16. Bayesov klasifikator II

Strojno učenje 1, UNIZG FER, ak. god. 2022./2023.

Jan Šnajder, predavanja, v3.2

Prošli puta pričali smo o Bayesovom klasifikatoru, koji modelira vjerojatnost da primjer pripada nekoj klasi i to čini pomoću Bayesovog pravila. Zapravo, rekli smo da Bayesov klasifikator modelira **zajedničku vjerojatnost** primjera i oznaka, i da se takvi modeli nazivaju **generativni modeli**. Zatim smo govorili o Gaussovom Bayesovom klasifikatoru, odnosno Bayesovom klasifikatoru kod kojeg su značajke kontinuirane i izglednosti klasa modelirane su Gaussovim gustoćama vjerojatnosti. Na kraju smo razmotrili nekoliko varijanti tog klasifikatora, koje se razlikuju u pretpostavkama o linearnoj zavisnosti značajki, kodirane matricom kovarijacije.

Danas nastavljamo s Bayesovim klasifikatorom. Zadržat ćemo se još malo na **Gaussovom Bayesovom klasifikatoru** i usporediti ga s **logističkom regresijom**. Ta usporedba otkrit će nam da su tva dva modela zapravo povezana, čime ćemo uspostaviti izravnu vezu između generativnog i diskriminartivnog strojnog učenja. Nakon toga razmotrit ćemo **naivan Bayesov klasifikator** za diskretne značajke, koji pretpostavlja uvjetnu nezavisnost između značajki unutar svake klase. Na kraju ćemo razmotriti **polunaivan Bayesov klasifikator**, koji relaksira pretpostavku o nezavisnosti, čime dobivamo složenije modele. 1

1 Bayesov klasifikator vs. logistička regresija

Na ovom predmetu volimo uočavati veze između naoko nepovezanih stvari. Zašto je to važno? Zato što su bitna načela, a načela nadilaze pojedinačne algoritme. Ako možete uočavati te veze, onda možete vidjeti što su zajedničkosti i različitosti, a to znači da vidite suštinu. Mi smo tako već uočavali razne veze između modela i algoritama. Zadnje što smo uočili jest da postoji veza između MLE-a i minimizacije pogreške, dakle između procjene parametara kao tipične tehnike za učenje generativnih modela i minimizacije pogreške kao tipične tehnike za učenje poopćenih linearnih modela.

Sada ćemo otkriti još jednu takvu zanimljivu povezanost. Pokazat ćemo da je logistička regresija zapravo kontinuirani Bayesov klasifikator, ili, obrnuto, da je kontinuirani Bayesov klasifikator zapravo poopćeni linearni model. Drugim riječima, naći ćemo točku na kojoj se spajaju dva svijeta iz strojnog učenja: generativni i diskriminativni.

Nakon ovog pretencioznog uvoda, pogledajmo o čemu se radi. Prisjetimo se najprije modela **logističke regresije**:

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

Ideja je, da krenuvši od kontinuiranog Bayesovog klasifikatora, pokušamo doći do modela logističke regresije. Ako to uspijemo, znači da su ovi modeli zapravo identični. Očito, budući da oba modela izražavaju aposteriornu vjerojatnost $P(y|\mathbf{x})$, ta vjerojatnost će nam biti pivotna točka usporedbe.

Krenimo od toga da napišemo kako Bayesov klasifikator izračunava aposteriornu vjerojatnost. Razmotrimo slučaj dvije klase, $y=1$ i $y=0$, budući da želimo uspostaviti vezu s binarnom logističkom regresijom. Aposteriorna vjerojatnost koju izračunava Bayesov klasifikator je sljedeća:

$$P(y=1|\mathbf{x}) = \frac{p(\mathbf{x}|y=1)P(y=1)}{p(\mathbf{x}|y=1)P(y=1) + p(\mathbf{x}|y=0)P(y=0)}$$

Iz ovoga nekako želimo doći do logističke funkcije $1/(1+\exp(-\alpha))$, jer bi to onda uspostavilo vezu prena logističkoj regresiji. S tim ciljem sada ćemo u nazivniku izlučiti prvi pribrojnik i pokratiti ga s brojnikom. Nadalje, u nazivniku želimo imati funkcijue exp, pa ćemo u tu svrhu na drugi pribrojnik u nazivniku primijeniti funkciju $\exp \circ \ln$ (kompoziciju funkcija exp i ln), koja je funkcija identiteta. Tako dobivamo:

$$\begin{aligned} P(y=1|\mathbf{x}) &= \frac{1}{1+\frac{p(\mathbf{x}|y=0)P(y=0)}{p(\mathbf{x}|y=1)P(y=1)}} \\ &= \frac{1}{1+\exp\left(\ln\frac{p(\mathbf{x}|y=0)P(y=0)}{p(\mathbf{x}|y=1)P(y=1)}\right)} \\ &= \frac{1}{1+\exp(-\alpha)} = \sigma(\alpha) \end{aligned}$$

gdje onda definiramo: 2

$$\alpha = \ln\frac{p(\mathbf{x}|y=1)P(y=1)}{p(\mathbf{x}|y=0)P(y=0)} = \ln\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = \ln\frac{P(y=1|\mathbf{x})}{1-P(y=1|\mathbf{x})}$$

Dobili smo definiciju modela koja oblikom odgovara onoj za logističku regresiju. Međutim, da bi korespondencija bila potpuna, mora vrijediti:

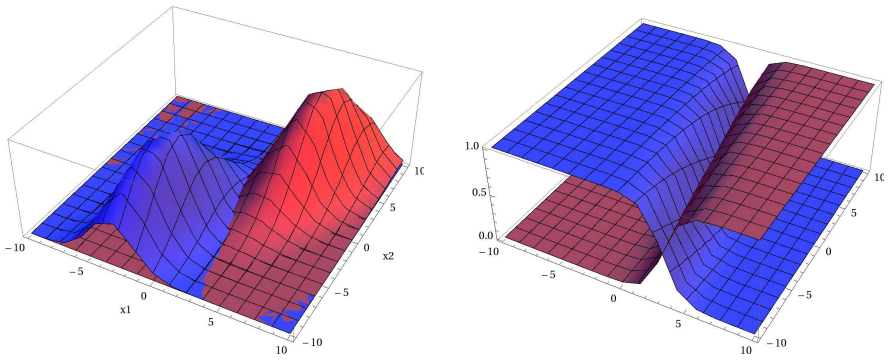
$$\alpha = \ln\frac{p(\mathbf{x}|y=1)P(y=1)}{p(\mathbf{x}|y=0)P(y=0)} = \underbrace{\ln p(\mathbf{x}|y=1)P(y=1)}_{h_1(\mathbf{x})} - \underbrace{\ln p(\mathbf{x}|y=0)P(y=0)}_{h_0(\mathbf{x})} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

tj. razlika $h_1(\mathbf{x}) - h_0(\mathbf{x})$ mora biti linearna funkcija od $\mathbf{x}$. Drugim riječima, granica između klasa $y=1$ i $y=0$ mora biti linearna!

