Strojno učenje

Jan Šnajder, Bojana Dalbelo Bašić

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Ovo je radna verzija skripte: **verzija 3.6 (2013-02-04)**.

Skripta se koristi za predmet *Strojno učenje* na Sveučilištu u Zagrebu, Fakultetu elektrotehnike i računarstva.
http://www.fer.unizg.hr/predmet/su

Ispravke, primjedbe i komentare šaljite na jan.snajder@fer.hr.

Sadržaj

Poglavlje 1

Nadzirano učenje

Postupcima nadziranog učenja mogu se rješavati dvije vrste problema: **klasifikacija** i **regresija**. Kod klasifikacije primjeru pridružujemo **klasu** (razred) kojoj taj primjer pripada. Kod regresije primjeru pridružujemo neku kontinuiranu vrijednost. Razlika je dakle u tome je li ciljna varijabla diskretna ili **nominalna** (klasifikacija) ili je kontinuirana (regresija). Razmotrimo najprije klasifikaciju.

1.1 Osnovni pojmovi

1.1.1 Primjeri za učenje

Svrha klasifikacije jest odrediti klasu $\mathcal{C}$ kojoj pripada primjer $\mathbf{x}$. Primjer ćemo definirati kao vektor značajki, $\mathbf{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$, gdje je n dimenzija vektora. Primjeri se mogu interpretirati kao točke u n-dimenzijskom vektorskom prostoru koji nazivamo **ulazni prostor** (engl. *input space*) ili **prostor primjera** (engl. *instance space*). Neka je $\mathcal{X}$ skup svih mogućih primjera. Pretpostavka svih algoritama strojnog učenja jest da su primjeri iz $\mathcal{X}$ uzorkovani nezavisno i iz iste zajedničke distribucije $P(\mathbf{x}, y)$. Ta se pretpostavka skraćeno označava s **iid** (engl. *independent and identically distributed*).

Kod nadziranog učenja unaprijed nam je poznata **oznaka klase** y kojoj pripada primjer $\mathbf{x}$ iz skupa za učenje. Ako se ograničimo na samo dvije klase, onda $y \in \{0, 1\}$, gdje $y = 1$ označava da je primjer za jednu (od ukupno dvije) klase pozitivan (pripada klasi), a $y = 0$ označava da je primjer za tu klasu negativan (ne pripada toj klasi). Klasifikator koji klasificira u dvije klase nazivamo **binarni klasifikator**. Učenje binarnog klasifikatora istovjetno je učenju Booleove funkcije, koje se u literaturi naziva i **učenje koncepta** (engl. *concept learning*).

Skup primjera za učenje $\mathcal{D}$ sastoji se od parova primjera i pripadnih oznaka, $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, gdje je N ukupan broj primjera za učenje, a i je indeks primjera odnosno njemu pripadne oznake. Skup $\mathcal{D}$ možemo prikazati tablično:

x_1	x_2	$\ldots$	x_n	y
$x_1^{(1)}$	$x_2^{(1)}$	$\ldots$	$x_n^{(1)}$	$y^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	$\ldots$	$x_n^{(2)}$	$y^{(2)}$
$\vdots$	$\vdots$		$\vdots$	$\vdots$
$x_1^{(N)}$	$x_2^{(N)}$	$\ldots$	$x_n^{(N)}$	$y^{(N)}$

1.1.2 Hipoteza

Zadaća klasifikacijskog algoritma jest inducirati (naučiti) **hipotezu** $h : \mathcal{X} \to \{0, 1\}$ koja određuje pripada li neki primjer $\mathbf{x}$ klasi $\mathcal{C}$ ili ne:

$$h(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \text{ pripada klasi } \mathcal{C} \\ 0 & \mathbf{x} \text{ ne pripada klasi } \mathcal{C} \end{cases}$$

Definicija 1 *Kažemo da primjer* $\mathbf{x} \in \mathcal{X}$ **zadovoljava** *hipotezu* $h \in \mathcal{H}$ *akko* $h(\mathbf{x}) = 1$.

Definicija 2 *Kažemo da je hipoteza* h **konzistentna** *s primjerom za učenje* $(\mathbf{x}, y)$ *akko* $h(\mathbf{x}) = y$. *Konzistentnost hipoteze sa svim primjerima za učenje iz* $\mathcal{D}$ *definiramo kao*

$$Consistent(h, \mathcal{D}) \iff \forall (\mathbf{x}, y) \in \mathcal{D}.\ \big(h(\mathbf{x}) = y\big)\,.$$

Često primjere treba klasificirati u više od jedne klase, što nazivamo **višeklasnom klasifikacijom** (engl. *multiclass classification*). Npr. klasifikacija novinskih članaka u rubrike ili klasifikacija rukom pisanih znamenki. U općenitom slučaju postoji K klasa, $\mathcal{C}_j$, gdje $j = 1, \ldots, K$. Tada je najprikladnije oznaku klase primjera $\mathbf{x}^{(i)}$ prikazati kao K-dimenzijski vektor, $\mathbf{y}^{(i)} = (y_1^{(i)}, y_2^{(i)}, \ldots, y_K^{(i)})^{\mathrm{T}}$ gdje

$$y_j^{(i)} = \begin{cases} 1 & \text{ako } \mathbf{x}^{(i)} \in \mathcal{C}_j \\ 0 & \text{inače} \end{cases}$$

Npr. $\mathbf{y}^{(2)} = (0, 0, 1, 0)^{\mathrm{T}}$ značilo bi da primjer $\mathbf{x}^{(2)}$ pripada klasi $\mathcal{C}_3$.

U još općenitijem slučaju jedan primjer može istovremeno pripadati u više klasa, što nazivamo **klasifikacija s višestrukim oznakama** (engl. *multilabel classification*) ili **klasifikacija jedan-na-više**; klasifikaciju tog tipa nećemo posebno razmatrati budući da se da izvesti kao klasifikacija tipa jedan-na-jedan.

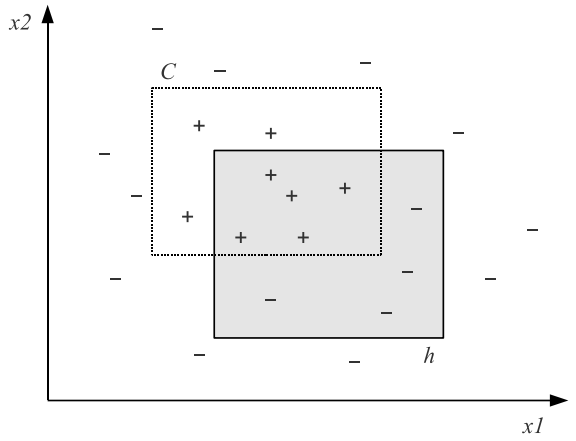
1.1.3 Model

Hipoteze ne izmišljamo ni iz čega, nego ih odabiremo iz pomno odabranog skupa mogućih hipoteza. Skup mogućih hipoteza $\mathcal{H}$ nazivamo **model** ili **prostor hipoteza** (ova dva pojma koristit ćemo ravnopravno). Npr. ako je prostor primjera dvodimenzionalan, $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$, klasa hipoteze može biti pravac koji razdjeljuje primjere dviju klasa ili to može biti pravokutnik koji obuhvaća sve primjere neke klase. Model $\mathcal{H}$ dakle određuje način prikaza hipoteze.

Učenje se zapravo svodi na pretraživanje prostora hipoteza $\mathcal{H}$ i nalaženje najbolje hipoteze $h \in H$. Najbolja hipoteza je ona koja najtočnije klasificira primjere. (Naravno, slijepo pretraživanje prostora $\mathcal{H}$ ne dolazi u obzir budući da je $\mathcal{H}$ obično vrlo velik.)

Koliko dobro hipoteza h klasificira primjere za učenje iskazuje **empirijska pogreška** ili **pogreška učenja** (engl. *training error*). Empirijska pogreška hipoteze h, mjerena na skupu $\mathcal{D}$, jednaka je udjelu primjera iz $\mathcal{D}$ koji nisu ispravno klasificirani:

$$E(h|\mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{h(\mathbf{x}^{(i)}) \neq y^{(i)}\} = \frac{1}{N} \sum_{i=1}^{N} |h(\mathbf{x}^{(i)}) - y^{(i)}|$$



Slika 1.1: Prostor primjera $\mathcal{X} = \mathbb{R}^2$, područje koje zadovoljava hipotezu h te područje koje odgovara klasi $\mathcal{C}$.

gdje je $\mathbf{1}\{P\}$ indikatorska funkcija čija je vrijednost 1 ako $P \equiv \top$, a 0 inače. Očito, hipoteza je konzistentna s primjerima za učenje akko $E(h|\mathcal{D}) = 0$. Kao primjer razmotrimo hipotezu sa slike 1.1: od ukupno 23 primjera, hipoteza pogrešno klasificira njih sedam, pa $E(h|D) = 7/23$. Primjeri koje hipoteza klasificira pozitivno, a zapravo su negativni, zovemo **lažno pozitivni** primjeri (engl. *false positives*, FP). Obrnuto, primjeri koje hipoteza klasifcira negativno, a zapravo su pozitivni, zovemo **lažno negativni** primjeri (engl. *false negatives*, FN).

Idealno, prostor hipoteza $\mathcal{H}$ uključuje klasu $\mathcal{C}$, tj. postoji $h \in \mathcal{H}$ takva da je h konzistentna s primjerima za učenje. No moguće je da takva hipoteza ne postoji, tj. da za sve $h \in \mathcal{H}$ vrijedi $E(h|\mathcal{D}) > 0$. Tada kažemo da model $\mathcal{H}$ nije dovoljnog **kapaciteta** (ili **složenosti**) da bi naučio klasu $\mathcal{C}$. Razmotrimo kao primjer prostor primjera sa slike 1.2. Ako je model $\mathcal{H}$ skup pravokutnika poravnatih s osima, onda niti jedna hipoteza iz $\mathcal{H}$ (npr. h_1) ne može naučiti klasu $\mathcal{C}$. U tom slučaju treba nam složeniji model (model u kojemu bismo mogli prikazati npr. hipotezu h_2).

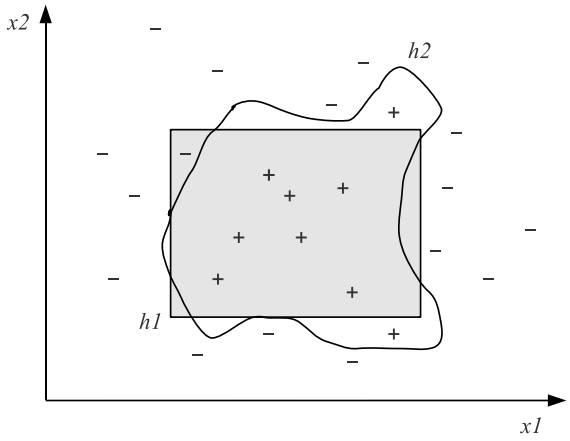
1.1.4 *Prostor inačica

Moguće je (i zapravo je vrlo često) da za neki skup za učenje $\mathcal{D}$ postoji više (moguće beskonačno mnogo) hipoteza modela $\mathcal{H}$ koje ispravno klasificiraju primjere iz $\mathcal{D}$. Skup takvih hipoteza nazivamo **prostor inačica** (engl. *version space*).

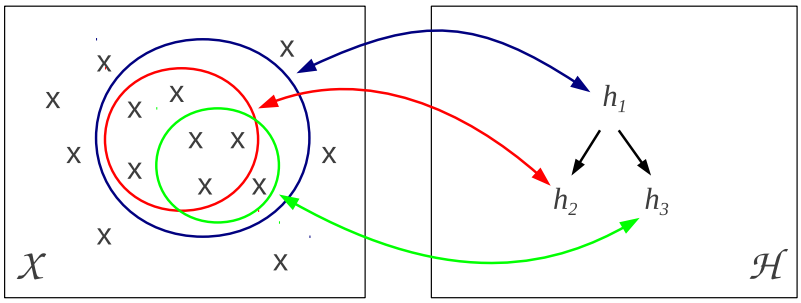
Definicija 3 (Prostor inačica) *Prostor inačica $VS_{\mathcal{H},\mathcal{D}} \subseteq \mathcal{H}$ modela $\mathcal{H}$ jest skup hipoteza koje su konzistentne s primjerima za učenje $\mathcal{D}$:*

$$VS_{\mathcal{H},\mathcal{D}} = \{h \in \mathcal{H} \mid \mathit{Consistent}(h, \mathcal{D})\}\,. \tag{1.1}$$

Prostor inačica može se zapisati kompaktnije ako se u obzir uzmu odnosi između hipoteza, kao što prikazuje slika 1.3. Za neki odabrani model $\mathcal{H}$, hipoteze $h \in \mathcal{H}$ razlikuju se po svojoj općenitosti, odnosno po tome koliko ih primjera može zadovoljiti.



Slika 1.2: Prostor primjera $X = \mathbb{R}^2$ i područje koje zadovoljava hipotezu h_1 (iz jednostavnijeg modela) odnosno hipotezu h_2 (iz složenijeg modela).



Slika 1.3: Prostor primjera $\mathcal{X} = \mathbb{R}^2$ i odgovarajući uređaj u prostoru hipoteza $\mathcal{H}$.

Definicija 4 (Relacija općenitija-ili-jednaka) *Kažemo da je hipoteza h_1 općenitija-ili-jednaka od hipoteze h_2, i pišemo $h_1 \geqslant_g h_2$, akko svi primjeri koji zadovoljavaju h_2 također zadovoljavaju i h_1:*

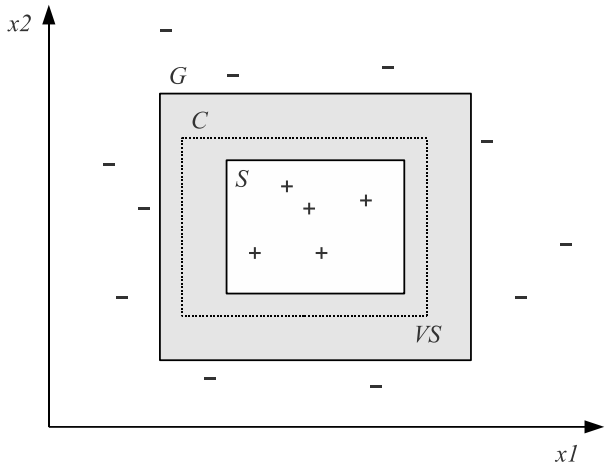
$$h_1 \geqslant_g h_2 \iff \forall \mathbf{x} \in \mathcal{X}.\big((h_2(\mathbf{x}) = 1) \Rightarrow (h_1(\mathbf{x}) = 1)\big)\,.$$

Relacija $\geqslant_g$ je relacija **parcijalnog uređaja** (refleksivna je, antisimetrična i tranzitivna), odnosno $(\mathcal{H}, \geqslant_g)$ je parcijalno uređen skup (*poset*). Skup je parcijalan jer nije svaki par hipoteza iz $\mathcal{H}$ međusobno usporediv (npr. hipoteze h_2 i h_3 sa slike 1.3 nisu usporedive).

