnu priču Bayesovog klasifikatora.
- Objasnite razliku između generativnih i diskriminativnih modela te navedite prednosti jednih i drugih.

1. a)

b) ~~parametarski~~, pretpostavlja distribuciju

c)  $p(y|x) = \frac{p(x|y) \cdot p(y)}{p(x)}$

1. izaberem klasu



2. samplam iz nje po Gauss.distr.

d) diskr. ne modeliraju zaj.y.

↳ manje param. za isti posao

generativni → prednost: interpretabilnost

3. [Svrha: Razumjeti faktorizaciju zajedničke vjerojatnosti uz pretpostavku uvjetne nezavisnosti te povezanost toga s induktivnom pristranošću i, posljedično, brojem značajki modela.]

- Definirajte naivan Bayesov klasifikator i pretpostavku na kojoj se temelji.
- Zašto nam treba pretpostavka o uvjetnoj nezavisnosti značajki te kojoj vrsti induktivne pristranosti ona odgovara?
- Naivan Bayesov klasifikator koristimo za klasifikaciju rukom pisanih znamenki u jednu od deset klasa. Znamenke su prikazane kao vektor binarnih značajki (crno/bijeli slikovni elementi) u matrici s razlučivošću  $32 \times 32$ . Odredite ukupan broj parametara naivnog Bayesovog klasifikatora.

3. b)

$$c) P(y|\bar{x}) = \operatorname{argmax}_j P(y=j) \prod_{i=1}^n P(x_i|y), K=10$$

$n = 32 \times 32 = 2^{10} = 1024$

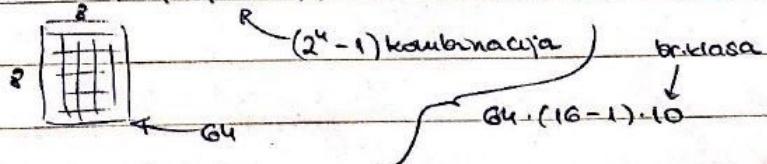
kaubinutljiva distr.  $(10-1)$

CPT, sve kombinacije - 1  
 $\rightarrow [(2^4 - 1) \cdot 10] \cdot 1024$   
 $2 \text{ mogućnosti}$  (crno-bijeli)  
 $K$  svih piksela  
 $= 10249$

\* polunajvni  $\rightarrow$  počinju spajati neke značajke (npr. susjedne piksele)

$\rightarrow$  npr. gledala sam četvorke  $\cdot P(x_1, x_2, x_3, x_4 | y)$

$\rightarrow$  blokovi se ne preklapaju



$\rightarrow$  po 4, ali s kliznim prozorom  $\rightarrow$  model složeniji  $\Rightarrow$  isto 1024 puta više param.

6. [Svrha: Razviti intuiciju za model kontinuiranog Bayesovog klasifikatora.]

Izrađujemo Bayesov model za klasifikaciju primjera iz  $\mathcal{X} = \mathbb{R}$  u tri klase. Učenjem na skupu primjera dobili smo sljedeće parametre modela:  $P(y=1) = 0.3$ ,  $P(y=2) = 0.2$ ,  $\mu_1 = -5$ ,  $\mu_2 = 0$ ,  $\mu_3 = 5$ ,  $\sigma_1^2 = 5$ ,  $\sigma_2^2 = 1$ ,  $\sigma_3^2 = 10$ . Skicirajte funkcije gustoće vjerojatnosti  $p(x|y)$ ,  $p(x,y)$ ,  $p(x)$  i  $p(y|x)$ .

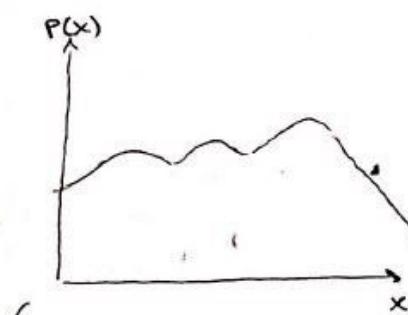
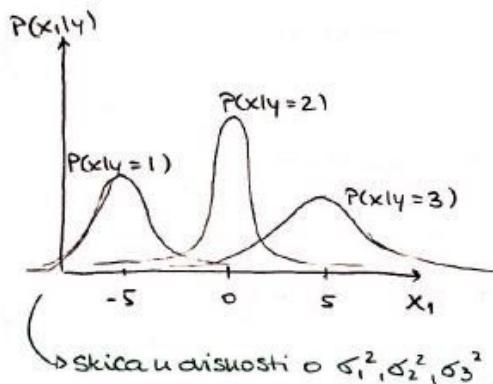
DZ 8 , 6. zadatak