Sada se prisjetimo što smo naučili prošli put: tada smo utvrdili da uz će, uz određenu pretpostavku na kovarijacijsku matricu $\mathbf{\Sigma}$ za gustoću vjerojatnosti izglednosti klase $p(\mathbf{x}|y)$, Gaussov Bayesov klasifikator dati linearnu granicu, tj. da će iz kvadratnog modela degenerirati u linearan model. Koja je to bila pretpostavka? Odgovor je: pretpostavka **dijeljene kovarijacijske matrice**. Naime, prisjetimo se, ako je kovarijacijska matrica dijeljena, granica između dviju klasa je:

$$\begin{aligned} h_{10}(\mathbf{x}) &= h_1(\mathbf{x}) - h_0(\mathbf{x}) \\ &= \mathbf{x}^{\mathrm{T}} \underbrace{\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)}_{\mathbf{w}} \underbrace{-\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_0^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_0 + \ln\frac{P(y=1)}{P(y=0)}}_{w_0} \\ &= \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 \end{aligned}$$

Krenuli smo od Gaussovog Bayesovog klasifikatora za dvije klase te smo, uz pretpostavku dijeljene kovarijacijske matrice stigli do logističke regresije. Time smo pokazali da je Gaussov Bayesov klasifikator s dijeljenom kovarijacijskom matricom zapravo ista stvar kao i (neregularizirana) binarna logistička regresija. 3 Što to zapravo znači? Znači da, ako radimo binarnu klasifikaciju, istu granicu možemo dobiti (neregulariziranom) logističkom regresijom ili Gaussovim Bayesovim klasifikatorom! To lijepo ilustrira sljedeći primjer u dvodimenzijskom ulaznom prostoru:



Lijeva slika prikazuje zajedničku gustoću vjerojatnosti, $p(\mathbf{x}, y)$, za dvije klase (plava i crvena), dok desna slika prikazuje aposteriornu vjerojatnost, $p(y|\mathbf{x})$, za iste te dvije klase. Aposteriorna se vjerojatnost može izračunati iz zajedničke vjerojatnosti pomoću Bayesovog pravila, i tako to radi Bayesov klasifikator. Međutim, logistička regresija izravno izračunava aposteriornu vjerojatnost. U konačnici, međutim, granica između ovih dviju klasa je identična.

Ovime smo povezali generativni model (Gaussov Bayesov klasifikator) s njemu odgovarajućim diskriminativnim modelom (logistička regresija). U strojnom učenju ima više takvih **generativno-diskriminativnih parova** modela. Gaussov Bayesov klasifikator i logistička regresija jedan su primjer. Drugi primjeri generativno-diskriminativnog para jesu **skriveni Markovljev model** (koji ćemo spomenuti idući put) i **model uvjetnih slučajnih polja** (engl. *conditional random field*), koji nećemo raditi.

Osim povezivanja generativnog i diksriminativnog, ovime smo zapravo napokon dobili i potpuno vjerodostojano opravdanje za probabilističku interpretaciju izlaza logističke regresije. Sjetite se da smo, kada smo pričali o logističkoj regresiji, rekli da koristimo sigmoidnu funkciju kako bi izlaz modela bio ograničen na interval $(0, 1)$, i kako bismo onda tu vrijednost mogli tumačiti kao vjerojatnost oznake $y = 1$ za primjer $\mathbf{x}$. Međutim, nije bilo jasno uz koje zapravo pretpostavke možemo tako tumačiti izlaz modela (samo zato što je neki broj u intervalu $[0, 1]$ ne znači automatski da odgovara vjerojatnosti nekog događaja). No, sada je to jasno: ako pretpostavimo da su (1) primjeri iz obiju klasa normalno distribuirani oko srednje, prototipne vrijednosti (tj. izglednost je Gaussova gustoća vjerojatnosti) i (2) da postoji linearna zavisnost između izvora šuma koja je u obje klase identična (tj. kovarijacijska matrica je dijeljena), onda izlaz logističke regresije doista odgovara aposteriornoj vjerojatnosti oznake y za primjer $\mathbf{x}$. S druge strane, ako ove pretpostavke ne vrijede, onda nemamo teorijski model uz koje bi izlaz logističke regresije odgovarao aposeriornoj vjerojatnosti. Međutim, u praksi se time previše ne zamaramo, tj. izlaz logističke regresije tumačimo kao vjerojatnost neovisno o tome koliko podatci doista odgovaraju navedenim pretpostavkama. No, to također u praksi znači da, ako postoji veliko odstupanje od ovih pretpostavki (npr. kovarijacije su bitno različite u dvjema klasama), onda će naš model loše raditi (bit će podnaučen).

Da sažmemo: logistička regresija je **diskriminativan model** koji izravno modelira aposteriornu vjerojatnost (ovdje smo w_0 uključili u vektor $\mathbf{w}$):

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

dok je Bayesov klasifikator njoj odgovarajući **generativni model** koji tu istu vjerojatnost modelira neizravno:

$$P(y = 1|\mathbf{x}) = \frac{p(\mathbf{x}|y = 1)P(y = 1)}{p(\mathbf{x}|y = 1)P(y = 1) + p(\mathbf{x}|y = 0)P(y = 0)}$$

Zadnje opažanje koje ćemo napraviti tiče se broja parametara modela. Koliko parametara imaju ovi modeli za n-dimenzijski ulazni prostor? Logistička regresija ima samo vektor $\mathbf{w}$ kao parametre, pa dakle logistička regresija ima $n+1$ parametar. S druge strane, Bayesov klasifikator

ima $\frac{n}{2}(n+1)+2n+1$ parametara (kovarijacijsku matricu $\mathbf{\Sigma}$, dva vektora srednjih vrijednosti $\boldsymbol{\mu}_j$ te jedan parametar apriorne Bernoullijeve vjerojatnosti). Ovo ilustrira tipičnu situaciju: generativni modeli koji ostvaruju istu složenost (dakle isti oblik granice u ulaznome prostoru) općenito imaju više parametara od njihovog diskriminativnog para. Zbog toga generativni modeli općenito trebaju više primjera za učenje nego diskriminativni modeli, odnosno, zbog toga diskriminativni modeli općenito rade bolje od generativnih na istom skupu podataka.

Nakon ove nešto šire slike na strojno učenje, vratimo se opet na Bayesov klasifikator. Sada kada znamo kako radi Bayesov klasifikator za kontinuirane značajke, razmotrimo Bayesov klasifikator za diskretne značajke, krenuvši od **naivnog Bayesovog klasifikatora**.

2 Naivan Bayesov klasifikator

2.1 Model Naivnog Bayesovog klasifikatora

Prisjetimo se, općenit model Bayesovog klasifikatora je:

$$P(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)P(y)}{\sum_{y'} p(\mathbf{x}|y')P(y')} = \frac{p(x_1,\ldots,x_n|y)P(y)}{\sum_{y'} p(x_1,\ldots,x_n|y')P(y')}$$

Kod diskretnog klasifikatora, značajke su diskretne, dakle varijable x_k su **kategoričke (multinulijeve)** varijable (ili **Bernoullijeve**, ako imaju samo dvije moguće vrijednosti).