Neka je S skup maksimalno specifičnih hipoteza konzistentnih s primjerima za učenje, a G skup maksimalno općenitih hipoteza konzistentnih s primjerima za učenje. Formalno:

$$\begin{aligned} S &= \big\{s \in \mathcal{H} \mid \mathit{Consistent}(s, \mathcal{D}) \ \wedge \ \forall h \in \mathcal{H}.\big((s \geqslant_g h) \wedge \mathit{Consistent}(h, \mathcal{D}) \Rightarrow (h \geqslant_g s)\big)\big\} \\ G &= \big\{g \in \mathcal{H} \mid \mathit{Consistent}(g, \mathcal{D}) \ \wedge \ \forall h \in \mathcal{H}.\big((h \geqslant_g g) \wedge \mathit{Consistent}(h, \mathcal{D}) \Rightarrow (g \geqslant_g h)\big)\big\} \end{aligned}$$

Prostor inačica sada se može sažeto prikazati kao skup svih hipoteza koje su specifičnije od neke hipoteze (ili jednake nekoj hipotezi) iz skupa G i općenitije od neke hipoteze (ili jednake nekoj hipotezi) iz S:



Slika 1.4: Prostor primjera $\mathcal{X} = \mathbb{R}^2$ i prostor inačica.

$$VS_{\mathcal{H},\mathcal{D}} = \{h \in \mathcal{H} \mid \exists g \in G, \exists s \in S. \, (g \geqslant_g h \geqslant_g s)\} \, . \tag{1.2}$$

Ova definicija ekvivalentna je definiciji (1.1). Na prvi pogled možda nije očito da su hipoteze h za koje vrijedi $g \geqslant_g h \geqslant_g s$ ujedno i konzistente s primjerima za učenje $\mathcal{D}$. Razmotrimo zašto je to ipak slučaj. Hipotezu s po definiciji skupa S zadovoljavaju svi pozitivni primjeri. Budući da $h \geqslant_g s$, svi pozitivni primjeri zadovoljavaju i hipotezu h. Slično, hipotezu g po definiciji skupa G ne zadovoljava niti jedan negativan primjer. Budući da $g \geqslant_g h$, to niti jedan negativni primjer ne zadovoljava hipotezu h. Budući da h zadovoljavaju svi pozitivni primjeri, a niti jedan negativan, to je hipoteza h konzistentna s primjerima za učenje.

Slika 1.4 prikazuje prostor primjera $\mathcal{X} = \mathbb{R}^2$ te pravokutnike koji odgovaraju najopćenitijoj i najspecifičnijoj hipotezi. Zasivljeno područje odgovara hipotezama iz prostora inačica.

Ako $G = S$, prostor inačica sadrži hipoteze koje su jednako općenite (a idealno samo jednu takvu hipotezu), pa možemo reći da je klasa $\mathcal{C}$ potpuno naučena. S druge strane, ako $VS = \varnothing$, onda ne postoji hipoteza $h \in \mathcal{H}$ koja bi bila konzistentna s primjerima $\mathcal{D}$, odnosno model $\mathcal{H}$ nije dovoljnog kapaciteta.

Premda je svaka hipoteza iz prostora inačica konzistentna s primjerima za učenje, ne znači da svaka hipoteza ispravno klasificira još neviđene primjere. Činjenica da prostor inačica postoji posljedica je toga što u skupu za učenje $\mathcal{D}$ nije postojao niti jedan primjer koji bi u ulaznom prostoru pao u područje koje odgovara prostoru inačica. Primjer iz tog područja je svaki onaj koji zadovoljava neku hipotezu iz G, ali ne zadovoljava njoj specifičniju hipotezu iz S, ili obrnuto. Formalno, to je primjer $\mathbf{x} \in \mathcal{X}$ za koji vrijedi

$$\exists g \in G, \exists s \in S. \, (g \geqslant_g s) \wedge \big(g(\mathbf{x}) \neq s(\mathbf{x})\big) \, .$$

Ako bismo u skup $\mathcal{D}$ pridodali primjer $\mathbf{x}$, mogli bismo iz prostora inačica ukloniti sve one hipoteze koje s tim primjerom nisu konzistentne, pa bi se time prostor inačica suzio. Možemo se zapitati: ako bi klasifikacijski algoritam mogao sam birati primjere za učenje (odnosno ako bi mogao generirati upite), koji bi primjer bilo najbolje odabrati u ovakvoj

situaciji? Budući da klasifikacija tog primjera još nije poznata, valja pretpostaviti da ona s jednakom vjerojatnošću može biti i pozitivna i negativna. Algoritam bi onda trebao odabrati primjer koji će zadovoljavati polovica hipoteza iz prostora inačica, tj. takav primjer $\mathbf{x} \in \mathcal{X}$ za koji vrijedi

$$\sum_{h \in VS_{\mathcal{H},\mathcal{D}}} h(\mathbf{x}) = \left\lceil \frac{1}{2} |VS_{\mathcal{H},\mathcal{D}}| \right\rceil .$$

Takvom strategijom algoritmu će trebati $\mathcal{O}(\log_2 |VS_{\mathcal{H},\mathcal{D}}|)$ primjera da potpuno nauči klasu $\mathcal{C}$. Na ovoj se ideji temelji tehnika **aktivnog učenja** (engl. *active learning*): algoritam postavlja upite samo za one primjere koji mu trebaju da bi suzio prostor inačica. To može znatno ubrzati i pojeftiniti izgradnju klasifikatora (osobito onda kada se primjeri označavaju ručno).

1.2 Vapnik-Chervonenkisova dimenzija

Modeli nisu jednakog kapaciteta: neki modeli su fleksibilniji i mogu se bolje prilagoditi podacima, a neki su manje fleksibilni. Razmatranja ovog tipa u domeni su **statističke teorije učenja** (engl. *statistical learning theory*) odnosno **računalne teorije učenja** (engl. *computational learning theory, COLT*). Jedan način iskazivanja kapaciteta modela jest Vapnik-Chervonenkisova dimenzija.

Vapnik-Chervonenkisova dimenzija (VC-dimenzija) iskazuje kapacitet modela $\mathcal{H}$ u smislu broja primjera za klasifikaciju s kojim se model $\mathcal{H}$ može uspješno nositi.

Pretpostavimo da skup podataka za učenje sadrži N primjera (odnosno točaka u ulaznom prostoru). Proizvoljno odaberimo jednu konfiguraciju tih točaka u prostoru. Ako se ograničimo na samo jednu klasu, svaku od tih N točaka moguće je označiti kao pozitivnu ili negativnu. Dakle postoji 2^N mogućih označavanja, odnosno 2^N mogućih problema. Ako za svako označavanje možemo pronaći hipotezu $h \in \mathcal{H}$ takvu da h razdvaja pozitivne primjere od negativnih (tj. da je h konzistentna s $\mathcal{D}$), kažemo da $\mathcal{H}$ **razdjeljuje** (engl. *shatters*) N točaka. Formalno:

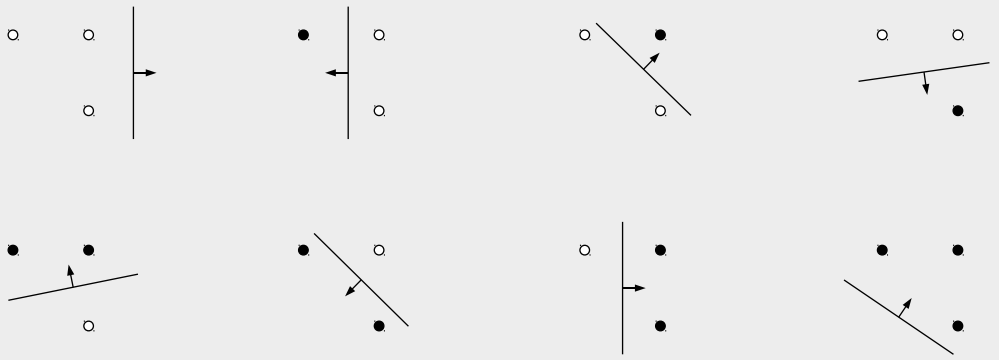
Definicija 5 (Razdjeljivanje primjera) *Neka je funkcija $y : \mathcal{X} \to \{0, 1\}$ funkcija koja primjerima iz $\mathcal{X}$ dodjeljuje oznake klase. Model $\mathcal{H}$ razdjeljuje N primjera akko*

$$\exists \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\} \subseteq \mathcal{X}, \forall y, \exists h \in \mathcal{H}, \forall i \in \{1, \ldots, N\}. \left(h(\mathbf{x}^{(i)}) = y(\mathbf{x}^{(i)})\right) .$$

Definicija 6 (VC-dimenzija) *VC-dimenzija modela $\mathcal{H}$, označena kao $VC(\mathcal{H})$, jest najveći broj primjera koje model $\mathcal{H}$ može razdijeliti.*

Primijetite da je odabir N primjera u ulaznom prostoru proizvoljan, no jednom kada je on fiksiran, razdvajanje mora biti moguće za svih 2^N označavanja. VC-dimenzija ne ovisi o konkretnom skupu za učenje; to je teoretska mjera složenosti klase hipoteze, a ne složenosti ulaznog skupa podataka. Također valja primijetiti da je moguće $VC(\mathcal{H}) = \infty$.

Primjer 1.1 (VC-dimenzija pravca) Neka je ulazni prostor $\mathcal{X} = \mathbb{R}^2$ te neka je model $\mathcal{H}$ skup pravaca. Za tri nekolinearne točke svako od 2^3 označavanja moguće je razdvojiti pravcem:



pa zaključujemo $VC(\mathcal{H}) \geqslant 3$. Međutim, za četiri točke, kako god odabrane, ne može se za svako označavanje pronaći razdvajajući pravac. Naime, u potpuno povezanom grafu s četiri vrha, dva brida se uvijek sijeku. Ako se u vrhove jednog takvog brida smjeste dva pozitivna primjera, a u vrhove drugog dva negativna primjera, onda nije moguće povući pravac koji bi razdvojio pozitivne primjere od negativnih. Stoga vrijedi $VC(\mathcal{H}) < 4$, pa zaključujemo $VC(\mathcal{H}) = 3$.

Može se pokazati da hiperavnina u prostoru $\mathbb{R}^n$ može razdijeliti najviše $n+1$ točaka, pa je dakle VC-dimenzija linearnog klasifikatora u n dimenzijskom prostoru jednaka $n+1$.

VC-dimenzija u stvari daje dosta pesimističnu ocjenu kapaciteta hipoteze. Ispada da linearni klasifikator (klasifikator za koji je $\mathcal{H}$ skup pravaca) može naučiti klasu s najviše tri primjera. Međutim, VC-dimenzija ne uzima u obzir distribuciju primjera: u praksi su slični primjeri bliži jedan drugome u ulaznome prostoru (to naravno ovisi o izboru značajki), pa sva označavanja nisu jednako vjerojatna.

VC-dimenzija je donekle povezana s **brojem parametara** modela: modeli s većom VC-dimenzijom tipično imaju više parametara koje treba optimirati, dok modeli s manjom VC-dimenzijom takvih parametara obično imaju manje. No to nije pravilo; moguće je napraviti model u kojem je više parametara stopljeno u jedan parametar, kao i model koji ima suvišne, međusobno funkcijski zavisne parametre (v. primjer 1.5).

Zadatak 2.1. Neka je $\mathcal{X} = \mathbb{R}^2$. Koliko iznosi $VC(\mathcal{H})$, ako je model $\mathcal{H}$:

- skup pravokutnika čije su stranice poravnate s osima?
- skup kružnica sa središtem u ishodištu?
- skup elipsa sa središtem u ishodištu?
- skup konveksnih poligona?
- look-up tablica koja pohranjuje primjere?

Napomena: Dokaz za $VC(\mathcal{H}) = N$ potrebno je napraviti u dva koraka. Najprije treba pokazati da $\mathcal{H}$ razdjeljuje N primjera (tj. da za proizvoljno odabranih N točaka za svako moguće označavanje postoji konzistentna hipoteza), a zatim da $\mathcal{H}$ ne razdjeljuje $N+1$ primjera (tj. da za svakih $N+1$ točaka postoji neko označavanje za koje ne postoji konzistentna hipoteza).

1.3 Induktivna pristranost

Učenje hipoteze interesantan je problem utoliko što je riječ o **loše definiranom problemu** (engl. *ill-posed problem*): primjeri za učenje $\mathcal{D}$ nisu sami po sebi dovoljni da bi se na temelju njih jednoznačno inducirala hipoteza h. Drugim riječima, iz $\mathcal{D}$ ne slijedi (deduktivno) koja će hipoteza dobro klasificirati nove primjere koji se nisu našli u $\mathcal{D}$. Svojstvo hipoteze da odredi (predvidi) klasifikaciju još neviđenih primjera naziva se **generalizacija**.

Primjer 1.2 (Učenje Booleove funkcije) Razmotrimo kao primjer učenje Booleove funkcije od n varijabli. Ovdje je riječ o klasifikaciji primjera $\mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$ u dvije klase, $y \in \{0, 1\}$. (Digresija: Je li VC-dimenzija linearnog modela dovoljna da se nauči Booleova funkcija od $n = 2$ varijable?) Postoji 2^{2^n} različitih Booleovih funkcija od N varijabli. Ako $|\mathcal{D}| = N$, još uvijek postoji $2^{2^n - N}$ hipoteza koje su konzistentne s primjerima za učenje. Npr., za $n = 3$ i $N = 5$:

x_1	x_2	x_3	y
0	0	0	?
0	0	1	?
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	?
1	1	1	1

postoji 8 konzistentih hipoteza. Postavlja se pitanje: koja je od njih ispravna? Problem nije dobro definiran, pa na to pitanje nije moguće odgovoriti. Činjenica je da su sve hipoteze jednako vjerodostojne.

Očito, učenje, a tako i generalizacija, nisu mogući bez dodatnih pretpostavki. Skup naših (apriornih) pretpostavki koje omogućavaju induktivno učenje nazivamo **induktivna pristranost** (engl. *inductive bias*). Činjenicu da je učenje bez pristranosti uzaludno izražava i jedan od teorema nazvan *No Free Lunch Theorem*.[1]

Induktivnu pristranost odabiremo na način da: (1) definiramo model $\mathcal{H}$, čime je određeno koje hipoteze uopće dolaze u obzir te (2) definiramo način na koji se iz prostora $\mathcal{H}$ odabire točno jedna hipoteza h. U tom smislu razlikujemo dvije vrste induktivne pristranosti:

(1) **Pristranost ograničavanjem** ili **pristranost jezika** (engl. *restriction bias, language bias*) – odabiremo model $\mathcal{H}$ i time ograničavamo skup hipoteza koji se mogu prikazati tim modelom;

(2) **Pristranost preferencijom** ili **pristranost pretraživanja** (engl. *preference bias, search bias*) – definiramo način pretraživanja hipoteza unutar $\mathcal{H}$ i na taj način zapravo dajemo prednost jednim hipotezama u odnosu na druge.

[1] http://www.no-free-lunch.org

Većina algoritama učenja kombinira obje vrste induktivne pristranosti. Npr. induktivna pristranost može biti: (1) $\mathcal{H}$ je skup pravokutnika te (2) odabiremo pravokutnik najmanje površine konzistentan s primjerima za učenje (tj. najspecifičniju hipotezu).

Premda izgleda da je pristranost preferencijom – budući da ne ograničava skup pretraživih hipoteza – bolja vrsta pristranosti, u praksi se ipak odlučujemo za neki ograničen model $\mathcal{H}$. Jednostavniji modeli imaju niz pogodnosti (v. odjeljak 1.6).