3 klase

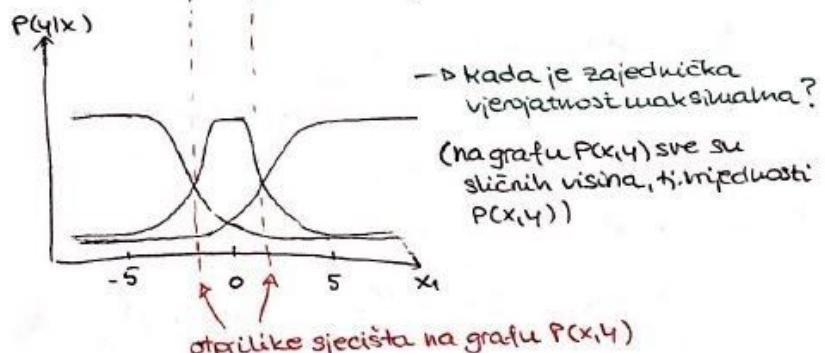
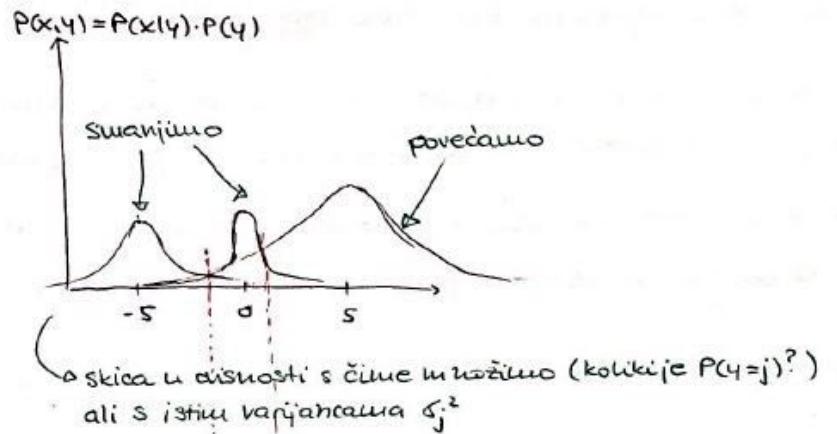
$$P(y=1) = 0.3, P(y=2) = 0.2$$