Sad je pitanje: kako ćemo točno modelirati izglednost klase $p(\mathbf{x}|y)$, ako znamo da su x_k kategoričke/Bernoullijeve varijable? To će zapravo biti jedina razlika u modelu u odnosu na Gaussov Bayesov klasifikator, kod kojega smo, prisjetite se, izglednosti klasa modelirali multivarijatnom Gaussovom distribucijom. Prva mogućnost koja bi nam mogla pasti na pamet jest da $p(\mathbf{x}|y)$ tretiramo kao **kategoričku razdiobu**, čije su vrijednosti sve moguće kombinacije pojedinačnih kategoričkih varijabli, tj. pojedinačnih značajki. Pogledajmo primjer.

▸ **PRIMJER**

Imamo tri značajke: prve dvije značajke imaju tri moguće vrijednosti, $K_1 = 3$ i $K_2 = 3$, dok je treća značajka binarna, $K_3 = 2$. Budući da su to kategoričke varijable, prikazat ćemo ih kao binarne vektore indikatorskih varijabli duljine 3, 3, odnosno 2. Te vektore konkateniramo u jedan vektor, $\mathbf{x}$. Npr., vektor:

$$\mathbf{x} = (\underbrace{0,1,0}_{x_1}, \underbrace{0,0,1}_{x_2}, \underbrace{1,0}_{x_3})$$

odgovara kombinaciji vrijednosti $x_1 = 1$ (od mogućih vrijednosti $\{0,1,2\}$), $x_2 = 2$ (od mogućih vrijednosti $\{0,1,2\}$) i $x_3 = 0$ (od mogućih vrijednosti $\{0,1\}$). Vektor $\mathbf{x}$ sada možemo tretirati kao jednu kategoričku varijablu, koja ima $3 \times 3 \times 2 = 18$ mogućih različitih vrijednosti. Tomu odgovarajuća kategorička distribucija ima onoliko parametara μ_k koliko ima različitih vrijednosti. Da bismo naučili takvu distribuciju, trebamo dakle procijeniti svih 18 parametara μ_k. Zapravo, dovoljno je da procijenimo njih 17, jer znamo da mora vrijediti $\sum \mu_k = 1$. Međutim, ovu procjenu moramo napraviti za svaku oznaku klase y. Ako imamo npr. $K = 3$ klase, onda moramo procijeniti ukupno $17 \times 3 = 51$ parametar. Te procjene za μ_k mogu se prikazati kao **tablica uvjetne vjerojatnosti** (engl. *conditional probability table; CPT*). U ovom konkretnom slučaju CPT bi izgledala ovako:

k	x_1	x_2	x_3	y	$\mu_k = p(\mathbf{x}\|y)$
1	0	0	0	0	...
2	0	0	1	0	...
3	0	1	0	0	...
⋮		⋮		⋮	⋮
17	2	2	0	0	...
18	0	0	0	1	...
⋮		⋮		⋮	⋮
34	2	2	1	1	...
35	0	0	0	2	...
⋮		⋮		⋮	⋮
51	2	2	1	2	...

gdje bismo umjesto "..." imali neke konkretne vrijednosti, koje se za svaku klasu y zbrajaju u 1. Primijetite da smo zbog toga za svaku klasu uštedjeli jedan redak, znajući da se vjerojatnosti $p(\mathbf{x}|y)$ za svaku klasu y moraju zbrajati u 1.

Općenito, ako pojedinačne kategoričke varijable x_k imaju svaka K_k mogućih vrijednosti, ukupno imamo $\prod_{k=1}^{n} K_k$ različitih vrijednosti. Trebamo procijeniti te vrijednosti za svaku od K klasa. Ukupan broj parametara distribucije, odnosno broj redaka u tablici uvjetne vjerojatnosti, jednak je:

$$K \cdot (\prod_{k=1}^{n} K_k - 1)$$

($K_k - 1$ jer se vjerojatnosti $P(\mathbf{x}|y)$ za svaki pojedini y moraju sumirati na jedinicu.)

Ovdje sad odmah vidimo i što je problem s ovakvim modeliranjem izglednosti klase. Zapravo, postoje dva problema. Prvi je problem očit: **velik broj parametara**. Naime, broj parametara raste eksponencijalno s brojem mogućih vrijednosti pojedinačnih značajki.

▸ **PRIMJER**

Neka su varijable x_k **binarne**. Npr., radimo klasifikaciju crno-bijelih slika, pa je svaka varijabla x_k jedan piksel. Imamo n značajki, tj. n piksela. Recimo da je riječ o binarnoj klasifikaciji, tj. imamo $K = 2$ klasa. Koliko parametara moramo procijeniti za modeliranje distribucije $P(x_1, x_2, \ldots, x_n|y)$, tj. koliko će redaka imati CPT?

k	x_1	...	x_{n-1}	x_n	y	$\mu_k = p(\mathbf{x}\|y)$
1	0	...	0	0	0	...
2	0	...	0	1	0	...
3	0	...	1	0	0	...
⋮	⋮	...	⋮	⋮	⋮	⋮
2^{n-1}	0	...	1	0	0	...
2^n	0	...	1	0	1	...
⋮	⋮	...	⋮	⋮	⋮	⋮
$2 \cdot (2^n - 1)$	0	...	1	0	1	...

Ukupno imamo $2 \cdot (2^n - 1)$ redaka. Drugim riječima, broj parametara **eksponencijalno** ($\mathcal{O}(2^n)$) ovisi o broju značajki. To je u stvarnosti potpuno neprihvatljivo! Npr., za klasifikaciju binarnih slika dimenzija 100×100 u dvije klase, trebalo bi nam 19998 parametara, što je neprihvatljiva složenost. Trebalo bi nam vrlo mnogo primjera da dobro naučimo takav model. (Zašto? Zato što bismo za svaku kombinaciju vrijednosti značajki $x_1, \ldots, x_n$, tj. za svaki redak CPT-a, morali imati dovoljno primjera s točno takvom kombinacijom vrijednosti značajki, a da bismo mogli dovoljno dobro procijeniti parametar μ_k kategoričke distribucije.)

Dakle, prvi problem, ako $p(\mathbf{x}|y)$ modeliramo kao kategoričku distribuciju gdje jednostavno sve vektore pojedinačnih varijabli konkateniramo u jedan golemi vektor, jest da imamo previše parametara. Drugi je problem povezan s time, ali je zapravo fundamentalniji: takav model uopće **ne može generalizirati**. Zašto? Zato što ćemo vjerojatnosti moći procijeniti samo za one primjere koje smo vidjeli (koje imamo u skupu za učenje). Ukupna masa vjerojatnosti bit će raspodijeljena samo na te primjere. Svi ostali primjeri – a to su neviđeni primjeri – imat će $\mu_k = 0$. To znači da je vjerojatnost neviđenog primjera jednaka nuli. Ako pogledate Bayesovo pravilo, to znači da će nam i brojnik i nazivnik za takve primjere biti jednaki nuli, pa aposteriorna vjerojatnost uopće neće biti definirana. Očito, to je vrlo loša generalizacija na neviđene primjere!