Induktivnu pristranost moguće je definirati i nešto formalnije. Ako znamo da hipotezu nije moguće odrediti bez induktivne pristranosti, onda induktivnu pristranost možemo shvatiti kao dodatnu informaciju koja nam omogućava da na temelju nepotpune informacije iz primjera za učenje ipak možemo zaključiti o kojoj je točno hipotezi riječ. Rečeno drugačije: induktivna pristranost je skup pretpostavki temeljem kojih klasifikacija (novog) primjera slijedi **deduktivno**. Formalno:

Definicija 7 (Induktivna pristranost) *Neka je $\mathcal{L}$ algoritam za učenje, neka je $h_{\mathcal{L}}$ hipoteza inducirana pomoću $\mathcal{L}$ na skupu primjera $\mathcal{D}$ i neka je $h_{\mathcal{L}}(\mathbf{x})$ klasifikacija primjera $\mathbf{x} \in \mathcal{X}$ temeljem te hipoteze. Induktivna pristranost od $\mathcal{L}$ je bilo koji skup minimalnih pretpostavki $\mathcal{B}$ takvih da*

$$\forall \mathcal{D}. \forall \mathbf{x} \in \mathcal{X}. \left((\mathcal{B} \wedge \mathcal{D} \wedge \mathbf{x}) \ \vdash \ h_{\mathcal{L}}(\mathbf{x}) \right).$$

Induktivnu pristranost tako možemo shvatiti kao vezu između induktivnog zaključivanja i deduktivnog zaključivanja – ono što je nedostajalo indukciji da bi bila dedukcija.

Primjer 1.3 Razmotrimo ponovno učenje Booleove funkcije, ali najprije funkcije od dvije varijable, $n = 2$. Ukupno mogućih funkcija je $2^{2^n} = 16$. Ako kao model $\mathcal{H}$ odaberemo pravac, uveli smo induktivnu pristranost ograničenjem. Naime, za $n = 2$ broj primjera je $2^n = 4$, no kako je $VC(\mathcal{H}) = 3$, to kapacitet modela $\mathcal{H}$ sigurno nije dovoljan da bi se naučila baš svaka Booleova funkcija dvije varijable. Odabirom pravca kao modela $\mathcal{H}$, ograničili smo se na 14 od 16 hipoteza odnosno Booleovih funkcija (koje hipoteze nisu u $\mathcal{H}$)?

Primjer 1.4 Vratimo se primjeru 1.2 i učenju Booleove funkcije od $n = 3$ varijable. Prostor primjera je $\mathcal{X} = \{0, 1\}^3$. Uvedimo induktivnu pristranost ograničenjem: neka je model $\mathcal{H}$ ravnina u $\mathbb{R}^3$. Je li ta induktivna pristranost dovoljna da bi se naučila zadana funkcija? Koja je veličina prostora inačica te koja je klasifikacija preostala tri primjera? Što ako se iz skupa $\mathcal{D}$ ukloni primjer $(1, 0, 1)^{\mathrm{T}}$? Kolika je tada veličina prostora inačica i kako izgleda poset $(VS_{\mathcal{H},\mathcal{D}}, \geqslant_g)$? Koja je najspecifičnija, a koja najopćenitija hipoteza? Koju bismo dodatnu pristranost mogli uvesti (pristranost ograničenjem) tako da klasifikacija novog primjera ipak slijedi deduktivno (tj. da $|VS_{\mathcal{H},\mathcal{D}}| = 1$)?

1.4 Problem šuma

Šum je neželjena anomalija u podacima. Mogući uzroci šuma su:

- nepreciznost pri mjerenju značajki,

- pogreške u označavanju (engl. *teacher noise*),
- postojanje skrivenih značajki (latentnih varijabli),
- nejasne granice klasa (subjektivnost).

U prisustvu šuma ne postoji jednostavna granica između pozitivnih i negativnih primjera, čak i onda kada je problem zapravo inherentno jednostavan. Zbog toga jednostavni modeli (npr. modeli s niskom VC-dimenzijom) ne mogu ostvariti $E(h|\mathcal{D}) = 0$. Problem je to što u načelu šum nije moguće razdvojiti od pravih podataka. Iznimka su pojedinačni primjeri koji po nekoj vrijednosti znatno odskaču od većine drugih primjera (engl. *outliers*).

1.5 Regresija

Kod regresije ciljna vrijednost y je kontinuirana, $y \in \mathbb{R}$. Na temelju primjera $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}$ potrebno je naučiti nepoznatu funkciju $f : \mathcal{X} \to \mathbb{R}$ tako da, idealno, $y^{(i)} = f(\mathbf{x}^{(i)})$. Učenje funkcije možemo tumačiti kao interpolaciju između točaka $\mathbf{x}^{(i)}$, odnosno ekstrapolaciju izvan točaka $\mathbf{x}^{(i)}$. Međutim, zbog prisustva šuma, zapravo učimo funkciju $y^{(i)} = f(\mathbf{x}^{(i)}) + \varepsilon$, gdje je ε slučajni šum.

Regresijom nad skupom $\mathcal{D}$ dobivamo funkciju (hipotezu) h kao aproksimaciju funkcije f. Empirijska pogreška hipoteze h na skupu za učenje $\mathcal{D}$ najjednostavnije se može definirati kao

$$E(h|\mathcal{D}) = \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - h(\mathbf{x}^{(i)})\right)^2.$$

Pogrešku mjerimo kao zbroj kvadratnih odstupanja predviđene vrijednosti $h(\mathbf{x})$ i stvarne vrijednosti y. Pogrešku bismo mogli iskazati i na neki drugi način (razlog zašto smo odabrali baš ovaj oblik bit će jasan kasnije). Faktor $\frac{1}{2}$ uvršten je da bi pojednostavio račun.

Sada se najprije treba odlučiti za model $\mathcal{H}$ koji će biti dovoljnog kapaciteta da minimizira empirijsku pogrešku. Izaberimo linearan model:

$$h(\mathbf{x}) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + w_0 = \sum_{i=1}^{n} w_ix_i + w_0 = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

gdje su w_i parametri koje treba naučiti na temelju skupa primjera $\mathcal{D}$. Budući da vrijednost $h(\mathbf{x})$ linearno ovisi o ulaznim vrijednostima $\mathbf{x}$, ovu vrstu regresije nazivamo **linearna regresija**. Pretpostavimo, zbog jednostavnosti, da je prostor primjera $\mathcal{X}$ jednodimenzijski, $\mathcal{X} = \mathbb{R}$. Tada je hipoteza h definirana jednadžbom pravca

$$h(x) = w_1x + w_0$$

dok je empirijska pogreška

$$E(h|\mathcal{D}) = \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - (w_1x^{(i)} + w_0)\right)^2.$$

Naš je cilj pronaći hipotezu h koja minimizira empirijsku pogrešku. Budući da je empirijska pogreška dana kao zbroj kvadrata pogrešaka koje nastaju na pojedinačnim primjerima, riječ je o **postupku najmanjih kvadrata** (engl. *least squares*). Oblik hipoteze fiksiran je modelom, pa pronaći optimalnu hipotezu zapravo znači pronaći parametre w_0 i w_1 takve da $\nabla_{w_0,w_1} E(h|\mathcal{D}) = 0$.

Pronađimo najprije minimum s obzirom na parametar w_0:

$$\frac{\partial}{\partial w_0}\Big[\frac{1}{2}\sum_i^N \big(y^{(i)} - (w_1 x^{(i)} + w_0)\big)^2\Big] = \tag{1.3}$$
$$\sum_i^N (-y^{(i)} + w_1 x^{(i)} + w_0) = -\sum_i^N y^{(i)} + w_1 \sum_i^N x^{(i)} + N w_0 = 0\,.$$

Rješavanjem za w_0 i uvrštenjem $\bar{x} = \sum_i^N x^{(i)}/N$ i $\bar{y} = \sum_i^N y^{(i)}/N$ dobivamo

$$w_0 = \frac{1}{N}\big(\sum_i^N y^{(i)} - w_1 \sum_i^N x^{(i)}\big) = \frac{1}{N}\sum_i^N y^{(i)} - w_1 \frac{1}{N}\sum_i^N x^{(i)} = \bar{y} - w_1 \bar{x}\,. \tag{1.4}$$

Za w_1 dobivamo

$$\frac{\partial}{\partial w_1}\Big[\frac{1}{2}\sum_i^N \big(y^{(i)} - (w_1 x^{(i)} + w_0)\big)^2\Big] = \tag{1.5}$$
$$\sum_i^N -x^{(i)}(y^{(i)} - w_1 x^{(i)} - w_0) = -\sum_i^N x^{(i)} y^{(i)} + w_1 \sum_i^N (x^{(i)})^2 + w_0 \sum_i^N x^{(i)} = 0\,.$$

Uvrštenjem (1.4) u (1.5) te zamjenom $\sum_i^N x^{(i)} = N\bar{x}$ dobivamo

$$-\sum_i^N x^{(i)} y^{(i)} + w_1 \sum_i^N (x^{(i)})^2 + (\bar{y} - w_1\bar{x}) N\bar{x} = w_1\big(\sum_i^N (x^{(i)})^2 - N\bar{x}^2\big) + N\bar{x}\bar{y} - \sum_i^N x^{(i)} y^{(i)} = 0\,.$$

iz čega slijedi

$$w_1 = \frac{\sum_i^N x^{(i)} y^{(i)} - N\bar{x}\bar{y}}{\sum_i^N (x^{(i)})^2 - N\bar{x}^2}\,.$$

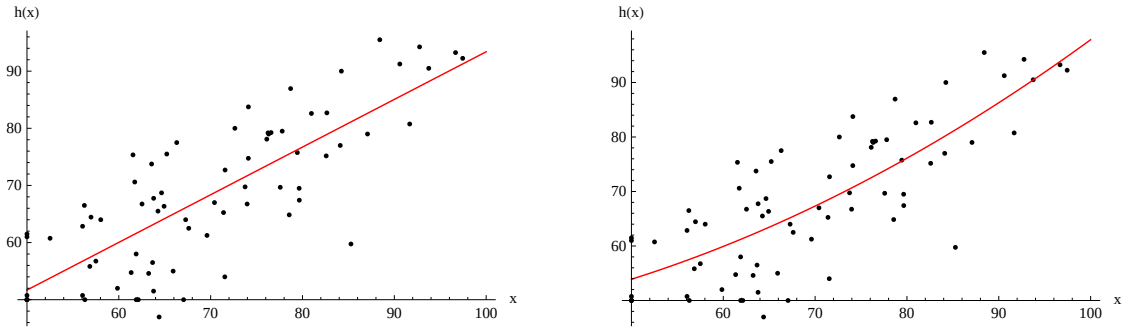
U ovom slučaju dakle postoji analitičko rješenje kojim se nalazi optimalna hipoteza. Također kažemo da rješenje postoji u **zatvorenoj formi** (engl. *closed-form solution*). U slučajevima kada analitičko rješenje ne postoji, učenje ćemo morati provesti **iterativnim optimizacijskim metodama**.

Ako je linearan model prejednostavan, empirijska pogreška bit će i nakon optimizacije prevelika. U tom slučaju možemo odabrati složeniji model, npr. polinom drugoga reda

$$h(x) = w_2 x^2 + w_1 x + w_0\,.$$

Ovo je primjer **polinomijalne regresije**, za koju također postoji rješenje u zatvorenoj formi. (Napomena: Polinomijalna regresija specifičan je slučaj tzv. poopćenog linearnog modela. Taj model zovemo linearnim jer je linearan u parametrima $\mathbf{w}$, premda nije linearan u ulazima $\mathbf{x}$).

Na slici 1.5 prikazan je primjer linearne i polinomijalne regresije za $\mathcal{X} = \mathbb{R}$.



(a) (b)

Slika 1.5: Regresija u $\mathcal{X} = \mathbb{R}$: (a) linearna regresija i (b) polinomijalna regresija drugog stupnja.

1.6 Odabir modela

Zaključili smo da je učenje bez induktivne pristranosti uzaludno te da je nužno odlučiti se za neki model $\mathcal{H}$. Taj postupak nazivamo **odabir modela** (engl. *model selection*). Budući da se odabir svodi na optimizaciju tzv. **hiperparametara** nekog fiksiranog modela, često se koriste i nazivi **optimizacija modela**, **optimizacija parametara** i **odabir parametara**.

Postavlja se pitanje: kako odabrati idealan model odnosno idealnu induktivnu pristranost? Očito je da što je kapacitet modela veći, to je manja pogreška $E(h|\mathcal{D})$. No važno je imati na umu da svrha hipoteze nije da dobro klasificira primjere za učenje – klasifikacija tih primjera već nam je poznata. Želimo da hipoteza ispravno klasificira nove, buduće primjere, tj. da ima svojstvo **generalizacije**. Hipoteza je bezvrijedna ako nema takvu prediktivnu moć.

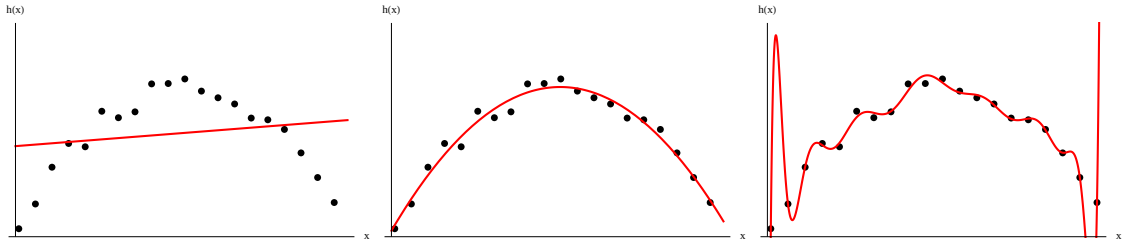
U načelu preferiramo što jednostavnije modele. Za to postoji niz razloga:

1. Jednostavan (ali ne prejednostavan) model bolje generalizira;
2. Jednostavan model je lakše koristiti (manja računalna složenost);
3. Jednostavan model je lakše naučiti (složeniji modeli imaju više parametara koje treba optimirati);
4. Jednostavan model lakše je tumačiti te iz njega ekstrahirati znanje (npr. pravila).

Preferencija jednostavnijih modela nad složenijim modelima ima svoje uporište u filozofiji znanosti, gdje je to načelo poznato kao **Occamova britva** (engl. *Occam's razor*) ili **načelo parsimonije**.[2]

U praksi je problematično odlučiti što je jednostavno, a što ipak prejednostavno. Kako bi hipoteza što bolje generalizirala, trebamo odabrati model $\mathcal{H}$ čija složenost najbolje odgovara složenosti stvarne funkcije koju nastojimo naučiti. To zapravo znači da moramo odabrati odgovarajuću vrstu induktivne pristranosti. Ovdje susrećemo dvije krajnosti:

[2]William od Ockhama (1288–1348), engleski franjevac i skolastičar. Njegovo načelo ontološke parsimonije iskazuje izreka *"Entia non sunt multiplicanda sine necessitate"* (entitete ne treba umnožavati bez potrebe), a u suvremenoj inačici: *"Keep it short and simple"* (KISS).



(a) (b) (c)

Slika 1.6: Regresija funkcije $f(x) = -x^2 + \varepsilon$: (a) podnaučenost (linearna regresija), (b) optimalan model (regresija polinomom drugog stupnja), (c) prenaučenost (regresija polinomom 15. stupnja).

- **Prenaučenost** (engl. *overfitting*)[3] – Ako je model $\mathcal{H}$ previše složen (prevelikog kapaciteta) u odnosu na stvarnu funkciju, hipoteze $h \in \mathcal{H}$ previše su prilagodljive, pa podaci iz $\mathcal{D}$ nisu dovoljni da ih ograniče. Dobivamo "preraskošne" hipoteze koje pretpostavljaju više nego što postoji u stvarnim podatcima. Npr. ako model od dva pravokutnika pokušavamo naučiti na primjerima koji zapravo potječu od samo jednog pravokutnika, ili ako polinomom petog stupnja pokušamo modelirati točke koje su zapravo linearno zavisne. Pored toga, ako je model $\mathcal{H}$ previše složen, hipoteze će se prilagoditi šumu u podacima. U oba slučaja gubimo svojstvo generalizacije.