$$\mu_1 = -5, \mu_2 = 0, \mu_3 = 5$$

$$\sigma_1^2 = 5, \sigma_2^2 = 1, \sigma_3^2 = 10$$

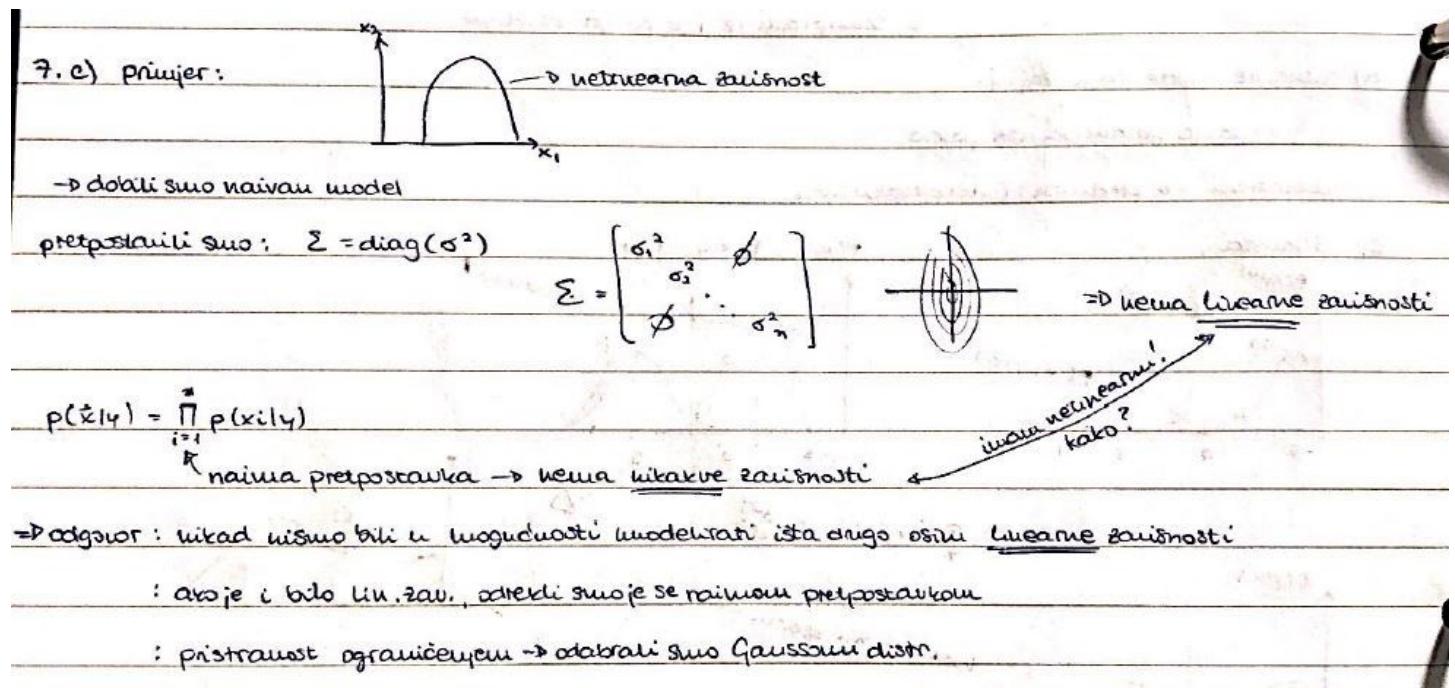


$$\Rightarrow P(x) = \sum_y p(x,y), \text{otprilike zbroj s grafa } P(x,y)$$



7.

- (c) Objasnite zašto je izglednost faktorizirana u produkt univarijatnih razdioba, što odgovara pretpostavci o uvjetnoj nezavisnosti, premda značajke mogu biti nelinearno uvjetno zavisne.



8. [Svrha: Razviti intuiciju za složenost modela kontinuiranog Bayesovog klasifikatora i shvatiti kako se problem u konačnici svodi na odabir optimalnog modela.] Želimo izgraditi klasifikator za klasifikaciju bruča u jednu od dvije klase:  $y = 1 \Rightarrow$  "Završava FER u roku" i  $y = 2 \Rightarrow$  "Produljuje studij". Svaki je primjer opisan sa šest ulaznih varijabli: prosjek ocjena 1.–4. razreda (četiri varijable), bodovi državne mature iz matematike te bodovi državne mature iz fizike. Raspolažemo trima modelima: modelom  $\mathcal{H}_1$  s dijeljenom kovarijacijskom matricom, modelom  $\mathcal{H}_2$  s diagonalnom (i dijeljenom) kovarijacijskom matricom i modelom  $\mathcal{H}_3$  s izotropnom kovarijacijskom matricom.

- Koliko svaki od ova tri modela ima parametara?
- Za koji od ova tri modela očekujete da će najbolje generalizirati u ovom konkretnom slučaju (uzmite u obzir prirodu problema i očekivane odnose između značajki)? Zašto?
- Nacrtajte skicu funkcije empirijske pogreške i pogreške generalizacije i naznačite na njoj točke koje označavaju navedenim trima modelima.
- Kako biste u praksi odredili koji ćete model upotrijebiti?

8.a)  $n=6$   $\Sigma = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}^G$

$H_1$  - puna kovarijacijska matrica (učeščemu lin. zavisnosti, imam sve trikotne unutra)

• br. param. =  $(6 + \frac{6 \cdot 7}{2}) \cdot 2 + 1$   
 $\uparrow$   $\uparrow$   $\uparrow$   $\uparrow$   
 $\mu$   $\Sigma$  2 klase verojatnost klase

$H_1$  - dijagonala

• br. param. =  $2 \cdot 6 + \frac{6 \cdot 7}{2} + 1$   
 $p(4)$

$H_2$  - dijagonala  $\Sigma = \begin{bmatrix} \phi & & \\ & \phi & \\ & & \phi \end{bmatrix}$

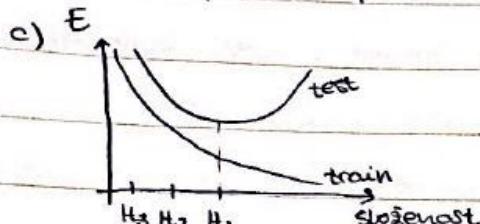
br. param. =  $2 \cdot 6 + 6 + 1$

$H_3$ ,  $\Sigma = \sigma^2 I = \begin{bmatrix} \sigma^2 & & \\ & \sigma^2 & \\ & & \ddots & \\ & & & \sigma^2 \end{bmatrix} \Rightarrow$  br. param. =  $2 \cdot 6 + 1 + 1$

8.b) u praksi: unakrarna projekcija

Ovdje: postoji korelacija!  $\rightarrow$  onda kojiboji smaj s punom  $\Sigma$

$H_1$  kojibije generalizira



$\rightarrow$  točka je čak i pretežirana, može i više učesno

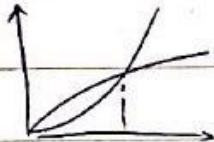
d) unakrarna projekcija

9. [Svrha: Razumjeti vezu između Bayesovog klasifikatora i logističke regresije, odnosno probabiličku interpretaciju logističke regresije. Razumjeti razliku u broju parametara između diskriminativnog i generativnog modela te utjecaj broja klasa i broja primjera na taj odnos.]