Kako riješiti ovaj problem? Ako model ne može dobro generalizirati, to znači da trebamo pojačati **induktivnu pristranost**, tj. uvesti još pretpostavki. Kod generativnih modela to možemo učiniti tako da **pojednostavimo** gustoću vjerojatnosti $p(\mathbf{x}|y)$ (odnosno vjerojatnost $P(\mathbf{x}|y)$, za slučaj diskretnih značajki). To ćemo ostvariti tako da tu vjerojatnost na prikladan način **faktoriziramo**.

Pogledajmo sada što znači faktorizirati vjerojatnost. Prošli smo puta uveli dva osnovna pravila teorije vjerojatnosti: pravilo zbroj i pravilo umnoška. Zatim smo upotrijebili ta dva pravila da bismo izveli Bayesovo pravilo. Još jedno pravilo koje možemo izvesti, i to jednostavnom primjenom pravila umnoška, jest **pravilo lanca** (engl. *chain rule*). Za zajedničku vjerojatnost sa tri varijable, to pravilo izgleda ovako:

$$P(x, y, z) = \underbrace{\underbrace{P(x)P(y|x)}_{P(x,y)} P(z|x, y)}_{P(x,y,z)}$$

Vidimo da zajedničku vjerojatnost od tri varijable možemo napisati kao umnožak triju vjerojatnosti. Umnožak se sastoji od marginalne vjerojatnosti jedne varijable (ovdje je to varijabla x) te uvjetnih vjerojatnosti za preostale dvije varijable, gdje u uvjetni dio dodajemo sve više varijabli. Pritom je redoslijed kojim to radimo proizvoljan, tj.:

$$P(x, y, z) = P(x)P(y|x)P(z|x, y) = P(z)P(y|z)P(x|y, z) = P(z)P(x|z)P(y|x, z) = \ldots$$

(ima ukupno $3! = 6$ mogućnosti). Ovo je bio primjer s tri varijable, ali princip naravno vrijedi i općenito, za n-dimenzijski slučajan vektor. Općenito, pravilo lanca za zajedničku vjerojatnost od n varijabli je:

$$P(x_1, \ldots, x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \cdots P(x_n|x_1, \ldots, x_{n-1}) = \prod_{k=1}^{n} P(x_k|x_1, \ldots, x_{k-1})$$

Pravilo lanca nam je vrlo korisno kada zajedničku vjerojatnost sastavljenu od mnogo varijabli želimo dekomponirati na umnožak više jednostavnijih uvjetnih vjerojatnosti. Kada zajedničku vjerojatnost dekomponiramo na umnožak više vjerojatnosti, onda se pojedinačne vjerojatnosti zovu **faktori**, a "rastavljanje" zajedničke vjerojatnosti na faktore zove se, očekivano, **faktorizacija**.

Sada kada ovo znamo, primjenom pravila lanca, izglednost klase mogli bismo faktorizirati na sljedeći način:

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

Ovo je identično gore navedenom pravilu lanca, samo što smo cijelu vjerojatnost uvjetovali varijablom y, tj. oznakom klase. To samo znači da u gornjoj jednakosti varijablu y jednostavno dodajemo u uvjetni dio vjerojatnosti i na lijevoj i na desnoj strani.

Sada znamo kako izglednost klase $P(x_1, \ldots, x_n|y)$ raspisati kao umnožak više uvjetnih vjerojatnosti. Međutim, nažalost moramo utvrditi da ovime nismo napravili baš nikakvo pojednostavljenje: to je samo drugačiji način zapisa zajedničke vjerojatnosti za $\mathbf{x}$. Broj parametara je isti kao i ranije. Kako bismo pojednostavili model, moramo uvest dodatne pretpostavke o **uvjetnoj nezavisnosti varijabli**. Konkretno, uvest ćemo vrlo radikalnu pretpostavku:

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

tj. za svaki faktor prepostavit ćemo da je uvjetovan samo oznakom y. Drugim riječima, to znači da smo pretpostavili da su značajke x_k za primjere unutar svake klase y međusobno nezavisne. Pogledat ćemo uskoro detaljnije što to točno znači. Međutim, već sada vidimo da se, uz ovu pretpostavku, izglednost faktorizira ovako:

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

I to nas onda konačno dovodi do modela **naivnog Bayesovog klasifikatora** (engl. *naïve Bayes classifier*):

$$h(x_1, \ldots, x_n) = \operatorname*{argmax}_y P(y) \prod_{k=1}^{n} P(x_k|x_1, \ldots, x_{k-1}, y) = \operatorname*{argmax}_y \; P(y) \prod_{k=1}^{n} P(x_k|y)$$

gdje smo, dakle, kod druge jednakosti iskoristili prepostavku o uvjetnoj nezavisnosti značajki za zadanu klasu. Primijetimo još da nam ovako definiran model ne daje vjerojatnost (jer nedostaje nazivnik Bayesovog pravila), pa, ako želimo vjerojatnost, trebamo normalizirati sukladno Bayesovom pravilu. 5

Dakle, uz pretpostavku uvjetne nezavisnosti značajki za zadanu klasu, izglednost smo faktorizirali na n faktora, svaki od kojih se bavi samo jednom značajkom. Je li tako faktorizirani model jednostavniji od nefaktoriziranog modela (ili, što je ekvivalentno, od modela gdje smo faktorizaciju proveli bez dodatnih pretpostavki, primjenjujući samo pravilo lanca)? Naravno da je! Pogledajmo koliko nam parametara treba za jedan faktor $P(x_k|y)$. Ako kategorička varijabla x_k poprima K_k mogućih vrijednosti, a klasificiramo u K klasa, faktor ćemo prikazati tablicom uvjetne vjerojatnosti (CPT) s ovoliko parametara:

$$(K_k - 1) \cdot K$$

Za ukupno n značajki imamo n faktora, pa je ukupan broj parametara modela naivnog Bayesovog klasifikatora, kada se još uračunaju parametri apriorne distribucije klasa, jednak:

$$\sum_{k=1}^{n} (K_k - 1) \cdot K + K - 1$$

Primijetite razliku u odnosu na nefaktoriziranu razdiobu: tamo smo morali imati po jedan parametar za svaku kombinaciju, kojih je bilo eksponencijalno mnogo u broju značajki. Ovdje imamo **linearnu ovisnost** u broju značajki!