  Ako je model suviše složen, hipoteze će se vrlo varirati u ovisnosti o skupu primjera za učenje $\mathcal{D}$ – male promjene u skupu $\mathcal{D}$ dovest će do velikih oscilacija u hipotezi. Zbog toga kažemo da složeni modeli imaju visoku **varijancu**. Modeli s visokom varijancom skloni su prenaučenosti.

- **Podnaučenost** (engl. *underfitting*)[4] – Ako je model $\mathcal{H}$ prejednostavan (premalog kapaciteta) u odnosu na stvarnu funkciju, hipoteza se ne može dovoljno prilagoditi podatcima, pa onda loše opisuje i podatke iz samog skupa za učenje $\mathcal{D}$. Ako hipoteza ne može ispravno klasificirati podatke za učenje, izgledno je da će još lošije klasificirati nove primjere, tj. takva će hipoteza loše generalizirati.

  Jednostavan model ima malu varijancu, budući da je rigidniji. S druge strane, u jednostavan model ugrađeno je više pretpostavki, stoga kažemo da jednostavan model ima veću **pristranost**. (Ovdje se misli na pristranost u statističkome smislu: koliko izlaz klasifikatora odstupa od ciljne vrijednosti. Što je induktivna pristranost *bolja*, tj. što model više odgovara podatcima, to je statistička pristranost *manja*.) Modeli s velikom pristranošću skloni su podnaučenosti.

Pojavu podnaučenosti i prenaučenosti kod regresije ilustrira slika 1.6. Podatci su generirani funkcijom $f(x) = -x^2$, uz dodatak slučajnog šuma. Podnaučeni model ne uspijeva modelirati nelinearnost, dok prenaučen model modelira oscilira te ne uspijeva pratiti opći trend. U ovom slučaju optimalan model je polinom drugog stupnja.

Model $\mathcal{H}$ ne smije dakle biti previše složen, kako bi se izbjegla prenaučenost, ali opet ne smije biti suviše jednostavan, kako ne bi bio podnaučen. Odabir modela često se formulira

[3] Također: *pretreniranost*.
[4] Također: *podtreniranost*.

kao **dvojba između pristranosti i varijance** (engl. *bias-variance dilemma*): optimalan model je onaj koji minimizira i pristranost i varijancu, i tako ostvaruje najbolju generalizaciju. Ovo je iskazano tzv. **pretpostavkom induktivnog učenja**:

> *Ako je (1) pogreška hipoteze na dovoljno velikom skupu primjera za učenje mala i (2) ako model nije suviše složen, hipoteza će dobro klasificirati i nove, (3) slične primjere.*

Pretpostavka iskazuje da je generalizacija moguća, pod trima uvjetima: (1) da nije došlo do podnaučenosti, (2) da nije došlo do prenaučenosti te – što se često zaboravlja – (3) da su novi primjeri slični onima na kojima je model bio učen. Pretpostavka je dakle da su primjeri za učenje iz iste distribucije kao i primjeri na kojima će se klasifikator koristiti, tj. da su primjeri za učenje reprezentativni za problem koji rješavamo.

1.6.1 Unakrsna provjera

Postavlja se pitanje: kako utvrditi je li model prenaučen ili podnaučen? Jednostavan način da se to kvantitativno ocijeni jest **unakrsna provjera**[5] (engl. *cross-validation*). Kod unakrsne provjere skup podataka razdvajmo na dva djela: **skup za učenje** (engl. *training set*) i **skup za provjeru** (engl. *validation set*). Model učimo na skupu za učenje, a njegovu generalizacijsku sposobnost provjeravamo na skupu za provjeru. Budući da klasifikator nije učen na primjerima iz skupa za provjeru, na ovaj način možemo vrlo dobro procijeniti kako će se klasifikator ponašati na neviđenim primjerima. Što bolje hipoteza klasificira primjere iz skupa za provjeru, to je bolja njezina generalizacijska sposobnost. Pogreška hipoteze mjerena na skupu koji nije korišten za učenje (engl. *off-training-set error*) naziva se **pogreška generalizacije**. Ako se unakrsna provjera ne koristi za odabir modela, već za utvrđivanje konačne pogreške već odabranog modela, onda se skup na kojemu se mjeri pogreška generalizacije naziva **ispitni skup** (engl. *test set*).

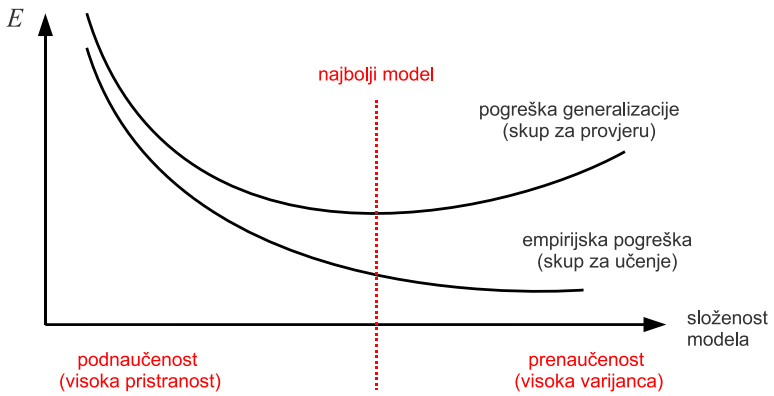
Skup za učenje, skup za provjeru i ispitni skup trebaju biti međusobno disjunktni. Naime, skup za učenje ne smije sadržavati iste primjere kao i skup za provjeru ili ispitni skup jer u protivnom ne možemo odrediti pogrešku generalizacije. Također, ispitni skup ne smije sadržavati iste primjere kao i skup za provjeru jer je skup za provjeru efektivno korišten za izgradnju modela, pa izmjerena pogreška generalizacije opet ne bi bila realna.

Tipično ponašanje empirijske pogreške (mjerene na skupu za učenje) odnosno pogreške generalizacije (mjerene na skupu za provjeru) u ovisnosti o složenosti modela prikazano je slikom 1.7. Za prenaučenost (visoku varijancu) je tipično da je greška na skupu za provjeru znatno veća od greške na skupu za učenje. Kod podnaučenosti (visoke pristranosti) greške su podjednako loše na oba skupa. Razumijevanje ovih odnosa bitno je pri dijagnosticiranju rada klasifikatora odnosno regresijskog postupka.

1.6.2 Drugi načini odabira modela

Unakrsna provjera najčešće je korišten način odabira modela. Ako međutim skup za provjeru nije dostupan (npr. skup za učenje je premalen da bi se razdijelio), model treba odabrati na neki drugi način. Neke od mogućnosti, temeljene na računalnoj teoriji učenja, jesu:

[5] Također: *unakrsna validacija*, *križna validacija*.



Slika 1.7: Empirijska pogreška i pogreška generalizacije u ovisnosti o složenosti modela.

- Načelo minimizacije strukturnog rizika (engl. *Structural risk minimization*, SRM);
- Akaikeov informacijski kriterij (engl. *Akaike information criterion*, AIC);
- Bayesov informacijski kriterij (engl. *Bayesian information criterion*, BIC);
- Načelo minimalnog opisa (engl. *Minimum description length*, MDL).

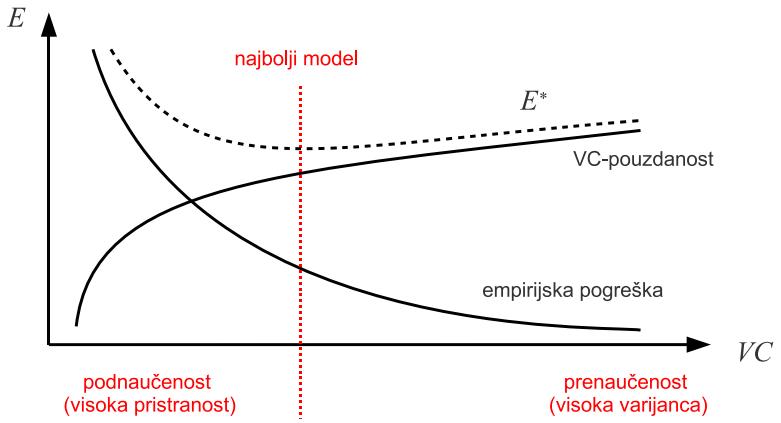
Ilustrirajmo ukratko pristup temeljen na načelu minimizacije strukturnog rizika pomoću VC-dimenzije (engl. *structural risk minimization with VC-dimension*, SRMVC). Pretpostavimo da se moramo odlučiti između modela $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_n$. Modele ćemo poredati tako da $VC(\mathcal{H}_1) \leqslant VC(H_2) \leqslant \cdots \leqslant VC(\mathcal{H}_n)$. Označimo sa E^* očekivanu pogrešku generalizacije. Važan rezultat is statističke teorije učenja jest da je pogreška E^* s vjerojatnošću $1 - \eta$ takva da vrijedi

$$E^*(h) \leqslant E(h|\mathcal{D}) + \underbrace{\sqrt{\frac{VC(\mathcal{H})\big(\log(2N/VC(\mathcal{H})) + 1\big) - \log(\eta/4)}{N}}}_{\text{VC-pouzdanost}}.$$

Vrijednost E^* je zapravo procjena gornje granice pogreške (engl. *upper bound error estimate*) izračunata na temelju empirijske pogreške $E(h|\mathcal{D})$ i tzv. VC-pouzdanosti (pribrojnik čija vrijednost ovisi o VC-dimenziji modela $VC(\mathcal{H})$ i broju primjera za učenje N). S porastom VC-dimenzije, empirijska pogreška $E(h|\mathcal{D})$ će se smanjivati, ali će VC-pouzdanost rasti (slika 1.8). Optimalan model je onaj kod kojeg je zbroj ta dva pribrojnika minimalan.

1.7 Komponente algoritma nadziranog učenja

Navedimo na jednom mjestu što je sve uključeno u postupak nadziranog učenja. Klasifikacijski problem definiran je (doduše nepotpuno) skupom primjera za učenje, $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$. Primjeri su nezavisno i identično distribuirani (iid), što znači da poredak primjera nije bitan i da su primjeri uzorkovani iz iste zajedničke distribucije $P(\mathbf{x}, y)$. Naš je cilj pronaći hipotezu h koja što bolje aproksimira vrijednosti $y^{(i)}$. Pritom moramo definirati sljedeće tri komponente:



Slika 1.8: Procjena pogreške generalizacije i odabir modela temeljem načela SRMVC.

1. **Model** ili prostor hipoteza. Definicija modela kao skupa hipoteza $\mathcal{H}$ nije dovoljno operativna. U praksi je lakše ako model predočimo kao funkciju $h(\cdot|\theta)$ definiranu do na neke parametre θ, a hipotezu kao jednu konkretnu instancu te funkcije. Model je dakle skup funkcija h koje su parametrizirane s θ:

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

dok je hipoteza jedna konkretna funkcija h s fiksiranim vrijednostima parametra θ.

Primjer 1.5 Razmotrimo neke primjere modela. Model pravokutnika u $\mathcal{X} = \mathbb{R}^2$ možemo definirati kao

$$h(x_1, x_2|\theta_{x1}, \theta_{y1}, \theta_{x2}, \theta_{y2}) = \mathbf{1}\{(\theta_{x1} \leqslant x_1 \leqslant \theta_{x2}) \wedge (\theta_{y1} \leqslant x_2 \leqslant \theta_{y2})\}$$

a jedna konkretna hipoteza mogla bi biti $h(x_1, x_2|0, 2, 1, 8)$.

Neki drugi primjeri modela:

- dvodimenzijski linearan model: $h(x_1, x_2|\theta_0, \theta_1, \theta_2) = \mathbf{1}\{\theta_1 x_1 + \theta_2 x_2 + \theta_0 > 0\}$
- kružnica u ravnini: $h(x_1, x_2|\theta) = \mathbf{1}\{x_1^2 + x_2^2 - \theta < 0\}$
- n-dimenzijski linearan model: $h(\mathbf{x}|\boldsymbol{\theta}, \theta) = \mathbf{1}\{\boldsymbol{\theta}^\mathrm{T}\mathbf{x} + \theta > 0\}$
- hiperkugla: $h(\mathbf{x}|\theta) = \mathbf{1}\{\mathbf{x}^\mathrm{T}\mathbf{x} - \theta < 0\}$
- linearna regresija: $h(\mathbf{x}|\boldsymbol{\theta}, \theta_0) = \boldsymbol{\theta}^\mathrm{T}\mathbf{x} + \theta_0$

2. **Funkcija gubitka** (engl. *loss function*) L, koja za dane parametre modela θ izračunava razliku između ciljne vrijednosti $y^{(i)}$i njezine aproksimacije $h(\mathbf{x}^{(i)}|\theta)$. Kod regresije, funkcija gubitka tipično je definirana kao kvadratno odstupanje:

$$L\big(y^{(i)}, h(\mathbf{x}^{(i)}|\theta)\big) = (h(\mathbf{x}^{(i)}|\theta) - y^{(i)})^2\,.$$

Kod klasifikacije, funkcija gubitka tipično je definirana kao

$$L\big(y^{(i)}, h(\mathbf{x}^{(i)}|\theta)\big) = \mathbf{1}\{h(\mathbf{x}^{(i)}|\theta) \neq y^{(i)}\} = |h(\mathbf{x}^{(i)}|\theta) - y^{(i)}|\,. \tag{1.6}$$

Empirijska pogreška definira se kao očekivanje funkcije gubitka nad primjerima iz skupa za učenje, uz pretpostavku uniforme distribucije primjera (tj. $P(\mathbf{x}) = 1/N$):

$$E(\theta|\mathcal{D}) = \mathbb{E}_{\mathcal{D},\theta}[L] = \frac{1}{N}\sum_{i=1}^{N} L\big(y^{(i)}, h(\mathbf{x}^{(i)}|\theta)\big)\,. \tag{1.7}$$

Faktor $1/N$ u (1.7) u načelu je nebitan jer ne utječe na to za koje vrijednosti θ funkcija E doseže minimum (kod regresije se tipično koristi faktor $1/2$ umjesto $1/N$).

U nekim slučajevima pogreške klasifikacije nemaju jednaku težinu, tj. gubitci su asimetrični. Npr., kod klasifikacije zloćudnog tumora, lažno pozitivne slučajeve preferiramo nad lažno negativnim slučajevima, dok je kod filtriranja neželjenih poruka ili ocjene kreditne sposobnosti situacija upravo obrnuta. U takvim se slučajevima funkcija gubitka definira pomoću **matrice gubitka** (engl. *loss matrix*). Matrica gubitka $L = [L_{kj}]$ definira gubitak L_{kj} koji nastaje uslijed klasifikacije primjera $\mathbf{x}$, koji zapravo pripada klasi $\mathcal{C}_k$, u klasu $\mathcal{C}_j$ (vrijedi $L_{kk} = 0$). Poseban slučaj je **matrica gubitka nula-jedan** (engl. *zero-one loss matrix*), za koju $L_{kj} = \mathbf{1}\{k \neq j\}$, i koja odgovara gubitku definiranom sa (1.6).

Primjer 1.6 (Matrica gubitka) Matrica gubitka za klasifikaciju zloćudnog tumora mogla bi biti definirana kao

$$L = \begin{array}{c} \\ \text{rak} \\ \neg\,\text{rak} \end{array} \begin{array}{c} \begin{array}{cc} \text{rak} & \neg\,\text{rak} \end{array} \\ \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \end{array}$$

gdje retci predstavljaju stvarne klase, a stupci odabrane klase. U ovom slučaju lažno negativna klasifikacija nanosi 1000 puta veći gubitak od lažno pozitivne klasifikacije.