- Izvedite model logističke regresije krenuvši od generativne definicije za  $P(y = 1|\mathbf{x})$ . Izvod napravite korak po korak te se uvjerite da možete obrazložiti svaki korak u izvodu. Napišite sve pretpostavke koje ste ugradili u izvod.
- Model logističke regresije koristimo za binarnu klasifikaciju primjera s  $n = 100$  značajki. Odredite broj parametara modela logističke regresije te njemu odgovarajućeg generativnog modela.
- Izračunajte broj parametara za isti slučaj, ali sa  $K = 5$  klasa.
- Prepostavite da klasificiramo u  $K = 10$  klasa. Izračunajte koliko velika mora biti dimenzija prostora značajki  $n$ , a da bi se logistička regresija isplatila jer ima manje parametara od odgovarajućega generativnog modela.

9. → ISPIT!

\* Čim imam djelejnu kov.matriцу ⇒ imam lin.



Bayes:  $\frac{n}{2}(n+1) + 2n + 1$       Log.reg:  $n+1$

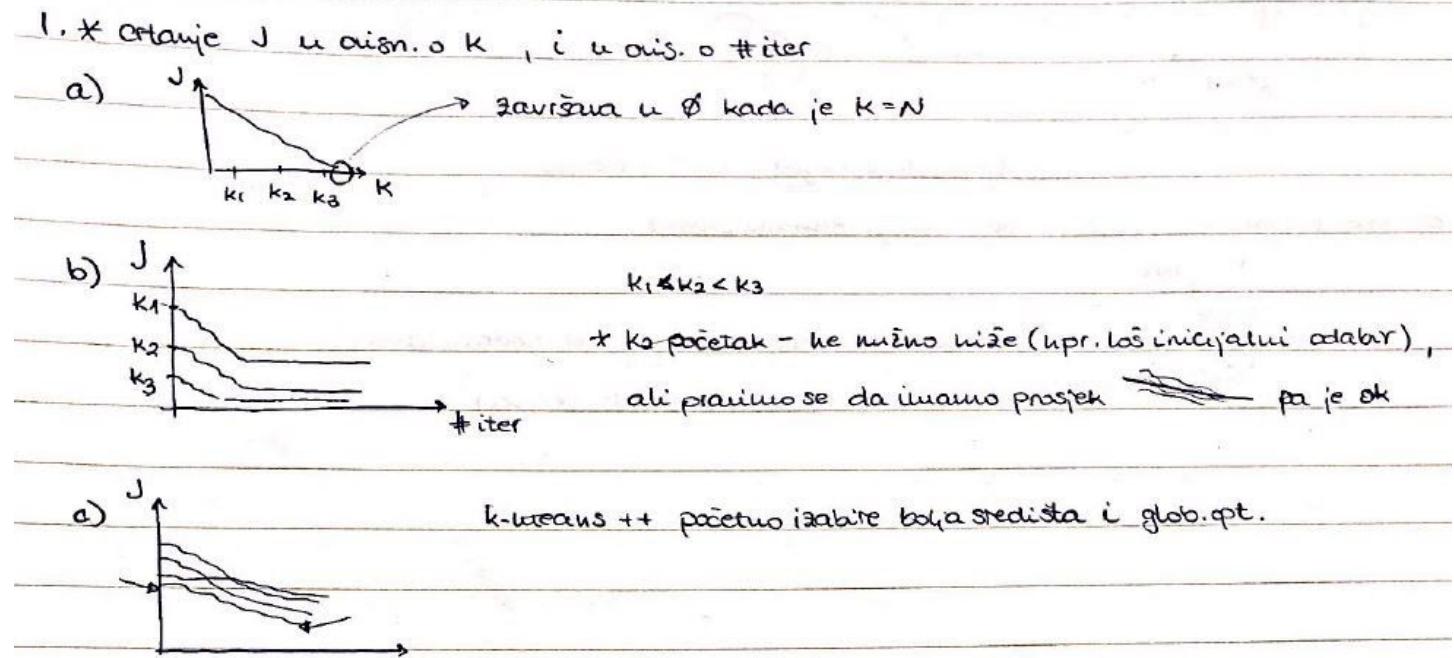
9. Gotovo sigurno dolazi u ispit - netko c i d?

## DZ 10 - Grupiranje

1. [Svrha: Razumjeti rad algoritma  $k$ -sredina u smislu minimizacije kriterija pogreške. Razumjeti kako rad algoritma ovisi o broju grupa  $K$  i odabiru početnih središta.]

Algoritam  $k$ -sredina minimizira kriterij pogreške  $J(\mu_1, \dots, \mu_K | \mathcal{D})$ . Vrijednost tog kriterija ovisi o broju grupa  $K$ , koji je unaprijed postavljen, te o položajima središta, koja se mijenjaju kroz iteracije.