2.2 Učenje naivnog Bayesovog klasifikatora

Izvrsno, definirali smo model naivnog Bayesovog klasifikatora! Sad je pitanje kako ga naučiti. Prisjetimo se: (naivan) Bayesov klasifikator je probabilistički model. Što znači naučiti probabilistički model? To znači **procijeniti parametre**. Dakle, trebamo procijeniti parametre za sve distribucije koje se koriste u modelu. Konkretno, trebamo procijeniti parametre apriorne distribucije $P(y)$ i parametre za izglednosti klasa, i to, budući da smo izglednost faktorizirali,

tu procjenu trebamo napraviti posebno za svaki faktor $P(x_k|y)$. Procjena svih ovih parametara je vrlo jednostavna. Za apriornu distribuciju možemo koristiti procjenitelj MLE:

$$P(y=j) = \hat{\mu}_j = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = j\} = \frac{N_j}{N}$$

tj. to je **relativna frekvencija** klase j u skupu svih primjera. Slično, za faktor $P(x_k|y)$, procjena MLE je:

$$P(x_k|y=j) = \hat{\mu}_{k,j} = \frac{\sum_{i=1}^{N} \mathbf{1}\{x_k^{(i)} = x_k \wedge y^{(i)} = j\}}{\sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = j\}} = \frac{N_{kj}}{N_j}$$

tj. to je relativna frevencija vrijednosti x_k u svim primjerima označenima sa $y = j$.

Međutim, za procjenu parametara faktora $P(x_k|y)$ nam nije pametno koristiti procjenitelj MLE. Zašto? Problem je u tome što se lako može dogoditi da u nekoj od klasa neka značajka baš nikada ne poprimi neku vrijednost. Sličan problem imali smo kada smo sve značajke konkatenirali u jedan vektor. Ovdje doduše imamo samo jednu varijablu, a ne cijeli vektor, pa je vjerojatnost da nam se dogodi da se dotična kombinacija značajke i oznake klase ne pojavljuje u skupu za učenje sigurno manja nego da nam se to dogodi za cijeli vektor značajki, međutim ipak se to u praksi događa. U tom slučaju će procjena MLE za tu kombinaciju biti jednaka nula. To efektivno znači da tu kombinaciju smatramo nemogućom. Što se onda događa kad kod predikcije dođe primjer koji ima baš tu kombinaciju? Onda je vjerojatnost tog faktora jednaka nula. Budući da se faktori međusobno množe, to će aposteriorna vjerojatnost klase za taj primjer biti jednaka nuli!

Kako ćemo riješiti taj problem? Tako da ne dopustimo da vjerojatnosti budu jednake nuli. Kako? Tako da radimo **zaglađivanje** (engl. *smoothing*) procjena, što znači da zapravo umjesto procjenitelja MLE koristimo procjenitelj MAP. Najjednostavnije je da koristimo **Laplaceov procjenitelj** (za koji, dakako, još od prošlog tjedna znamo da je MAP procjenitelj s Dirichletovom distribucijom kao apriornom distribucijom i hiperparametrima $\alpha_k = 2$):

$$P(x_k|y=j) = \hat{\mu}_{k,j} = \frac{\sum_{i=1}^{N} \mathbf{1}\{x_k^{(i)} = x_k \wedge y^{(i)} = j\} + 1}{\sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = j\} + K_k} = \frac{N_{kj} + 1}{N_j + K_k}$$

Ovime smo definirali algoritam naivnog bayesovog klasifikatora: definirali smo model, koji koristi pretpostavku o uvjetnoj nezavisnosti značajki unutar zadane klase, te optimizacijski postupak, a to je MLE (za apriorne vjerojatnosti) odnosno MAP (za izglednosti klase). Funkciju gubitka nismo eksplicitno definirali: ona je implicitna u postupku MLE odnosno MAP. 6

Naivan Bayesov klasifikator jednostavan je i učinkovit algoritam. Zovemo ga naivnim zbog pretpostavke o uvjetnoj nezvisnosti značajki unutar klase. Ta pretpostavka je nekada prenaivna, i onda umjesto naivnog želimo koristiti **polunaivan Bayesov klasifikator** (engl. *semi-naïve Bayes classifier*), koji relaksira neke od pretpostavki uvjetne nezavisnosti. U nastavku ćemo razmotriti polunaivni Bayesov klasifikator. Međutim, prije nego što to napravimo, pogledajmo detaljnije što je to uopće uvjetna nezavisnost.

3 Uvjetna nezavisnost

Što, zapravo, znači da su varijable **uvjetno nezavisne**? Prisjetimo se najprije što znači da su varijable **(marginalno) nezavisne**:

$$P(X, Y) = P(X) \cdot P(Y)$$

što se može napisati kao:

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

Ovaj drugi oblik puno je intuitivniji: varijabla X nezavisna je od varijable Y ako poznavanje ishoda varijable Y ne utječe na vjerojatnosti ishoda varijable X (i obrnuto). Da su varijable X i Y nezavisne (marginalno nezavisne) označavamo s $X \perp Y$.

Uvjetna nezavisnost znači da dvije varijable, X i Y, postaju nezavisne ako nam je poznat ishod neke treće varijable, Z. To znači da vrijedi:

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

što je ekvivalentno sa:

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

Da su dvije varijable, X i Y, uvjetno nezavisne, uvjetovano na varijablu Z, označavat ćemo s $X \perp Y|Z$. Pogledajmo primjer.

▸ **PRIMJER**

Razmotrimo situaciju upisa studenata na elitne fakultete. Neka:

$$X = \text{“studentica je primljena na FER”}$$
$$Y = \text{“studentica je primljena na PMF-MO”}$$

U stvarnom životu, iskustveno, ako znamo da se ostvario X, onda to mijenja vjerojatnost našeg znanja da se ostvario Y. Tj.:

$$P(Y|X) \neq P(Y)$$

odnosno varijable nisu marginalno nezavisne, što pišemo kao $X \not\perp Y$. S druge strane, neka:

$$Z = \text{“studentica je sudjelovala na matematičkim olimpijadama”}$$

Ako znamo da se ostvario Z, onda to objašnjava prijem na oba faksa, pa spoznaja o X više ne utječe na Y, tj. vrijedi:

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

Drugim riječima, varijable X i Y su uvjetno nezavisne uz Z, što pišemo kao $X \perp Y|Z$.

Vratimo se naivnom Bayesovom klasifikatoru. Što mislite, vrijedi li općenito ta uvjetna nezavisnost, npr. za dvije značajke: $x_i \perp x_k|y$? Pogledajmo jedan primjer koji će nam pomoći rasvijetliti to pitanje.