3. **Optimizacijski postupak** kojim nalazimo vrijednosti θ^* za koje je empirijska pogreška najmanja, tj.

$$\theta^* = \operatorname*{argmin}_{\theta} E(\theta|\mathcal{D})$$

gdje argmin daje vrijednost argumenta koji minimizira funkciju. U nekim slučajevima (npr. kod linearnog modela regresije) minimum je moguće odrediti analitički, odnosno rješenje je u zatvorenoj formi. Kada to nije moguće, potrebno je koristiti iterativne metode optimizacije. Učenje se dakle svodi na optimizacijski problem, koji se može rješavati raznim numeričkim optimizacijskim postupcima. (Ovu optimizaciju ne treba miješati s optimizacijom složenosti modela, koja je zapravo optimizacija optimizacije, tj. *metaoptimizacija*.)

Algoritmi strojnog učenja međusobno se dakle razlikuju po (1) modelu, (2) funkciji gubitka i (3) optimizacijskom postupku. Ta tri odabira definiraju ujedno i induktivnu pristranost algoritma.

1.8 Pristupi nadziranom učenju

1.8.1 Generativni i diskriminativni modeli

Osnovna podjela klasifikacijskih postupaka jest na generativne i diskriminativne modele. Razlika se svodi na to kako modeliramo pripadnost primjera klasi. Analogna podjela vrijedi i za modele regresije.

Generativni modeli. Ovi modeli pretpostavljaju da je vjerojatnost da primjer $\mathbf{x}$ pripada klasi $\mathcal{C}_j$ proporcionalna zajedničkoj vjerojatnosti primjera $\mathbf{x}$ i klase $\mathcal{C}_j$, tj.:

$$P(\mathcal{C}_j|\mathbf{x}) \propto P(\mathbf{x}, \mathcal{C}_j)\,.$$

Generativni modeli modeliraju zajedničku vjerojatnost $P(\mathbf{x}, \mathcal{C}_j)$. Temeljem te vjerojatnosti može se, primjenom Bayesovog pravila, izračunati posteriorna vjerojatnost $P(\mathcal{C}_j|\mathbf{x})$, tj. vjerojatnost da primjer $\mathbf{x}$ pripada klasi $\mathcal{C}_j$. Ovakav pristup, koji modelira zajedničku razdiobu primjera $\mathbf{x}$ i klasa $\mathcal{C}_j$, nazivamo generativnim jer modelira postupak generiranja podataka. Model se također može koristiti za generiranje sintetičkih primjera u ulaznome prostoru, uzorkovanjem iz zajedničke distribucije $P(\mathbf{x}, \mathcal{C}_j)$.

Tipični generativni modeli su Bayesov klasifikator, mješavina Gaussovih distribucija (engl. *Gaussian mixture model*, GMM), latentna Dirichletova alokacija (engl. *latent Dirichlet allocation*, LDA), Bayesove mreže i skriveni Markovljev model (engl. *hidden Markov models*, HMM).

Diskriminativni modeli. Diskriminativni modeli ne modeliraju zajedničku vjerojatnost $P(\mathbf{x}, \mathcal{C}_j)$, već izravno modeliraju pripadnost primjera $\mathbf{x}$ klasi $\mathcal{C}_j$. Diskriminativni modeli mogu biti probabilistički ili neprobabilistički. Probabilistički diskriminativni modeli izravno modeliraju posteriornu vjerojatnost $P(\mathcal{C}_j|\mathbf{x})$. Neprobabilistički diskriminativni modeli modeliraju funkciju $h(\mathbf{x})$ koja primjeru $\mathbf{x}$ izravno dodjeljuje oznaku klase $\mathcal{C}_j$. Ovaj pristup nije probabilistički jer ne barata s vjerojatnostima.

Logistička regresija tipičan je primjer probabilističkog diskriminativnog modela. Većina diskriminativnih modela nisu probabilistički, međutim u mnogo slučajeva izlaz modela može se koristiti kao indikacija pouzdanost klasifikacije. Tipični primjeri neprobabilističkih diskriminativnih modela jesu perceptron, višeslojni perceptron (engl. *multilayer perceptron*, MLP), stroj s potpornim vektorima (engl. *support vector machine*, SVM), stabla odluke, k-najbližih susjeda i linearna diskriminantna analiza (LDA).

Generativni modeli ponekad se u literaturi nazivaju **zajednički modeli** (engl. *joint models*), dok se probabilistički diskriminativni modeli nazivaju i **uvjetni modeli** (engl. *conditional models*). Diskriminativni modeli (probabilistički i neprobabilistički) ponekad se nazivaju **metode temeljene na granici** (engl. *boundary-oriented methods*). Neprobabilistički diskriminativni modeli nazivaju se i **diskriminacijske funkcije**.

Kod probabilističkih pristupa klasifikacija se odvija kroz dvije faze: fazu **zaključivanja** i fazu **odlučivanja**. U fazi zaključivanja (odnosno učenja)[6] koristimo primjere za učenje kako bismo izračunali $P(\mathcal{C}_j|\mathbf{x})$ (kod generativnih modela na jedan, a kod diskriminativnih na drugi način). U fazi odlučivanja koristimo posteriornu vjerojatnost $P(\mathcal{C}_j|\mathbf{x})$ kako bismo klasificirali nove primjere. Kod neprobabilističkih diskriminativnih modela zaključivanje i odlučivanje stopljeni su u jedan korak.

Generativni modeli imaju niz pogodnosti:

- Ocjena pouzdanosti klasifikacije – izlaz klasifikatora može se tumačiti kao vjerojatnost ili pouzdanost da primjer $\mathbf{x}$ pripada klasi $\mathcal{C}_j$;
- Interpretabilnost rezultata – generativni modeli nude vrlo intuitivnu interpretaciju podataka temeljenu na teoriji vjerojatnosti;
- Ugradnja pozadinskog znanja – u generativne modele lako je ugraditi pozadinsko znanje stručnjaka i takvo znanje kombinirati sa znanjem dobivenim na temelju podataka;
- Odbijanje klasifikacije – ako je za neki primjer $\mathbf{x}$ izlaz klasifikatora manji od unaprijed zadanog praga, klasifikator može odbiti klasificirati primjer $\mathbf{x}$ i tako smanjiti broj pogrešnih klasifikacija. (Primjeri koje klasifikator odbije klasificirati mogu se proslijediti na ručnu klasifikaciju.);
- Nalaženje vrijednosti koje odskaču – marginalizacijom vjerojatnosti $P(\mathbf{x}, \mathcal{C}_j)$ možemo odrediti vjerojatnost primjera $P(\mathbf{x})$ i tako detektirati vrijednosti koje odskaču (engl. *outliers*);
- Minimizacija rizika – u slučajevima kada pogreške klasifikacije nemaju jednaku težinu (tj. kada matrica gubitka $[L_{kj}]$ nije tipa nula-jedan), probabilistički model može donositi optimalne odluke u smislu minimizacije rizika.

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.

1.8.2 Parametarski i neparametarski modeli

Nezavisno od gornje podjele, s obzirom na odnos između broja primjera za učenje i broja parametara modela, nadzirane postupke možemo podijeliti na parametarske i neparametarske modele.

[6] *Zaključivanje* (engl. *inference*) u statističkome smislu odnosi se na izgradnju modela koji opisuje podatke, što je postupak koji je, u kontekstu strojnog učenja, istovjetan učenju.

Tablica 1.1: Podjela nadziranih pristupa klasifikaciji

	Generativni	**Diskriminativni**
Parametarski	• Bayesov klasifikator • Bayesove mreže • latentna Dirichletova alokacija • skriven Markovljev model	• logistička regresija • perceptron • višeslojni perceptron • stroj s potpornim vektorima (primarna formulacija) • linearna diskriminantna analiza
Neparametarski		• k-najbližih susjeda • stabla odluke • klasifikacijska pravila • stroj s potpornim vektorima (dualna formulacija)

Parametarski modeli. Kod parametarskih modela složenost modela ne ovisi o broju primjera za učenje. Konkretno, probabilistički parametarski postupci pretpostavljaju da se podaci pokoravaju nekoj teorijskoj razdiobi (npr. Gaussovoj razdiobi). Učenje se svodi na nalaženje parametara pretpostavljene distribucije, broj kojih ne ovisi o broju primjera.

Neparametarski modeli. Kod neparametarskih modela broj parametara, a time i složenost modela, raste s brojem primjera za učenje. Ovdje ne pretpostavljamo da se podatci pokoravaju nekoj teoretskoj distribuciji. Treba napomenuti da neparametarski modeli (nazivu unatoč) imaju parametre, ali da to nisu parametri neke pretpostavljene distribucije.

Generativni modeli, poput Bayesovog klasifikatora i skrivenog Markovljevog modela, jesu parametarski modeli jer se kod tih postupaka učenje svodi na optimizaciju fiksnog broja parametara neke pretpostavljene distribucije. Linearna i polinomijalna regresija su parametarski modeli budući da pretpostavljaju oblik funkcije koju aproksimiraju i da je broj parametra unaprijed zadan. Diskriminativni modeli, poput perceptrona, višeslojnog perceptrona, stroja s potpornim vektorima (SVM) i linearne diskriminantne analize također su parametarski modeli budući da imaju fiksiran broj parametara i da je složenost modela fiksna. S druge strane, stabla odluke, Parzenovi prozori (regresija) i algoritam k-najbližih susjeda tipični su primjeri neparametarskih modela koji ne pretpostavljaju nikakvu distribuciju primjera i kod kojih broj parametara odnosno složenost modela raste s porastom brojem primjera za učenje. Ovi su odnosi za klasifikacijske postupke sažeto prikazani u tablici 1.1.

Parametarski modeli očito imaju jače pretpostavke o podatcima. Ako su te pretpostavke točne (npr. ako koristimo Gaussovu razdiobu primjera, a primjeri se doista pokoravaju toj razdiobi), onda su u pravilu parametarski modeli bolji od neparametarskih.

Međutim, ako se stvarni podatci ne pokoravaju pretpostavljenoj teorijskoj razdiobi, pogreška klasifikacije bit će razmjerno velika.

1.8.3 Linearni i nelinearni modeli

Konačno, nadzirane modele možemo podijeliti s obzirom na granicu kojom ti modeli u ulaznom prostoru razdvajaju pozitivne primjere od negativnih (u slučaju klasifikacije), odnosno s obzirom na krivulju kojom aproksimiraju funkciju (u slučaju regresije).

Linearni modeli. Povlače linearnu granicu između primjera dviju klasa, odnosno funkciju aproksimiraju linearnim modelom. U dvodimenzijskom prostoru granica je pravac, a u trodimenzijskom granica je ravnina. Općenito, za ulazni prostor $\mathcal{X}$ dimenzije n, granica je $(n-1)$-dimenzijska hiperravnina.

Primjeri linearnih modela su naivan (diskretan) Bayesov klasifikator, logistička regresija, perceptron, stroj s potpornim vektorima (SVM), linearna diskriminantna analiza i linearna regresija.

Nelinearni modeli. Povlače nelinearnu granicu između primjera dviju klasa, odnosno funkciju aproksimiraju nelinarnom hiperravninom.

Primjeri nelinearnih modela su algoritam k-najbližih susjeda, stabla odluke, višeslojni perceptron, SVM s jezgrenim funkcijama (kao i sve druge metode proširene nelinearnim jezgrenim funkcijama) i polinomijalna regresija.

Očito je da nelinearni modeli imaju veći kapacitet (veću VC-dimenziju) i da su zbog toga sposobni rješiti klasifikacijske probleme koji nisu rješivi linearnim modelima (tipičan primjer je problem *isključivo-ili*). U praksi većina interesantnih klasifikacijskih problema nije linearno razdvojiva. To je osobito izraženo kada je broj primjera znatno veći od dimenzije ulaznog prostora, $N \gg n$, budući da je tada ulazni prostor vrlo gusto naseljen. Suprotno tome, ako primjera nije mnogo više nego što je dimenzija ulaznog prostora, ulazni prostor neće biti gusto naseljen i veća je vjerojatnost da je problem linearno razdvojiv. Ova činjenica predstavlja motivaciju za **jezgrene metode** (engl. *kernel methods*), kod kojih se linearna razdvojivost ostvaruje prikladnim povećanjem dimenzije ulaznog prostora.

Poglavlje **2**

Teorija vjerojatnosti

Statističko strojno učenje zasniva se na teoriji vjerojatnosti i statistici. Ovdje razmatramo koncepte koji su ključni za razumijevanje algoritama strojnog učenja.

2.1 Osnove teorije vjerojatnosti

Diskretno statističko obilježje interpretira se kao **diskretna slučajna varijabla** X sa zadanim skupom vrijednosti $\{x_i\}$. Vrijednost $P(X = x)$ jest vjerojatnost da slučajna varijabla X poprimi vrijednost x, tj. vjerojatnost da se slučajna varijabla X realizira kao x. U nastavku ćemo, osim ako postoji mogućnost zabune, $P(X = x)$ pisati kraće kao $P(x)$. Vrijedi $P(x_i) \geqslant 0$ i $\sum_i P(x_i) = 1$, čime je definirana **diskretna razdioba (distribucija) vjerojatnosti**.

Vjerojatnost da slučajna varijabla X poprimi vrijednost x i da slučajna varijabla Y poprimi vrijednost y pišemo kao $P(X = x, Y = y)$, odnosno kraće $P(x, y)$, i nazivamo **zajednička vjerojatnost** (engl. *joint probability*). Korištenjem **pravila zbroja**

$$P(x) = \sum_y P(x, y) \tag{2.1}$$

varijablu Y možemo **marginalizirati**, odnosno možemo izračunati **marginalnu vjerojatnost** $P(x)$ varijable X. Naravno, isto tako možemo marginalizirati varijablu X, odnosno izračunati marginalnu vjerojatnost $P(y)$ varijable Y.

Uvjetna vjerojatnost $P(y|x)$, odnosno vjerojatnost da varijabla Y poprimi vrijednost y, pod uvjetom da je varijabla X poprimila vjerojatnost x, definirana je kao

$$P(y|x) = \frac{P(x, y)}{P(x)} \tag{2.2}$$

što se često iskazuje kao **pravilo umnoška**

$$P(x, y) = P(y|x)P(x)\,. \tag{2.3}$$

Budući da vrijedi simetričnost $P(x, y) = P(y, x)$, primjenom pravila umnoška na lijevu i desnu stranu jednakosti dobivamo

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

iz čega slijedi poznato **Bayesovo pravilo**

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \tag{2.4}$$

Ovdje treba primijetiti da se sve vrijednosti koje se pojavljuju u (2.4) mogu dobiti iz zajedničke vjerojatnosti $P(x, y)$ i to marginalizacijom odnosno normiranjem. Primjenom pravila zbroja i zatim pravila umnoška, marginalna vjerojatnost $P(x)$ može se izraziti kao

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

što je vrlo pogodno jer su onda izrazi u nazivniku isti kao i oni u brojniku. (Primijetite da se marginalna vjerojatnost $P(x)$ mogla izraziti i kao $\sum_y P(y|x)P(x)$, ali time ne bismo ništa dobili.) Bayesovo pravilo možemo dakle pisati kao

$$P(y|x) = \frac{P(x|y)P(y)}{\sum_y P(x|y)P(y)}\,. \tag{2.5}$$

U gornjim razmatranjima ograničili smo se na slučaj diskretne slučajne varijable. Za kontinuiranu (neprekidnu) slučajnu varijablu definira se **funkcija gustoće vjerojatnosti** (engl. *probability density function*, PDF) $p(x)$, za koju vrijedi[1]

$$p(x) \geqslant 0 \tag{2.6}$$

$$\int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1 \tag{2.7}$$

Vjerojatnost da kontinuirana slučajna varijabla X poprimi vrijednost iz intervala $[a, b]$ $(a, b \in \mathbb{R},\ a \leqslant b)$ dana je s

$$P(a \leqslant X \leqslant b) = \int_a^b p(x)\,\mathrm{d}x\,.$$

Primijetite da za kontinuiranu varijablu X vrijedi $P(X = a) = 0$, tj. vjerojatnost da kontinuirana varijabla poprimi bilo koju pojedinačnu vrijednost jednaka je nuli. Kažemo da funkcijom gustoće vjerojatnosti $p(x)$ definirana **kontinuirana razdioba (distribucija) vjerojatnosti**.[2]

Pravilo zbroja i pravilo umnoška vrijede i za kontinuirane varijable (a također i za mješovite razdiobe, tj. kombinacije diskretnih i kontinuiranih varijabli):

$$p(x) = \int_{-\infty}^{\infty} p(x, y)\,\mathrm{d}y \tag{2.8}$$

$$p(x, y) = p(y|x)p(x) \tag{2.9}$$

[1] Funkcija gustoće vjerojatnosti u matematičkoj literaturi uobičajeno se označava s $f(x)$. Mi ćemo u nastavku koristiti oznaku $p(x)$, koja je tipična u literaturi za strojno učenje.