- Nacrtajte skicu vrijednosti kriterija pogreške  $J$  kao funkcije broja grupa  $K$ . Koja je minimalna vrijednost funkcije  $J$  i zašto?
  - Izaberite na skici iz zadatka (a) tri vrijednosti za  $K$  i skicirajte na jednom grafikonu vrijednost kriterija pogreške  $J$  kao funkcije broja iteracija (tri krivulje).
  - Izaberite na skici iz zadatka (a) jednu vrijednost za  $K$ . Skicirajte na jednom grafikonu vrijednosti kriterija pogreške  $J$  kao funkcije broja iteracija, ali ovaj put uvezši u obzir stohastičnost uslijed slučajnog odabira početnih središta (nacrtajte nekoliko mogućih krivulja na istom grafikonu). Koje od tih krivulja su izglednije za algoritam  $k$ -means++?
- c) dakle bezveze krivulje neke ispod neke iznad, nek se sijeku, stohastično je :D, a onda onu koja je NAJNIŽE POČELA (početno bolje izabire) i NAJNIŽE ZAVRŠILA (globalni optimum) uzimamo kao najvjerojatniju za kmeans++



2. [Svrha: Isprobati rad algoritma k-sredina i k-medoida na konkretnom primjeru.  
Shvatiti da je složenost ovog drugog puno nepovoljnija.] Raspolažemo skupom neoznačenih primjera:

$$\mathcal{D} = \{a = (5, 2), b = (7, 1), c = (1, 4), d = (6, 2), e = (2, 8), f = (3, 6), g = (0, 4)\}.$$

- (a) Izvedite jedan korak algoritma k-sredina uz  $K = 3$ . Za početna središta odaberite  $\mu_1 = b$ ,  $\mu_2 = c$  i  $\mu_3 = e$ .
- (b) Izvedite jedan korak algoritma k-medoida uz  $K = 3$ . Za početna središta odaberite primjere  $b$ ,  $c$  i  $e$ .
- (c) Usporedite računalnu složenost algoritma k-sredina i k-medoida.
- (d) Što su prednosti, a što nedostatci algoritma k-medoida?

2.d) pred: općenita mjera sličnosti

ako imam  
euklidsku udalj.

ned: nem. složenost (kvadratna), prostor ne istoga k-means i k-medoida

$$* S = : \begin{pmatrix} & 1 & 2 & \dots \\ 1 & \bullet & \bullet & \dots \\ 2 & \bullet & \bullet & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \text{ matrica sličnosti}$$

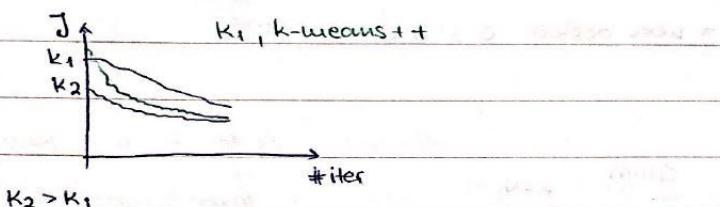
\* ISPIT: PSEUDOKOD!

3. [Svrha: Razumjeti model miješane gustoće i razlog zašto maksimizacija log-izglednosti nije analitički rješiva. Razumjeti kako uvođenje latentnih varijabli rješava taj problem. Razumjeti, na općenitoj razini, E-korak i M-korak. Razumjeti rad algoritma kao maksimizacije log-izglednosti i razumjeti kako ishod ovisi o broju grupa i početnoj inicijalizaciji.] Algoritam maksimizacije očekivanja (EM-algoritam), kada se koristi za grupiranje, zapravo je poopćenje algoritma k-sredina.

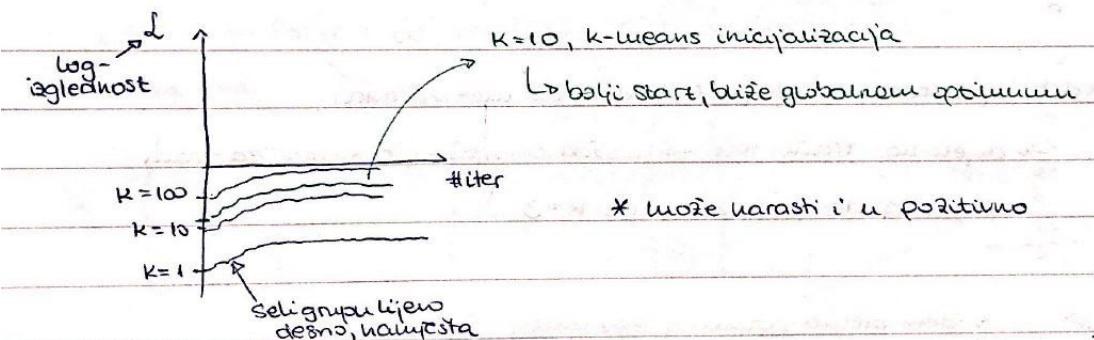
- Što je prednost, a što nedostatak, algoritma maksimizacije očekivanja u odnosu na algoritam k-sredina?
- Napišite izraz za gustoću  $p(\mathbf{x})$  za model miješane gustoće (bez latentnih varijabli) i izraz za pripadnu (nepotpunu) log-izglednost.
- Napišite izraz za mješavinu s latentnim varijablama i izvedite izraz za (potpunu) log-izglednost tog modela. Možemo li dalje raditi izravno s tom log-izglednošću? Zašto?
- Definirajte E-korak i M-korak algoritma maksimizacije očekivanja primjenjene na Gaussovou mješavinu.
- Skicirajte vrijednost log-izglednosti  $\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$  modela Gaussove mješavine kao funkcije broja iteracija, i to za tri različite vrijednosti parametra  $K$  (broj grupa):  $K = 1$ ,  $K = 10$  i  $K = 100$ . Na istom grafikonu skicirajte krivulju za  $K = 10$  kada se za inicijalizaciju središta koristi algoritam k-sredina.