▸ **PRIMJER**

Radimo klasifikaciju novinskog teksta u tematske rubrike. Želimo izgraditi naivan Bayesov klasifikator koji novinski tekst, sastavljen od riječi, klasificira u jednu od tri rubrike: sport, politika, kriminal (to ionako pokriva većinu tema dnevnog tiska). Imamo, dakle, primjer $\mathbf{x}$, koji je je sastavljen od riječi, i oznake y iz skupa {sport, politika, kriminal}. Značajke neka indiciraju prisustvo pojedine riječi u novinskome tekstu. Konkretno, pogledajmo ove tri značajke:

$$x_1 = \mathbf{1}\{\text{“rezultat”} \in \mathbf{x}\}$$
$$x_2 = \mathbf{1}\{\text{“lopta”} \in \mathbf{x}\}$$
$$x_3 = \mathbf{1}\{\text{“gol”} \in \mathbf{x}\}$$

To jest, značajka x_1 će biti jednaka 1 ako u novinskome tekstu negdje pojavljuje riječ “rezultat”, a inače će biti jednaka 0. Značajke x_2 i x_3 funkcioniraju identično, za riječi “lopta” odnosno “gol” (ova zadnja kao imenica, ne kao pridjev). 7

Naivan Bayesov klasifikator pretpostavlja $x_1 \perp x_2|y$ i $x_2 \perp x_3|y$. Provjerimo bi li ove pretpostavke doista vrijedile u praksi. Pogledajmo prvo je li $x_1 \perp x_2|y$. Za riječi "rezultat" i "lopta" možemo očekivati da će se pojavljivati za $y =$ sport, pa nas dakle zanima vrijedi li:

$$P(\text{rezultat}|\text{sport}) = P(\text{rezultat}|\text{lopta}, \text{sport})$$

Intuitivno, čini se da ovo vrijedi, jer čim se u novinskom tekstu spominje sport, onda to određuje i vjerojatnost da se spomene rezultat, a spominjanje lopte ne utječe na tu vjerojatnost. Dakle, zaključujemo da bi ove varijable u praksi doiste mogle biti uvjetno nezavisne, dakle:

$$x_1 \perp x_2|y$$

Primijetimo, međutim, da ove dvije varijable nisu marginalno nezavisne. Naime:

$$P(\text{rezultat}) \neq P(\text{rezultat}|\text{lopta})$$

jer spominjanje lopte u tekstu općenito povećava vjerojatnost spominjanja rezultata u tekstu.

Pogledajmo sada vrijedi li $x_2 \perp x_3|y$?. Pitamo se, vrijedi li:

$$P(\text{lopta}|\text{sport}) = P(\text{lopta}|\text{gol}, \text{sport})$$

Čini se da ovo ne vrijedi: ako je $y =$ sport, onda vjerojatnost da se u tekstu spomene "lopta" ovisi o tome je li se u tekstu spomeno "gol" (konkretno, vjerojatnost za "lopta" raste, ako se u tekstu spomeno "gol", i obrnuto). Prema tome, varijable x_2 i x_3 nisu uvjetno nezavisne, tj.:

$$x_2 \not\perp x_3|y$$

Vidimo, dakle, da uvjetna nezavisnost značajki unutar neke klase nekada vrijedi, a nekada ne vrijedi. Istina je da u praksi **savršena uvjetna nezavisnost rijetko vrijedi**. Ipak, unatoč tome što ne vrijedi, pokazuje se da ta pretpostavka daje modele koji sasvim dobro funkcioniraju.

Ali ipak, što ako doista postoji jaka uvjetna zavisnost između varijabli, kao u ovom primjeru između riječi "lopta" i "gol"? Ako znamo da su neke varijable jako uvjetno zavisne, onda je bolje da budemo manje naivni i da ne pretpostavljamo uvjetnu nezavisnost. Naravno, ekstrem bi bio da ništa ne pretpostavimo, ali to smo već vidjeli da ne funkcionira (dobivamo prenaučen model koji nikako ne generalizira). Umjesto toga, ideja bi bila da samo za neke parove varijabli ne pretpostavimo uvjetnu nezavisnost. Tako dobivamo **polunaivan Bayesov klasifikator**. Pogledajmo to malo detaljnije.

4 Polunaivan Bayesov klasifikator

Motivirajmo polunaivan Bayesov klasifikator našim ranijim primjerom. Ako, na primjer, ne vrijedi $x_2 \perp x_3|y$, jer, npr., pojavljivanje riječi "lopta" i "gol" nije nezavisno za neku klasu, onda bi nam bilo pametnije da zajedničku vjerojatnost ne faktoriziramo kao

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

nego kao

$$P(x_1, x_2, x_3, y) = P(x_1|y)\underline{P(x_2, x_3|y)}P(y)$$

Ovdje je potcrtan "združeni faktor", koji nismo do kraja faktorizirali, tj. faktor kojim modeliramo zajedničku vjerojatnost varijabli x_2 i x_3 (uvjetovano na y).

Što bi bila prednost da model definiramo na ovakav način? Prednost je to što bolje modeliramo zavisnost koja postoje među varijablama x_2 i x_3, pa će model biti točniji. Što bi bio nedostatak? To što model postoje **složeniji**, tj. broj parametara raste. Naime, broj parametara

za združeni faktor je $(K_3 \cdot K_2 - 1) \cdot K$, te on dakle ovisi o ukupnom broju kombinacija mogućih vrijednosti varijabli x_2 i x_3.

Iz ovoga se nekako nameće zaključak da ima smisla združiti neke varijable, kako bismo dobili složeniji model, ali opet ne sve varijable, jer onda dobivamo presložen model. Ključno je, dakle, pitanje koje varijable združiti? To možemo formulirati kao **problem pretraživanja prostora stanja**: razmatramo sva moguća združivanja i po nekom kriteriju odaberemo optimalno združivanje. Pogledajmo najprije koliko mogućnosti za združivanje uopće postoji? Broj mogućnosti jednak je broju svih **particija**. Npr., za tri varijable, a, b i c, moguće particije su:

$$\{\{a\},\{b\},\{c\}\}$$
$$\{\{a\},\{b,c\}\}$$
$$\{\{b\},\{a,c\}\}$$
$$\{\{c\},\{a,b\}\}$$
$$\{\{a,b,c\}\}$$

Ukupan broj particija je **Bellov broj**. Bellov broj za skup od tri elementa je $B_3 = 5$, od četiri $B_4 = 15$, od pet $B_5 = 52$, a od deset $B_{10} = 115975$. Očito, to je previše mogućnosti za iscrpno pretraživanje. Dakle, treba nam **heurističko pretraživanje**. Kod tog pretraživanja, svaka particija odgovara jednom stanju. Pretraga kreće od stanja s potpuno odvojenim varijablama, u svakom koraku združujemo neke varijable, i na taj način pretražujemo prostor stanja u potrazi za optimalnim združivanjem. Treba nam još kriterij pretraživanja (odnosno heurstika), kojim ćemo ocijeniti koliko je neko stanje (odnosno particija) dobra. Taj kriterij treba nam odgovoriti na pitanje je li je li bolje združiti varijable x_j i x_k u zajednički faktor $P(x_j, x_k|y)$ ili ih je bolje ostaviti odvojenima kao dva zasebna faktora, $P(x_j|y)$ i $P(x_k|y)$? Za kriterij združivanja imamo dvije mogućnosti: 8

1. Koristimo unakrsnu provjeru te isprobavamo **točnost** modela na skupu za provjeru i združujemo one varijable koje povećavaju točnost. Primjer takvog algoritma je algoritam **FSSJ (Forward Sequential Selection and Joining)**;
2. Mjerimo **zavisnost** varijabli i združujemo one varijable koje su najviše zavisne. Primjeri algoritama koji tako funkcioniraju su **TAN** i k-**DB**.