[2] U nastavku ćemo koristiti izraz *"gustoća $p(x)$"* ili (pomalo neprecizno) *"razdioba $p(x)$"*, mislivši pritom na funkciju gustoće vjerojatnosti $p(x)$. Kada mislimo na konkretnu vrijednost te funkcije u točci x, koristit ćemo izraz *"vrijednost $p(x)$"*.

2.1.1 Očekivanje i varijanca

Prosječna vrijednost diskretne slučajne varijable X čija je razdioba $P(x)$ naziva se **(matematičko) očekivanje** varijable X i definira kao

$$\mathbb{E}[X] = \sum_x xP(x) \tag{2.10}$$

U slučaju kontinuirane slučajne varijable X s gustoćom vjerojatnosti $p(x)$, očekivanje je

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x\, p(x)\, \mathrm{d}x\,. \tag{2.11}$$

Vrijedi:

$$\mathbb{E}[aX + b] = a\mathbb{E}[X] + b \qquad (a, b \in \mathbb{R}) \tag{2.12}$$
$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y] \tag{2.13}$$

Varijanca slučajne varijable X iskazuje koliko vrijednosti varijable variraju oko očekivane vrijednosti:

$$\mathrm{Var}(X) = \sigma_X^2 = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2\,. \tag{2.14}$$

(Provjerite da druga jednakost doista vrijedi.) Iz (2.14) slijedi

$$\mathrm{Var}(aX) = \mathbb{E}\Big[(aX)^2\Big] - \mathbb{E}[aX]^2 = a^2\mathbb{E}[X^2] - a^2\mathbb{E}[X]^2 = a^2\mathrm{Var}(X)\,. \tag{2.15}$$

Kovarijanca opisuje odnos između dviju slučajnih varijabli, odnosno opisuje u kojoj mjeri slučajne varijable zajednički variraju oko svojih očekivanih vrijednosti. Kovarijanca varijabli X i Y definirana je kao

$$\mathrm{Cov}(X, Y) = \sigma_{X,Y} = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]\,. \tag{2.16}$$

(Provjerite da druga jednakost doista vrijedi.) Vrijedi $\mathrm{Cov}(X, Y) = \mathrm{Cov}(Y, X)$ i $\mathrm{Cov}(X, X) = \mathrm{Var}(X)$ odnosno $\sigma_{X,X} = \sigma_X^2$.

Za slučajne varijable X i Y za koje vrijedi $\mathrm{Var}(X) > 0$ i $\mathrm{Var}(Y) > 0$ definiran je **koeficijent korelacije**

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y}\,. \tag{2.17}$$

Koeficijent korelacije upućuje na mjeru linearne zavisnosti među varijablama X i Y. Za savršenu pozitivnu linearnu ovisnost vrijedi $\rho_{X,Y} = 1$, dok za savršenu negativnu linearnu ovisnost vrijedi $\rho_{X,Y} = -1$.

2.1.2 Nezavisnost varijabli

Dvije slučajne varijable X i Y su **(stohastički) nezavisne** akko za sve intervale A i B, $A, B \subseteq \mathbb{R}$, vrijedi

$$P(X \in A, Y \in B) = P(X \in A)P(Y \in B)\,.$$

Konkretno, varijable X i Y su nezavisne akko:

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

što je ekvivalentno s

$$P(X|Y) = P(X) \qquad \text{i} \qquad P(Y|X) = P(Y)\,.$$

Intuitivno, varijable X i Y su nezavisne ako znanje o ishodu varijable Y ni na koji način ne utječe na vjerojatnost ishoda varijable X (i obrnuto).

Za nezavisne varijable X i Y vrijedi

$$\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y] \tag{2.18}$$
$$\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) \tag{2.19}$$
$$\mathrm{Cov}(X,Y) = \rho_{X,Y} = 0 \tag{2.20}$$

Nezavisne varijable su nekorelirane, no obrat općenito ne vrijedi: koeficijent korelacije može biti jednak nuli, a da su varijable ipak nelinearno zavisne (koeficijent korelacije mjeri isključivo linearnu zavisnost varijabli).

U strojnom učenju važan koncept jest uvjetna nezavisnost varijabli.

Definicija 8 (Uvjetna nezavisnost.) *Slučajne varijable X i Y su* **uvjetno nezavisne** *uz danu varijablu Z, što označavamo kao $X \perp Y|Z$, akko*

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

ili, ekvivalentno

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

Intuitivno, varijable X i Y su uvjetno nezavisne ako, jednom kada je poznat ishod varijable Z, znanje o ishodu varijable Y ni na koji način ne utječe na ishod varijable X (i obrnuto). Općenito, nezavisnost varijabli X i Y ne implicira njihovu uvjetnu nezavisnost ni po kojoj varijabli, niti obrnuto.

2.1.3 Višedimenzijska slučajna varijabla

Istodobno opažanje više statističkih obilježja modeliramo n-dimenzijskim slučajnim vektorom $(X_1, \ldots, X_n)$. Za slučajan vektor definirana je **matrica kovarijacije**[3] $\boldsymbol{\Sigma}$ dimenzija $n \times n$, s elementima:

$$\boldsymbol{\Sigma}_{ij} = \mathrm{Cov}(X_i, X_j) = \sigma_{ij} = \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])] = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]\,.$$

Odnosno, u matričnom računu, kovarijacijska je matrica definirana kao:

$$\boldsymbol{\Sigma} = \mathbb{E}\Big[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^{\mathrm{T}}\Big]\,. \tag{2.21}$$

[3] Također *kovarijancna matrica* i *disperzijska matrica*.

Kovarijacijska matrica je kvadratna simetrična matrica koja na dijagonali ima varijance varijabli $X_1, \ldots, X_n$, a izvan dijagonale kovarijance svih parova varijabli:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \ldots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \ldots & \mathrm{Cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_n, X_1) & \mathrm{Cov}(X_n, X_2) & \ldots & \mathrm{Var}(X_n) \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \ldots & \rho_{1n}\sigma_1\sigma_n \\ \rho_{21}\sigma_2\sigma_1 & \sigma_2^2 & \ldots & \rho_{2n}\sigma_2\sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1}\sigma_n\sigma_1 & \rho_{n2}\sigma_n\sigma_2 & \ldots & \sigma_n^2 \end{pmatrix}$$

Primjer 2.1 (Kovarijacijska matrica) Razmotrimo slučaj dvodimenzijske kontinuirane slučajne varijable, $\mathbf{X} = (X_1, X_2)$. Neka su varijance $\sigma_1^2 = 1$ i $\sigma_2^2 = 4$ te neka su varijable pozitivno korelirane s faktorom $\rho_{12} = \rho_{21} = 0.75$. Kovarijacijska matrica je

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{21}\sigma_2\sigma_1 & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} 1 & 1.5 \\ 1.5 & 4 \end{pmatrix}$$

Ako su varijable $X_1, \ldots, X_n$ međusobno nezavisne, onda $\mathrm{Cov}(X_i, X_j) = 0$ i kovarijacijska je matrica dijagonalna matrica, $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_i^2)$. Nadalje, ako nezavisne varijable $X_1, \ldots, X_n$ imaju jednaku varijancu, onda $\sigma_i^2 = \sigma^2$, pa kovarijacijska matrica degenerira u $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$. Takav slučaj nazivamo **izotropnom kovarijancom**.

2.2 Teorijske razdiobe

U nastavku ćemo se prisjetiti nekoliko vjerojatnosnih razdioba koje su nam potrebne u strojnom učenju.

2.2.1 Bernoullijeva razdioba

Bernuoullijeva razdioba jest razdioba diskretne slučajne varijable X koja ima dva moguća ishoda: događaj je ili nastupio ili nije, $x \in \{0, 1\}$. Razdioba je dana kao

$$P(X = x|\mu) = \mu^x(1-\mu)^{1-x} = \begin{cases} \mu & \text{ako } X = 1 \\ 1-\mu & \text{inače} \end{cases} \tag{2.22}$$

gdje parametar μ definira vjerojatnost nastupanja događaja, tj. vjerojatnost $P(X = 1)$. Iz (2.10), (2.14) i (2.22) slijedi da su očekivana vrijednost i varijanca

$$\begin{aligned} \mathbb{E}[X] &= \mu \\ \mathrm{Var}(X) &= \mu(1-\mu) \end{aligned}$$

2.2.2 Multinomijalna razdioba

Razmotrimo poopćenje Bernoullijeve razdiobe na slučaj kada slučajna varijabla X može poprimiti jedno od K međusobno isključivih stanja, $K \geqslant 2$ (npr. jednu od K klasa). Takvu varijablu nazivamo **multinomijalna varijabla**. Multinomijalnu varijablu prikazujemo kao vektor indikatorskih (binarnih) varijabli

$$\mathbf{x} = (x_1, x_2, \ldots, x_K)^{\mathrm{T}}$$

gdje je $x_k = 1$ ako je ishod varijable k, a inače $x_k = 0$. Npr. $\mathbf{x} = (0, 0, 1, 0)$ označava da je multinomijalna varijabla poprimila treće stanje od četiri mogućih stanja. Pritom vrijedi $\sum_k x_k = 1$ (ishodi su međusobno isključivi). Označimo vjerojatnost $P(X_k = 1)$ sa μ_k. Razdioba je dana s

$$P(\mathbf{X} = \mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k} \tag{2.23}$$

gdje je $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)^{\mathrm{T}}$, pri čemu za parametre μ_k vrijedi $\sum_k \mu_k = 1$ i $\mu_k \geqslant 0$, budući da predstavljaju vjerojatnosti.

Navedena razdioba naziva se **kategorička razdioba** i zapravo je poseban slučaj **multinomijalne razdiobe** kod koje je broj eksperimenata jednak 1.[4] Multinomijalnu varijablu ne treba miješati s višedimenzijskom varijablom odnosno slučajnim vektorom (premda ovdje multinomijalnu varijablu modeliramo pomoću slučajnog vektora).

2.2.3 Gaussova razdioba

Neprekidna slučajna varijabla X ima Gaussovu (normalnu) razdiobu sa srednjom vrijednošću μ i varijancom σ^2, što označavamo kao $\mathcal{N}(\mu, \sigma^2)$, ako je njezina gustoća vjerojatnosti jednaka

$$p(X = x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}. \tag{2.24}$$

Za normalnu razdiobu vrijedi

$$\mathbb{E}[X] = \mu \tag{2.25}$$

$$\mathrm{Var}(X) = \sigma^2 \tag{2.26}$$

2.2.4 Multivarijatna Gaussova razdioba

Poopćenjem Gaussove razdiobe na n dimenzija dobivamo **multivarijatnu Gaussovu razdiobu**:

$$p(\mathbf{X} = \mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \right\} \tag{2.27}$$

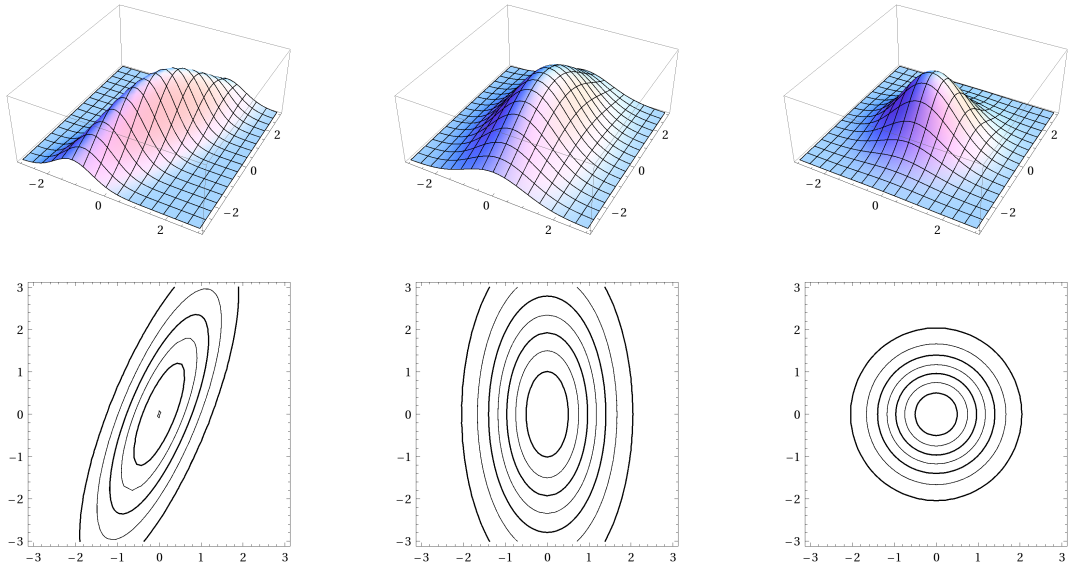
gdje je $\boldsymbol{\mu}$ n-dimenzijski vektor srednje vrijednosti, a $\boldsymbol{\Sigma}$ je kovarijacijska matrica dimenzija $n \times n$. Da bi razdioba (3.24) bila dobro definirana, kovarijacijska matrica $\boldsymbol{\Sigma}$ mora biti pozitivno definitna,[5] jer tada ima pozitivnu determinantu i nesingularna je (ima inverz). Vrijednost gustoće funkcijski zavisi o $\mathbf{x}$ preko **kvadratne forme**[6]

$$\Delta^2 = (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

[4] U strojnom učenju i nekim drugim područjima ova se razlika često zanemaruje.

[5] Matrica $\mathbf{A}$ je *pozitivno definitna* akko $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} > 0$ za svaki ne-nul vektor $\mathbf{x}$.

[6] Za kvadratnu matricu $\mathbf{A}$ funkcija $f : \mathbb{R}^n \to \mathbb{R}$ definirana kao $f(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}$ naziva se *kvadratna forma*. Pomoću skalarnog produkta kvadratna se forma može sažetije napisati kao $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} = (\mathbf{A}\mathbf{x}|\mathbf{x}) = (\mathbf{x}|\mathbf{A}\mathbf{x})$.