(D2) 3.c) Ne možemo jer je ov sluč.var.  $\rightarrow$  očekivanje

e) \* K-means



GMM+EM



4. [Svrha: Isprobati rad algoritma hijerarhijskog aglomerativnog grupiranja (HAC) na konkretnom primjeru, za slučaj kada primjeri nisu vektori. Uočiti razliku između udaljenosti i sličnosti te razliku između jednostrukih i potpunih povezivanih.] Jednako kao i algoritam k-medoida, algoritam hijerarhijskog algomerativnog grupiranja može se primijeniti u slučajevima kada primjeri nisu prikazani kao vektori značajki te kada umjesto mjere udaljenosti između vektora raspolažemo općenitjom mjerom sličnosti (ili različitosti). Neka je sličnost primjera iz  $\mathcal{D}$  definirana sljedećom matricom sličnosti:

$$S = \begin{matrix} & a & b & c & d & e \\ a & \left( \begin{array}{ccccc} 1.00 & 0.26 & 0.15 & 0.20 & 0.17 \\ 0.26 & 1.00 & 0.24 & 0.31 & 0.31 \\ 0.15 & 0.24 & 1.00 & 0.20 & 0.50 \\ 0.20 & 0.31 & 0.20 & 1.00 & 0.24 \\ 0.17 & 0.31 & 0.50 & 0.24 & 1.00 \end{array} \right) \end{matrix}$$

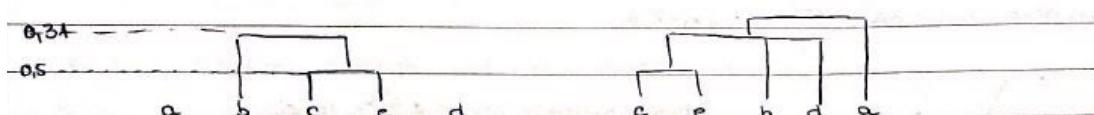
- (a) Izgradite dendrogram uporabom jednostrukog povezivanja. Kada bi bilo potrebno napraviti particiju grupe, na kojoj biste razini presjekli taj dendrogram?  
(b) Izgradite dendrogram uporabom potpunog povezivanja. Kada bi bilo potrebno napraviti particiju grupe, na kojoj biste razini presljekli taj dendrogram?

→ matrica sličnosti! (pozicija je zadana)

$$S = \begin{matrix} & a & b & c & d & e \\ a & \left( \begin{array}{ccccc} 1 & 0.26 & 0.15 & 0.20 & 0.17 \\ 0.26 & 1 & 0.24 & 0.31 & 0.31 \\ 0.15 & 0.24 & 1 & 0.20 & 0.50 \\ 0.20 & 0.31 & 0.20 & 1 & 0.24 \\ 0.17 & 0.31 & 0.50 & 0.24 & 1 \end{array} \right) \end{matrix}$$

1. svaki primjer u svojoj grupi  
2. najveća vr. → najsljednji par  
=⇒ c i e na 0,5

\* dečko



→ complete

→ min., ali max-udaljenosti

nova sličnost → jednostruko povezivanje → učinjam max sličnosti, min udaljenosti

$$\Rightarrow S = \begin{matrix} & a & b & c+e & d \\ a & \left( \begin{array}{ccccc} 1 & 0.26 & 0.17 & 0.20 \\ 0.26 & 1 & 0.31 & 0.31 \\ 0.17 & 0.31 & 1 & 0.24 \\ 0.20 & 0.31 & 0.24 & 1 \end{array} \right) \end{matrix}$$