U nastavku je pseudokod algoritma FSSJ.

▸ **Algoritam FSSJ**

1. Inicijaliziraj $X = \varnothing$. Početna faktorizacija:

$$P(x_1, \ldots, x_n, y) = P(x_1) \cdots P(x_n) P(y)$$
$$P(y|x_1, \ldots, x_n) = P(y)$$

   Klasificiraj primjere iz skupa za provjeru: $y^* = \text{argmax}_j P(y = j)$

2. Za svaku varijablu $x_k \notin X$ koja još nije uključena u model, razmotri:
   (a) Uključi x_k kao uvjetno nezavisnu u odnosu na ostale varijable za danu klasu j
   (b) Uključi x_k tako da se ona doda u zajednički faktor s nekom već uključenom varijablom
3. Izaberi x_k i opciju koja minimizira pogrešku generalizacije
4. Ponavljaj od koraka (2) do konvergencije pogreške

Drugi pristup je da nekako mjerimo koje su varijable uvjetno zavisne, pa da onda njih združimo. Ovdje se postavlja pitanje kako općenito mjeriti (ne)zavisnost varijabli? Zavisnost između varijabli mogli bismo pokušati izmjeriti Pearsonovim koeficijentom korelacije, kojeg smo se prisjetili prošli tjedan, no znamo da Pearsonov koeficijent korelacije mjeri samo linearnu zavisnost između varijabli. Ako postoji nelinearna zavisnost između varijabli – što je u stvarnosti itekako moguće – to nećemo moći otkriti Pearsonovim koeficijentom korelacije. Treba nam, dakle, nešto općenitije, nešto što mjeri bilo kakvu zavisnost između varijabli. Rješenje trebamo potražiti u samoj definiciji stohastičke nezavisnosti. Pogledajmo marginalnu zavisnost (ista zapažanja vrijedit će i za uvjetnu nezavisnost). Kada su varijable (marginalno) nezavisne? Onda kada vrijedi:

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

U stvarnosti, rijetko kada ćemo imati savršenu nezavisnost. Čak i da varijable jesu savršeno nezavisne, sjetimo se da mi procjenjujemo parametre njihovih distribucija na temelju uzorka, i te procjene nikada nisu savršene. Dakle, ne možemo očekivati da će u praksi situacija biti tako čista: nećemo imati savršenu nezavisnost. Umjesto toga, zavisnost je pitanje stupnja, tj. varijable će biti u određenoj mjeri zavisne. To znači da, što su varijable više zavisne, to ćemo više odstupati od gornje jednakosti. A to onda znači da, kako bismo mjerili zavisnost varijable, trebamo mjeriti koliko $P(X, Y)$ odstupa od $P(X)P(Y)$. Ako je to odstupanje blizu nule, onda možemo reći da su varijable X i Y nezavisne. Ako je odstupanje vrlo malo, možemo reći da su varijable malo zavisne. Ako je odstupanje veliko, onda znamo da gornja jednakost sigurno ne vrijedi, tj. znamo da su varijable zavisne (bilo linearno ili nelinearno).

Sada se postavlja pitanje kako možemo mjeriti koliko $P(X, Y)$ odstupa od $P(X)P(Y)$? Znamo da su $P(X, Y)$ i $P(X)P(Y)$ dvije distribucije, pa se dakle pitanje svodi na to kako mjeriti odstupanje jedne distribucije od druge. U statistici se za to koristi **divergencija** – funkcija koja mjeri udaljenost između dviju distribucija. Jedna takva mjera divergencije, često korištena u strojnom učenju, jest **Kullback-Leiblerova divergencija** – divergencija između distribucije $P(x)$ u odnosu na $Q(x)$:

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

Ova se mjera može lako izvesti iz **relativne entropije**. Što je vrijednost ove mjere veća, to su distribucije međusobno različitije. Nas ovdje konkretno zanima KL-divergencija između $P(X, Y)$ i $P(X)P(Y)$, koja je jednaka: 9

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

Ovako definirana KL-divergencija naziva se **uzajamna informacija** (engl. *mutual information*) i označava sa $I(x, y)$. Dakle, uzajamna informacije $I(x, y)$ je zapravo KL-divergencija između zajedničke distribucije $P(X, Y)$ i zajedničke distibucije uz pretpostavku nezavisnosti (koja je onda jednaka umnošku marginalnih distribucija, $P(X)P(Y)$). Što je $I(x, y)$ veća, to su distribucije $P(X, Y)$ i $P(X)P(Y)$ različitije, tj. to su varijable X i Y **više zavisne**, jer smo sve udaljeniji od jednakosti distribucija $P(X, Y)$ i $P(X)P(Y)$. S druge strane, ako $X \perp Y$, onda je $I(x, y) = 0$ (vidimo da ćemo tada imati $\log 1 = 0$).

Dva popularna algoritma za polunaivni Bayesov klasifikator koji koriste mjeru uzajamne informacije su **TAN** (engl. *tree augmented naive Bayes*) i k-**DB** (engl. *k-limited dependence Bayesian classifier*). Ti algoritmi koriste mjeru uzajamne informacije kako bi heuristički pretražili prostor stanja mogućih particija varijabli, s ciljem da se združe one varijable koje su najviše međusobno zavisne, čime se dobiva polunaivni Bayesov klasifikator. Ovdje nećemo ići u detalje tih algoritama. Dovoljno je da znamo da, ako trebamo nekako odrediti koje su varijable najviše međusobno zavisne, kako bismo ih združiti u zajednički faktor, za to možemo koristiti mjeru uzajamne informacije. 10 11

Sažetak

- Gaussov Bayesov klasifikator sa dijeljenom kovarijacijskom matricom daje istu granicu kao i **logistička regresija**, no ima više parametara
- Naivan Bayesov klasifikator faktorizira izglednost na temelju pretpostavke o **uvjetnoj nezavisnosti** značajki unutar klase, čime se smanjuje broj parametara i omogućava generalizacija
- Parametre naivnog Bayesovog klasifikatora možemo procijeniti pomoću MLE ili MAP
- **Polunaivni Bayesov klasifikator** modelira zavisnost između odabranih varijabli, čime dobivamo **složeniji model**

Bilješke

1 Prva tri poglavlja današnjeg predavanja slijede poglavlja 3.1–2 i 5.4–6 iz (Alpaydin, 2020).

2 Ovo je zapravo **logaritam omjera šansi** (engl. *log odds*):

$$\ln \frac{p}{1-p}$$

gdje je p neka vjerojatnost. Omjer šansi (engl. *odds*) alternativni je način da se iskaže izglednost nekog događaja, definiran jednostavno kao omjer vjerojatnosti da se događaj dogodi i vjerojatnosti da se događaj ne dogodi (ili, ako želimo izbjeći vjerojatnost, a shvatimo je frekventistički, kao broj pozitivnih realizacija u ponavljanju pokusa, onda je to omjer broja pozitivnih ishoda i broja negativnih ishoda. Logaritam omjera šansi je logaritam tog omjera, i to je zapravo inverzna funkcija sigmoidne funkcije. Drugačije rečeno, ako $p = \sigma(\alpha)$, onda je $\alpha = \ln \frac{p}{1-p}$. Logaritam omjera šansi naziva se i **logit** funkcija. Kolokvijalno, "logit" je broj koji dovodimo na ulaz sigmoidalne (također i softmax) funkcije. Logiti su važni u statističkoj analizi efekata pomoću logističke regresije.