(a) (b) (c)

Slika 2.1: Gustoća vjerojatnosti dvodimenzijske normalno distribuirane varijable sa srednjom vrijednošću $\boldsymbol{\mu} = (0,0)^{\mathrm{T}}$: (a) slučaj za $\sigma_1^2 = 1$, $\sigma_2^2 = 4$, $\rho_{12} = 0.75$, (b) za slučaj $\sigma_1^2 = 1$, $\sigma_2^2 = 4$, $\rho_{12} = 0$, (c) za slučaj $\sigma_1^2 = \sigma_2^2 = 1$, $\rho_{12} = 0$.

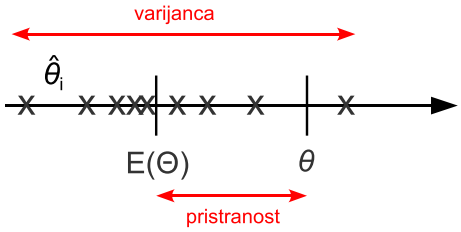
koja se javlja u eksponentu. Vrijednost Δ nazivamo **Mahalanobisova udaljenost** između $\mathbf{x}$ i $\boldsymbol{\mu}$. Mahalanobisova udaljenost je poopćenje euklidske udaljenosti koje je neosjetljivo na razlike u varijanci između pojedinih dimenzija te na korelacije između varijabli. Mahalanobisova udaljenost svodi se na euklidsku za $\boldsymbol{\Sigma} = \mathbf{I}$.

Za multivarijatnu normalnu razdiobu vrijedi

$$\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu} \tag{2.28}$$

$$\mathrm{Cov}(X_i, X_j) = \boldsymbol{\Sigma}_{ij} \tag{2.29}$$

Na slici 2.1 prikazane su gustoće vjerojatnosti i njima odgovarajuće konture jednake gustoće za dvodimenzijsku normalno distribuiranu varijablu sa srednjom vrijednošu $\boldsymbol{\mu} = (0,0)^{\mathrm{T}}$. Slika 2.1a prikazuje najopćenitiji slučaj u kojemu su varijance pojedinih komponenti različite i korelirane. U tom slučaju konture Gaussove razdiobe su nakošene elipse (odnosno općenito hiperelipsoidi). Slika 2.1b prikazuje slučaj nekoreliranih varijabli s različitim varijancama (dijagonalna kovarijacijska matrica). U tom slučaju konture Gaussove razdiobe su elipse čije su osi poravnate s apscisom i ordinatom (odnosno općenito hiperelipsoidi poravnati s osima). Slika 2.1c prikazuje slučaj nekoreliranih i identično distribuiranih komponenti (izotropna kovarijacijska matrica). U ovom slučaju konture Gaussove razdiobe su kružnice (odnosno općenito hipersfere).



Slika 2.2: Pristranost i varijanca procjenitelja: θ je parametar koji treba procijeniti, $\hat{\theta}_i$ su pojedinačne procjene na različitim uzorcima, a $\mathbb{E}[\Theta]$ je očekivana vrijednost procjenitelja.

2.3 Procjena parametara

Osnovna ideja procjene parametara jest na temelju uzorka najprije izračunati vrijednost određene slučajne varijable, tzv. **statistike**, koju koristimo kao **procjenitelj** (estimator) nepoznatog parametra $\boldsymbol{\theta}$ populacije (odnosno teorijske razdiobe).

2.3.1 Procjenitelj

Definicija 9 (Statistika, procjenitelj i procjena) *Neka je $(X_1, X_2, \ldots, X_n)$ uzorak, odnosno n-torka slučajnih varijabli koje su iid. Slučajna varijabla $\Theta = g(X_1, X_2, \ldots, X_n)$ naziva se* **statistika**. *Statistika Θ je* **procjenitelj (estimator)** *parametra populacije θ. Vrijednost procjenitelja $\hat{\theta} = g(x_1, x_2, \ldots, x_n)$ naziva se* **procjena**.

Statistika je dakle bilo kakva funkcija koja ovisi o uzorku, dok je procjenitelj ona statistika koja se koristi za procjenu nekog parametra populacije. Procjenitelj je slučajna varijabla, što znači da ima svoje očekivanje i varijancu. Očekivanje i varijancu procjenitelja koristimo kako bismo ocijenili kvalitetu procjenitelja, odnosno kako bismo ocijenili koliko dobro procjenitelj Θ procjenjuje parametar populacije θ.

Definicija 10 (Nepristran procjenitelj) *Procjenitelj Θ je* **nepristran (centriran) procjenitelj** *parametra θ akko $\mathbb{E}[\Theta] = \theta$.*

Vrijednost procjenitelja $\hat{\theta}$ na konkretnom uzorku $\mathcal{D}$ može biti različita od prave vrijednosti parametra θ, ali ako je procjenitelj nepristran, onda će se kod ponavljanja eksperimenta prosječna vrijednost procjenitelja približiti stvarnoj vrijednosti parametra. **Pristranost** procjenitelja definirana je kao

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

Idealan procjenitelj je nepristran i ima malu varijancu. Primijetite da su ove dvije veličine nezavisne te da pristranost ovisi o vrijednosti parametra, dok varijanca ne ovisi. Pristranost i varijanca procjenitelja skicirani su na slici 2.2.

Dodatno poželjno svojstvo jest da s porastom veličine uzorka (odnosno broja primjera za učenje) procjena sve manje odstupa od svoje očekivane vrijednosti. Drugim riječima, želimo da s povećanjem uzorka varijanca procjenitelja teži k nuli.

Definicija 11 (Valjan procjenitelj) *Nepristan procjenitelj Θ je* **valjan (konzistentan)** *procjenitelj ako $\lim_{N\to\infty} \mathrm{Var}(\Theta) = 0$.*

Primjer 2.2 (Procjenitelj srednje vrijednosti) Neka je X slučajna varijabla s vrijednostima $x \in \mathbb{R}$. Označimo očekivanje i varijancu ove varijable s $\mathbb{E}[X] = \mu$ odnosno $\mathrm{Var}(X) = \sigma^2$. Ova se varijabla pokorava razdiobi sa srednjom vrijednošću μ i varijancom σ^2. Neka je $\{x^{(i)}\}_{i=1}^N$ uzorak ove slučajne varijable.

Prave vrijednosti parametara μ i σ^2 nisu nam poznate, ali ih možemo procijeniti na temelju uzorka. Tako za procjenu parametra μ (srednja vrijednost razdiobe) možemo koristiti **sredinu uzorka**:

$$\hat{\mu} = \frac{1}{N}\sum_i x^{(i)}$$

Sredina uzorka nepristran je procjenitelj parametra razdiobe μ, budući da vrijedi

$$\mathbb{E}[\hat{\mu}] = \mathbb{E}\Big[\frac{\sum_i^N x^{(i)}}{N}\Big] = \frac{1}{N}\sum_i^N \mathbb{E}[X^{(i)}] = \frac{N\mu}{N} = \mu\,.$$

Pokažimo da je $\hat{\mu}$ ujedno i valjan procjenitelj. Vrijedi

$$\mathrm{Var}(\hat{\mu}) = \mathrm{Var}\Big(\frac{\sum_i^N x^{(i)}}{N}\Big) = \frac{1}{N^2}\sum_i^N \mathrm{Var}(X^{(i)}) = \frac{N\sigma^2}{N^2} = \frac{\sigma^2}{N} \tag{2.30}$$

pri čemu smo iskoristili jednakost (2.15) te jednakost (2.19), koja vrijedi jer su varijable $X^{(i)}$ iid. Očito

$$\lim_{N\to\infty}\frac{\sigma^2}{N} = 0$$

pa zaključujemo da je $\hat{\mu}$ valjan procjenitelj.

Primjer 2.3 (Procjenitelj varijance) Neka su $x^{(i)} \in \mathbb{R}$ uzorci slučajne varijable X koja se pokorava razdiobi sa srednjom vrijednošću μ i varijancom σ^2, tj. $\mathbb{E}[X] = \mu$ i $\mathrm{Var}(X) = \sigma^2$. Neka je $\hat{\mu}$ nepristran procjenitelj srednje vrijednosti (v. primjer 2.2). Provjerimo je li procjenitelj

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^N (x^{(i)} - \hat{\mu})^2$$

nepristran procjenitelj parametra σ^2. Uvrštavanjem $\sum_i x^{(i)} = N\hat{\mu}$ dobivamo

$$\hat{\sigma}^2 = \frac{1}{N}\Big(\sum_{i=1}^N (x^{(i)})^2 - 2N\hat{\mu}^2 + N\hat{\mu}^2\Big) = \frac{1}{N}\Big(\sum_{i=1}^N (x^{(i)})^2 - N\hat{\mu}^2\Big)\,.$$

Očekivanje je

$$\mathbb{E}[\hat{\sigma}^2] = \mathbb{E}\Big[\frac{1}{N}\Big(\sum_{i=1}^N (x^{(i)})^2 - N\hat{\mu}^2\Big)\Big] = \frac{1}{N}\Big(N\mathbb{E}[(X)^2] - N\mu^2\Big) = \mathbb{E}[X^2] - \mathbb{E}[\hat{\mu}^2]\,.$$

Iz (2.14) slijedi $\mathbb{E}[X^2] = \mathrm{Var}(X) + \mathbb{E}[X]^2$, pa vrijedi $\mathbb{E}[X^2] = \sigma^2 + \mu^2$. Slično, korištenjem (2.30), dobivamo $\mathbb{E}[\hat{\mu}^2] = \sigma^2/N + \mu^2$. Uvrštavanjem ovih jednakosti u gornju jednakost dobivamo

$$\mathbb{E}[\hat{\sigma}^2] = \sigma^2 + \mu^2 - \sigma^2/N - \mu^2 = \frac{N-1}{N}\sigma^2\,.$$

Budući da $\mathbb{E}[\hat{\sigma}^2] \neq \sigma^2$, zaključujemo da $\hat{\sigma}^2$ nije nepristran procjenitelj varijance σ^2. Preciznije, $\hat{\sigma}$ **podcjenjuje** pravu varijancu jer $\hat{\sigma} < \sigma$. Pristranost procjenitelja $\hat{\sigma}$ je

$$b(\hat{\sigma}^2) = \frac{N-1}{N}\sigma^2 - \sigma^2 = -\frac{\sigma^2}{N}$$

i ona se smanjuje kako $N \to \infty$. Za manje uzorke nepristranost može predstavljati problem i tada je procjenitelj moguće korigirati (učiniti nepristranim) tako da ga se pomnoži s $N/(N-1)$, tj. da se kao procjenitelj varijance koristi:

$$\hat{\sigma}^2_{\text{nepr.}} = \frac{1}{N-1}\sum_{i=1}^{N}(x^{(i)} - \hat{\mu})^2$$

2.3.2 Pogreška procjenitelja

Za procjenitelj Θ srednju kvadratnu pogrešku definiramo kao

$$r(\Theta, \theta) = \mathbb{E}[(\Theta - \theta)^2] \tag{2.31}$$

i to je u stvari funkcija rizika uz kvadratnu funkciju gubitka (prisjetite se da je funkcija rizika definirana kao očekivanje funkcije gubitka). Pokažimo kako se izraz (2.31) može rastaviti na pristranost i varijancu. Sasvim općenito, za slučajnu varijablu X i konstantu c vrijedi:

$$\begin{aligned}
\mathbb{E}[(X-c)^2] &= \mathbb{E}\Big[(X - \mathbb{E}[X] + \mathbb{E}[X] - c)^2\Big] \\
&= \mathbb{E}\Big[\big((X - \mathbb{E}[X]) + (\mathbb{E}[X] - c)\big)^2\Big] \\
&= \mathbb{E}\Big[(X - \mathbb{E}[X])^2 + 2(X - \mathbb{E}[X])(\mathbb{E}[X] - c) + (\mathbb{E}[X] - c)^2\Big] \\
&= \mathbb{E}\Big[(X - \mathbb{E}[X])^2\Big] + 2\underbrace{\mathbb{E}\Big[(X - \mathbb{E}[X])(\mathbb{E}[X] - c)\Big]}_{\mathbb{E}[(X-\mu)(\mu-c)] = (\mu-c)\mathbb{E}[X-\mu] = 0} + \mathbb{E}\Big[(\mathbb{E}[X] - c)^2\Big] \\
&= \mathbb{E}\Big[(X - \mathbb{E}[X])^2\Big] + (\mathbb{E}[X] - c)^2 \,.
\end{aligned} \tag{2.32}$$

Navedena jednakost vrijedi jer je $\mathbb{E}[X] = \mu$ konstanta, pa vrijedi $\mathbb{E}[X - \mu] = \mathbb{E}[X] - \mu = 0$, a također $\mathbb{E}[(\mathbb{E}[X] - c)^2] = (\mathbb{E}[X] - c)^2$. Primijenimo li (2.32) na (2.31), dobivamo:

$$r(\Theta, \theta) = \underbrace{\mathbb{E}\Big[(\Theta - \mathbb{E}[\Theta])^2\Big]}_{\text{varijanca}} + \underbrace{(\mathbb{E}[\Theta] - \theta)^2}_{\text{pristranost}^2} = \text{Var}(\Theta) + b_\theta(\Theta)^2 \,. \tag{2.33}$$

Srednja kvadratna pogreška procjenitelja može se dakle rastaviti na varijancu i kvadrat pristranosti. Pristranost nam kazuje koliko procjenitelj griješi neovisno o varijacijama u uzorku, dok varijanca kazuje koliko vrijednost procjenitelja varira oko očekivane vrijednosti kako uzorak varira. Primijetite da, ako je procjenitelj nepristran, $b_\theta(\Theta) = 0$, srednja kvadratna pogreška procjenitelja jednaka je njegovoj varijanci.

2.4 Procjenitelj najveće izglednosti

Najjednostavniji i u praksi najčešće korišten procjenitelj jest **procjenitelj najveće izglednosti** (engl. *maximum likelihood estimation*, MLE).[7] Neka je dan skup neoznačenih primjera $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$, za koje pretpostavljamo da su nezavisni i da potječu od identične razdiobe (pretpostavka iid). Pretpostavljamo da je riječ o nekoj nama poznatoj razdiobi $p(\mathbf{x}|\theta)$, definiranoj do na parametre $\boldsymbol{\theta}$:

$$\mathbf{x}^{(i)} \sim p(\mathbf{x}|\boldsymbol{\theta})\,.$$

Naš je zadatak odrediti *najizglednije* parametre $\boldsymbol{\theta}$, odnosno takve parametre da uzorkovanje primjera $\mathbf{x}^{(i)} \in \mathcal{D}$ bude što je vjerojatnije moguće. Naime, u nedostatku dodatnih spoznaja, jedino što opravdano možemo pretpostaviti jest da je baš za uzorak $\mathcal{D}$ – dakle uzorak koji imamo na raspolaganju – bilo najvjerojatnije da bude izvučen iz populacije, i da je to razlog zašto je realiziran baš taj uzorak. Budući da pretpostavljamo da su primjeri $\mathbf{x}^{(i)}$ iid, to je gustoća vjerojatnosti uzorka $\mathcal{D}$ jednaka umnošku gustoća vjerojatnosti pojedinačnih primjera:

$$p(\mathcal{D}|\boldsymbol{\theta}) = p(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) \;\equiv \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})\,. \tag{2.34}$$

Ovime je definirana gustoća vjerojatnosti za uzorak $\mathcal{D}$, parametrizirana parametrima $\boldsymbol{\theta}$ (ako je varijabla $\mathbf{x}$ diskretna, umjesto gustoće vjerojatnosti $p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$ koristi se vjerojatnost $P(\mathbf{x}^{(i)}|\boldsymbol{\theta})$). Međutim, moguće je gledati i obrnuto, pa reći da je to funkcija od $\boldsymbol{\theta}$, uz fiksirani parametar $\mathcal{D}$. Tada tu funkciju nazivamo **funkcija izglednosti** (engl. *likelihood function*) i označavamo s $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$. Ovo su zapravo dva različita pogleda na istu stvar: funkcija gustoće $p(\mathcal{D}|\boldsymbol{\theta})$ preslikava $\mathcal{D} \mapsto p(\mathcal{D}|\boldsymbol{\theta})$, dok funkcija izglednosti preslikava $\boldsymbol{\theta} \mapsto p(\mathcal{D}|\boldsymbol{\theta})$. Treba naglasiti da funkcija izglednosti $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$ nije funkcija gustoće vjerojatnosti od $\boldsymbol{\theta}$ te da ona ne mora biti normirana, tj. ne mora vrijediti $\int_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})\,\mathrm{d}\boldsymbol{\theta} = 1$. Očito je da $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$ nije normirana budući da općenito ne vrijedi $\int_y p(x|y)\mathrm{d}y = 1$.