→ što želim spojiti (sve jedno)  
b i (c+e)  
b i d

$$\hookrightarrow S = \begin{matrix} a \\ bce \\ d \end{matrix}$$

\* neki dečko c e b d a

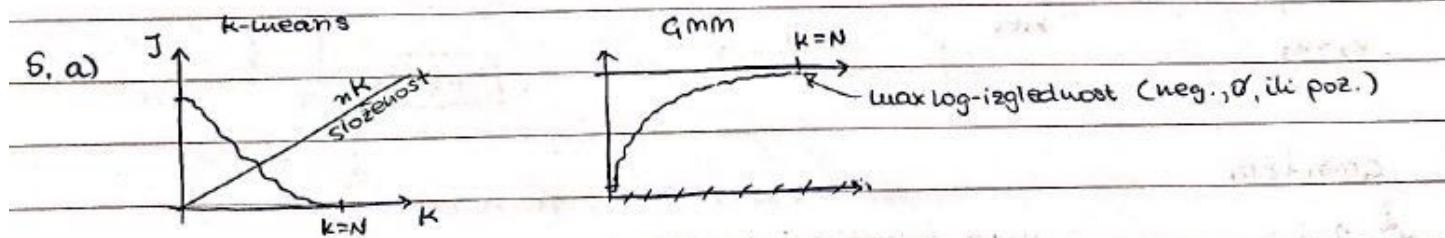
\* dečko i neki dečko su oznaće da je to rješenje napisao neki student a šnajder ga je gledao dok piše na ploču :)- gdje presjeći? :(9

Dakle lijevi dendrogram je onaj kojeg je počeo šnajder, a desni je gotov od tog nekog dečka

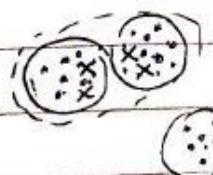
DZ10

5. zad

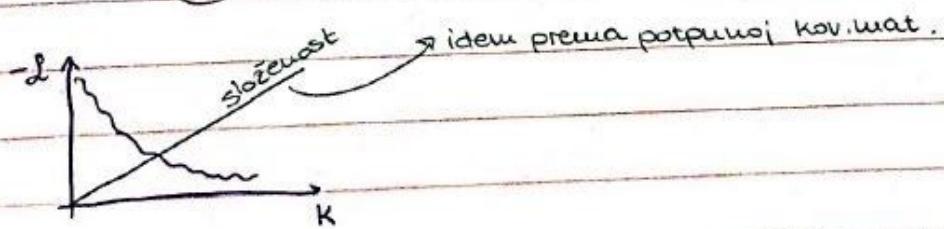
- (a) Skicirajte krivulju log-izglednosti kod EM-algoritma kao funkciju broja grupa  $K$ . Obrazložite izgled krivulje. Možete li temeljem ove krivulje odrediti optimalan broj grupa? Kako?



→ ne možemo odrediti optimalan br. grupe (eventualno met.koljena)



→ dijeli na train-test → možda dobiješ npr. x-eve za train  
pa uviđaš da je  $k=2$ , a ne  $k=3$