3 Prisjetite se da smo prošli puta spomenuli da je Gaussov Bayesov klasifikator s linearnom granicom između klasa istovjetan modelu **linearne diskriminantne analize (LDA**).

4 Preciznije, ako se primjer $\mathbf{x}$ nije pojavio u skupu za učenje niti sa jednom oznakom y, onda će aposteriorna vjerojatnost $P(y|\mathbf{x})$ biti 0/0, tj. nedefinirana. Ako se, međutim, primjer pojavio u skupu za učenje, ali se nije pojavio u nekoj od klasa, tj. $P(\mathbf{x}) \neq 0$, ali za neku oznaku y vrijedi $P(\mathbf{x}|y) \neq 0$, onda će aposteriorna vjerojatnost $P(y|\mathbf{x})$ za klase u kojima se primjer pojavio biti različita od nule, a za klase u kojima se primjer nije pojavit bit će jednaka nuli. No, u oba slučaja model loše generalizira.

5 Važan tehnički detalj koji smo ovdje zanemarili jest **podljev** (engl. *underflow*) koji se može dogoditi pri izračunu zajedničke vjerojatnosti u brojniku ili nazivniku Bayesovog pravila. Naime, množenjem pojedinačnih faktora, koji će u pravilu biti vrlo mali brojevi, dobivamo ekstremno male brojeve koji su manji od onoga što je prikazivo u zapisu s pomičnim zarezom. Rješenje za to je tzv. **log-sum-exp trik**; v. https://stats.stackexchange.com/q/105602/93766.

6 Legitimno pitanje je zašto ne bismo procjenitelj MAP koristili za procjenu parametara apriorne vjerojatnosti $P(y)$, a ne samo izglednosti $P(x_k|y)$? Odgovor je da za to nema potrebe. Kod faktora $P(x_k|y)$ lako se može dogoditi da se neka kombinacija (x_k, y) nikada nije pojavila u skupu za učenje, stoga, kako bismo izbjegli da vjerojatnost te kombinacije bude jednaka nuli (i time efektivno bude proglašena nemogućom), radimo zaglađivanje, tj. koristimo procjenitelj MAP. Kod vjerojatnosti $P(y)$, međutim, vjerojatnost će za neku vrijednost oznake y biti jednaka nuli samo onda kada u skupu za učenje ne postoji niti jedan primjer koji pripada dotičnoj klasi. A ako je to slučaj, onda nema smisla pokušavati izgraditi klasifikator za tu klasu. Zato nema potrebe zaglađivati procjenu za $P(y)$.

7 Jamačno ste se upitali kako bismo u tekstu automatski odredili je li riječ "gol" imenica (u značenju vratiju sastavljenih od dviju stativa, prečke i mreže, ili u značenju pogotka postignutog ubacivanjem

lopte) ili pridjev (u značenju onoga koji na sebi nema odjeće ili nečega što je bez svog bitnog svojstva). Taj zadatak u domeni je područja **obrade prirodnog jezika** (engl. *natural language processing, NLP*), i poznat je pod nazivom **označavanje vrste riječi** (engl. *part-of-speech tagging, PoS tagging*), i smatra se za većinu jezika zadovoljavajuće riješenim problemom. Algoritmi za označavanje vrste riječi temelje se na, dakako, na strojnom učenju, i to su uglavnom modeli već spomenutog **uvjetnog slučajnog polja** (engl. *conditional random field*), ili, u novije vrijeme, povratne neuronske mreže. Više: https://en.wikipedia.org/wiki/Part-of-speech_tagging i http://nlpprogress.com/english/part-of-speech_tagging.html.

8 **Bellov broj** može se izračunati različitim metodama, v. http://fredrikj.net/blog/2015/08/computing-bell-numbers/. Vjerojatno najjednostavnija metoda je **Bellov trokut**, v. https://code.sololearn.com/cxt4qe0Dy7hp/#py. Inače, Bellovi brojevi nazivaju se tako po matematičaru Ericu Bellu, koji se bavio teorijom brojeva, pa je tako pisao i o Bellovim brojevima, ali su ti brojevi matematičarima bili poznati i ranije. Korisno je upamtiti da $B_n \leqslant n!$, tj. broj particija nekog skupa nije veći od broja permutacija elemenata tog skupa (jer redoslijed elemenata u skupu ne igra ulogu).

9 Pokažimo kako je **Kullback-Leiblerova divergencija** izvedena iz relativne entroprije. Prisjetimo se, **entropija** je definirana kao

$$H(P) = -\sum_x P(x) \ln P(x)$$

i to je prosječan minimalan broj bitova potreban za kodiranje događaja iz distribucije P. **Unakrsna entropija** definirana je kao

$$H(P, Q) = -\sum_x P(x) \ln Q(x)$$

i to je prosječan broj bitova potreban za kodiranje događa, ako se upotrijebi shema kodiranja koja je optimalna za drugu distribuciju, Q. **Relativa entropija** $P(x)$ u odnosu na $Q(x)$ definirana je kao razlika unakrsne entropije i entropije:

$$\begin{aligned} H(P, Q) - H(P) &= -\sum_x P(x) \ln Q(x) - \Big(-\sum_x P(x) \ln P(x) \Big) \\ &= -\sum_x P(x) \ln Q(x) + \sum_x P(x) \ln P(x) \\ &= \sum_x P(x) \ln \frac{P(x)}{Q(x)} = D_{\mathrm{KL}}(P||Q) \end{aligned}$$

i to je Kullback-Leiblerova divergencija.

10 Uzajamnu informaciju lako možemo proširiti na **uvjetnu uzajamnu informaciju** (engl. *conditional mutual information*):

$$I(x, y|z) = \sum_z P(z_k) I(x, y|z) = \sum_z \sum_x \sum_y P(x, y, z) \ln \frac{P(x, y|z)}{P(x|z)P(y|z)}$$

11 Detalje o algoritmima **TAN** i k-**DB**, popraćene primjerima, možete naći u poglavlju 4.3 u skripti.

Literatura

E. Alpaydin. *Introduction to machine learning.* MIT press, 2020.