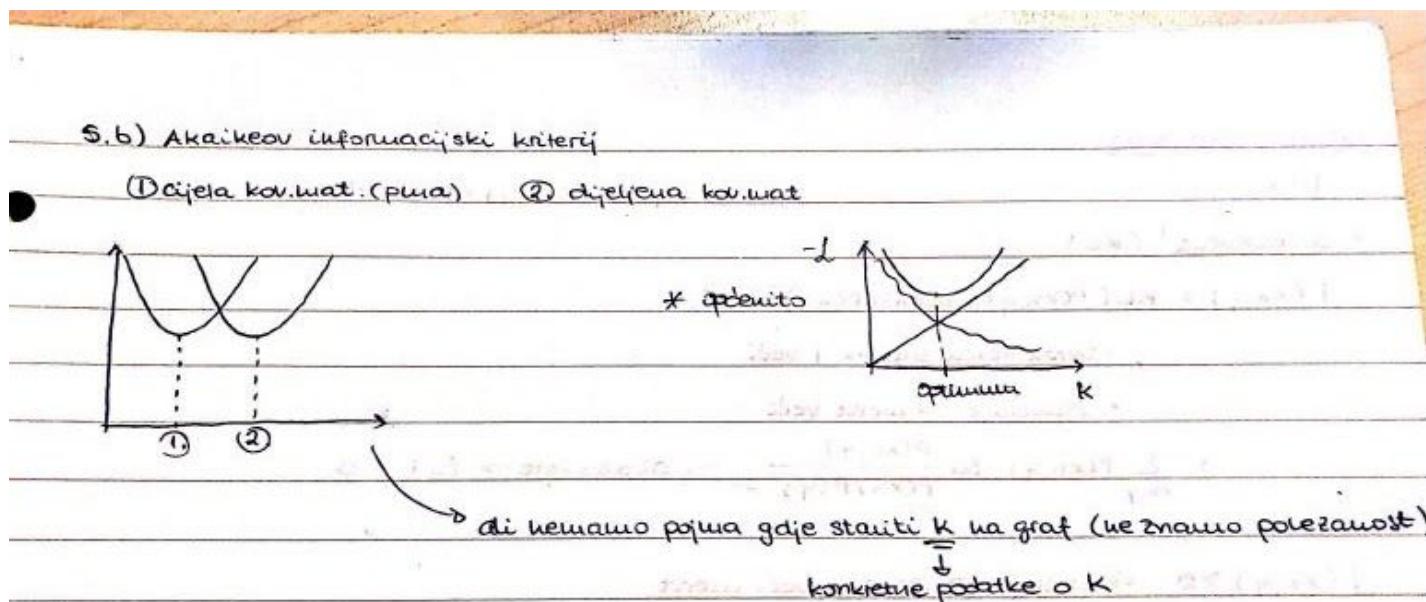


- (b) Optimizacija broja grupa  $K$  može se provesti nekim kriterijem koji kombinira funkciju pogreške (odnosno log-izglednost) i složenost modela. Takav kriterij odgovara strukturnome riziku modela, koji je minimalan za optimalan broj grupa. Jedan takav kriterij jest Akaikeov informacijski kriterij (AIC):

$$K^* = \operatorname{argmin}_K (-2 \ln \mathcal{L}(K) + 2q(K)) \quad (1)$$

gdje je  $-\ln \mathcal{L}(K)$  negativna log-izglednost podataka za  $K$  grupa, a  $q(K)$  je broj parametara modela s  $K$  grupa.

Prepostavite da podatci  $\mathcal{D}$  u stvarnosti dolaze iz  $K = 5$  grupa. Podatke grupiramo dvjema varijantama EM-algoritma: standardni algoritam i preinačeni algoritam s dijeljenom kovarijacijskom matricom (zajednička kovarijacijska matrica procijenjena nad čitavim skupom primjera  $\mathcal{D}$  na početku izvođenja algoritma). Skicirajte za ta dva algoritma funkciju koju minimizira Akaikeov minimizacijski kriterij.



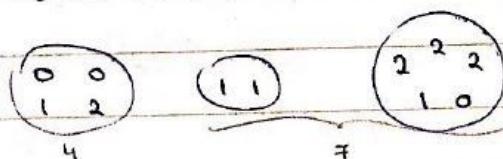
x-os je  $K$ , ali ovaj  $K=5$  ne označavamo jer ne znamo gdje je

- (c) Kada su primjeri ili podskup primjera označeni, kvaliteta grupiranja (uključivo i broj grupe  $K$ ) može se procijeniti Randovim indeksom. Randov indeks zapravo izračunava točnost s kojom ćemo par jednako označenih primjera smjestiti u istu grupu, odnosno par različito označenih primjera u različitu grupu. Izračunajte Randov indeks za sljedeću pariticiju označenih primjera (podskupovi su grupe dobivene grupiranje, a brojke su oznake klase primjera):

$$\{\{0, 0, 1, 2\}, \{1, 1\}, \{2, 2, 2, 1, 0\}\}.$$

- (d) Skicirajte vrijednost Randovog indeksa kao funkcije broja grupe  $K$ .
- (e) Randov indeks možemo koristiti samo ako su podatci označeni ili je podskup podataka označen. Međutim, čini se da to onda ujedno podrazumijeva da je unaprijed poznat broj grupe  $K$ . Imamo li koristi od Randovog indeksa čak i onda kada unaprijed znamo broj grupe? Možemo li ikako upotrijebiti Randov indeks, a da nam broj grupe nije unaprijed poznat?

5. c)  $\{\{0, 0, 1, 2\}, \{1, 1\}, \{2, 2, 2, 1, 0\}\}$



$$R = \frac{a+b}{\binom{N}{2}}$$

$N - \text{broj označenih primjera (veštački)}$   
 $\binom{11}{2} = 55$

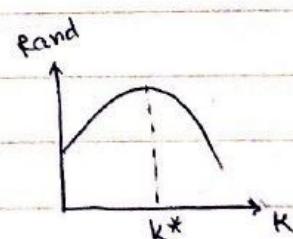
TP  
 $a \rightarrow$  ispravno u dobroj grupi      TN  
 $b \rightarrow$  ispravno u drugoj grupi

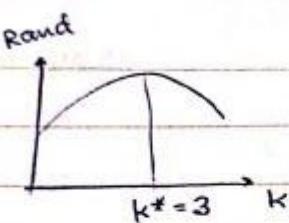
po parovima       $\begin{matrix} 0 & 0 \\ 1 & 2 \end{matrix} \rightarrow 6$  mogućih parova

$$a = 1+1+3 \longrightarrow \binom{3}{2} \quad (\text{one 3 dvojke koje su zajedno})$$

$$b = 2 \cdot 2 + 2 \cdot 4 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 2 + 2 \cdot 4$$

$$R = \frac{28+5}{55} = \frac{3}{5}$$





Zapravo  $K=5$

→ kako? npr.

- Što određuje izgled grupe? znaciške

c) Ako ne znam broj grupa, i daje mogu Randov indeks

$$R = \frac{TP + TN}{\binom{N}{2}}$$

za ovu mi ne trebaju labels jer gledam parove!  
(npr. dobijem listu parova)