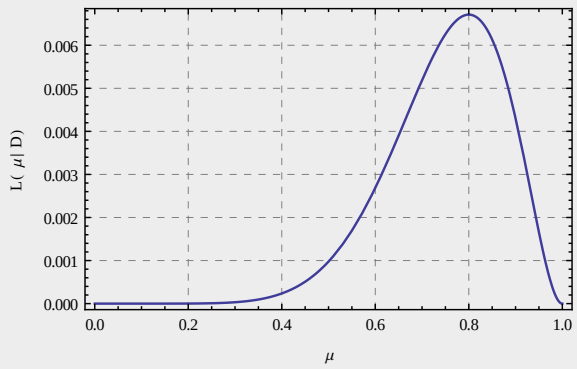
Primjer 2.4 (Funkcija izglednosti) U 10 bacanja novčića ($N = 10$), glavu (H) dobivamo 8 puta, a pismo (T) 2 puta. Ishodi bacanja novčića čine naš uzorak $\mathcal{D}$.

Neka je parametar μ jednak vjerojatnosti da u bacanju novčića dobijemo glavu. Vrijedi dakle $P(X = H|\mu) = \mu$ i $P(X = T|\mu) = 1 - \mu$. Primijetite da je ovdje riječ o diskretnoj varijabli koja se pokorava Bernoullijevoj razdiobi parametriziranoj parametrom μ prema (2.22). Funkcija izglednosti za uzorak $\mathcal{D}$ je

$$\mathcal{L}(\mu|\mathcal{D}) = P(\mathcal{D}|\mu) = \prod_{i=1}^{10} P(X|\mu) = \mu^8 \cdot (1-\mu)^2$$

te izgleda ovako

[7] Također: *metoda najveće vjerodostojnosti, metoda najveće vjerojatnosti, ML-metoda.*



Možemo se, na primjer, pitati kolika je izglednost da je vrijednost parametra μ jednaka 0.5. Vidimo da $\mathcal{L}(\mu = 0.5|\mathcal{D}) \approx 0.001$. Važno je naglasiti da to ne znači da je vjerojatnost da $\mu = 0.5$, uz zadani $\mathcal{D}$, jednaka 0.001 (funkcija izglednosti nije funkcija gustoće vjerojatnosti). Međutim, vrijedi obrnuto: 0.001 jest vjerojatnost uzorka $\mathcal{D}$, ako $\mu = 0.5$, tj. $P(\mathcal{D}|\mu = 0.5) = 0.001$.

Iz slike vidimo da je za uzorak najizglednije da $\mu = 0.8$, i to je vrijednost parametra kojom se maksimizira realizacija uzorka.

Kod procjene metodom najveće izglednosti naš je cilj pronaći $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$ koji maksimizira funkciju izglednosti, budući da time maksimiziramo vjerojatnost pojavljivanja uzorka $\mathcal{D}$:

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})\,.$$

Na slici 2.3 ta je ideja ilustrirana za Gaussovu razdiobu. Parametri Gaussove razdiobe su $\boldsymbol{\theta} = (\mu, \sigma^2)$, gdje je μ srednja vrijednost a σ^2 je varijanca. Gaussova razdioba koja maksimizira vjerojatnost danog uzorka je upravo ona koja je prikazana na slici 2.3a. Parametri te razdiobe su $\mu = 5$ i $\sigma^2 = 7$, i to su najizgledniji parametri za dani uzorak. Bilo koji drugi parametri promijenili bi izgled Gaussove krivulje i to bi dovelo do smanjenja umnoška $\prod p(x|\mu, \sigma^2)$, dakle do smanjenja vjerojatnosti uzorka $\mathcal{D}$.

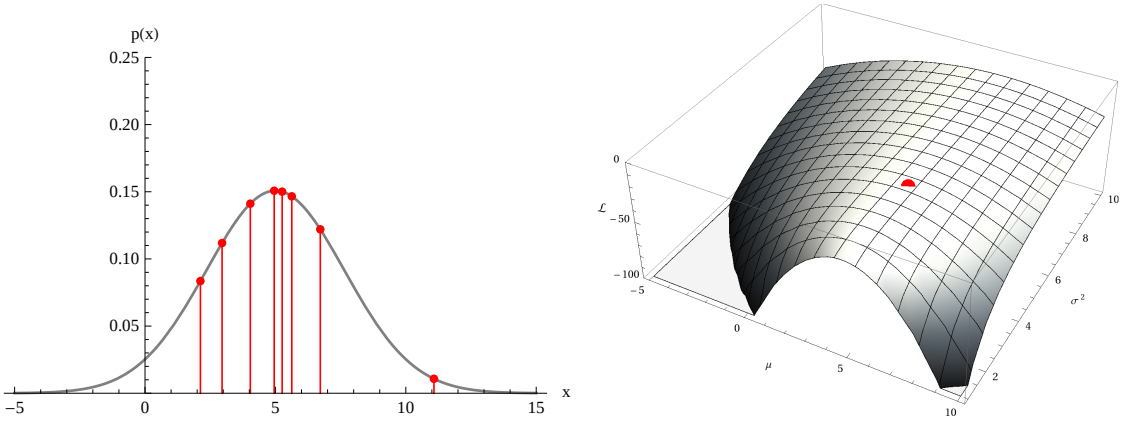
Umjesto maksimiziranja funkcije izglednosti, često je mnogo praktičnije maksimizirati (prirodan) logaritam te funkcije, odnosno funkciju **log-izglednosti** (engl. *log-likelihood*):

$$\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) \;\equiv\; \ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \ln \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta})\,. \tag{2.35}$$

Maksimizacija funkcije log-izglednosti istovjetna je maksimizaciji funkcije izglednosti budući da ove dvije funkcije maksimum poprimaju u istim točkama (funkcija ln je monotona). Maksimizacija ove funkcije je praktičnija jer umjesto umnoška radimo sa zbrojem, a lakše je i baratati s argumentima eksponencijalne funkcije. Na slici 2.3b dan je primjer funkcije log-izglednosti $\ln \mathcal{L}(\mu, \sigma^2|\mathcal{D})$ Gaussove razdiobe.

U slučajevima kada je to moguće i kada je to učinkovito, ovu optimizaciju provodimo analitički (nalaženjem nul-točaka prve derivacije). U mnogim slučajevima međutim analitičko rješenje ili nije moguće ili je računalno prezahtjevno, i tada pribjegavamo iterativnim optimizacijskim metodama.

U nastavku ćemo izvesti procjenitelje najveće izglednosti za nekoliko nama interesantnih teorijskih razdioba. U svim slučajevima koje ćemo razmotriti optimizaciju je moguće provesti analitički.



(a) (b)

Slika 2.3: Maksimizacija izglednosti parametara μ i σ^2 Gaussove razdiobe: (a) Gaussova razdioba $\mathcal{N}(\mu, \sigma^2)$ koja maksimizira vjerojatnost uzorka, (b) funkcija log-izglednosti $\ln \mathcal{L}(\mu, \sigma^2|\mathcal{D})$ (crvenom točkom označen je njezin maksimum).

2.4.1 ML-procjenitelj za Bernoullijevu razdiobu

Bernoullijeva razdioba ima μ kao jedini parametar. Izvedimo njegov procjenitelj najveće izglednosti. Funkcija log-izglednosti je

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

Deriviranjem i izjednačavanjem s nulom dobivamo

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mu} = \frac{1}{\mu} \sum_{i=1}^{N} x^{(i)} - \frac{1}{1-\mu}(N - \sum_{i=1}^{N} x^{(i)}) = 0$$

iz čega kao procjenitelj najveće izglednosti slijedi

$$\hat{\mu}_{\mathrm{ML}} = \frac{1}{N} \sum_{i=1}^{N} x^{(i)}\,. \tag{2.36}$$

Prema tome, najizglednija procjena za μ je **relativna frekvencija** događaja u uzorku, odnosno srednja vrijednost uzorka. Kako je pokazano u primjeru 2.2, očekivanje tog procjenitelja jednako je srednjoj vrijednosti razdiobe, $\mathbb{E}[x]$. Budući da za Bernoullijevu razdiobu vrijedi $\mathbb{E}[x] = \mu$, to slijedi $\mathbb{E}[\hat{\mu}_{\mathrm{ML}}] = \mu$, odnosno $\hat{\mu}_{\mathrm{ML}}$ je nepristran procjenitelj.

Važno je naglasiti da, premda je procjenitelj $\hat{\mu}_{\mathrm{ML}}$ nepristran (a također i valjan), procjena ipak ovisi o uzorku i ona ne mora uvijek biti dobra. Na primjer, za uzorak za koji $x^{(i)} = 1$, najizglednija procjena je $\hat{\mu}_{\mathrm{ML}} = 1$, što dovodi do **prenaučenosti** modela.

2.4.2 ML-procjenitelj za multinomijalnu varijablu

Funkcija log-izglednosti za razdiobu (2.23) je

$$\ln \mathcal{L}(\boldsymbol{\mu}|\mathcal{D}) = \ln \prod_{i=1}^{N} P(\mathbf{x}|\boldsymbol{\mu}) = \ln \prod_{i=1}^{N} \prod_{k=1}^{K} \mu_k^{x_k^{(i)}} = \sum_{k=1}^{K} \sum_{i=1}^{N} x_k^{(i)} \ln \mu_k \,. \tag{2.37}$$

Kako bismo dobili ML-procjenitelj, gornji izraz potrebno je maksimizirati s obzirom na μ_k, kao što smo radili i ranije. Međutim, u ovom slučaju u obzir moramo uzeti ograničenje $\sum_{k=1}^{K} \mu_k = 1$; ukoliko to ne učinimo, nećemo dobiti ispravno rješenje (uvjerite se u to). Optimizaciju s ograničenjem možemo provesti **metodom Lagrangeovih multiplikatora**. Uvodimo novu varijablu, tzv. **Lagrangeov multiplikator** λ, i umjesto maksimizacije izraza (2.37), maksimiziramo odgovarajuću Lagrangeovu funkciju:

$$\sum_{i=1}^{K} \sum_{i=1}^{N} x_k^{(i)} \ln \mu_k + \lambda \Big(\sum_{k=1}^{K} \mu_k - 1 \Big) \,.$$

Deriviranjem po μ_k i izjednačavanjem s nulom dobivamo

$$\mu_k = -\frac{1}{\lambda} \sum_{i=1}^{N} x_k^{(i)} \,. \tag{2.38}$$

Kako bismo izračunali vrijednost multiplikatora λ, dobiveni izraz uvrštavamo u ograničenje $\sum_k \mu_k = 1$ i tako dobivamo:

$$\sum_{k=1}^{K} \mu_k = -\frac{1}{\lambda} \underbrace{\sum_{k=1}^{K} \sum_{i=1}^{N} x_k^{(i)}}_{=N} = 1 \,.$$

Budući da svaka multinomijalna varijabla $\mathbf{x}$ ima jedinicu postavljenu na samo jednoj komponenti, zbroj komponenata svih varijabli jednak je broju primjera N. Vrijedi dakle $\lambda = -N$, pa uvrštavanjem u (2.38) za ML-procjenitelj konačno dobivamo:

$$\hat{\mu}_{k,\mathrm{ML}} = \frac{1}{N} \sum_{i=1}^{N} x_k^{(i)} = \frac{N_k}{N} \tag{2.39}$$

gdje smo s N_k označili broj koliko je puta varijabla u uzorku poprimila vrijednost k. Kao i kod Bernoullijeve varijable, i ovdje smo za najizgledniju procjenu dobili relativnu frekvenciju, što je očekivan rezultat. Alternativno, optimizaciju izraza (2.37) mogli smo provesti tako da smo multinomijalnu varijablu $\mathbf{x}$ tretirali kao K nezavisnih realizacija binarne varijable.

2.4.3 ML-procjenitelji za Gaussovu razdiobu

Izvedimo procjenitelje maksimalne izglednosti za parametre μ i σ^2 normalne razdiobe. Za dani uzorak $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ takav da $x^{(i)} \sim \mathcal{N}(\mu, \sigma^2)$, funkcija log-izglednosti je

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

Postavljanjem $\nabla \ln \mathcal{L}(\mu, \sigma|\mathcal{D}) = 0$ i rješavanjem po μ odnosno σ^2 dobivamo procjenitelje najveće izglednosti

$$\hat{\mu}_{\mathrm{ML}} = \frac{1}{N} \sum_{i=1}^{N} x^{(i)} \tag{2.40}$$

$$\hat{\sigma}^2_{\mathrm{ML}} = \frac{1}{N} \sum_{i=1}^{N} (x^{(i)} - \hat{\mu}_{\mathrm{ML}})^2 \,. \tag{2.41}$$

(Uvjerite se u ove jednakosti.) Primijetite da je procjenitelj $\hat{\sigma}^2_{\mathrm{ML}}$ izražen pomoću procjenitelja $\hat{\mu}_{\mathrm{ML}}$, zato jer nam je prava vrijednost parametra μ nepoznata.

Za procjenitelj srednje vrijednosti $\hat{\mu}_{\mathrm{ML}}$ već smo utvrdili da je to nepristran procjenitelj za koji vrijedi $\mathbb{E}[\hat{\mu}_{\mathrm{ML}}] = \mu$ i $\mathrm{Var}(\hat{\mu}_{\mathrm{ML}}) = \sigma^2/N$ (v. primjer 2.2). Za razliku od procjenitelja $\hat{\mu}_{\mathrm{ML}}$, koji je nepristran, procjenitelj varijance $\hat{\sigma}^2_{\mathrm{ML}}$ nije nepristran (v. primjer 2.3). Razlika dolazi do izražaja kod malenih uzoraka, kada je za procjenu bolje koristiti nepristran procjenitelj $N\hat{\sigma}^2_{\mathrm{ML}}/(N-1)$.

Kao što ovaj primjer ilustrira, sasvim je moguće da postupak najveće izlgednosti rezultira procjeniteljem koji nije nepristran (*najveća izglednost* ne znači nužno *nepristranost*).

2.4.4 ML-procjenitelji za multivarijatnu Gaussovu razdiobu

Izvedimo ML-procjenitelje za parametre $\boldsymbol{\mu}$ i $\boldsymbol{\Sigma}$ multivarijatne Gaussove razdiobe. Log-izglednost jednaka je

$$\begin{aligned} \ln \mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D}) &= \ln \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= -\frac{nN}{2} \ln(2\pi) - \frac{N}{2} |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) \,. \end{aligned} \tag{2.42}$$

Derivacijom po $\boldsymbol{\mu}$ i izjednačavanjem s nulom dobivamo

$$\frac{\partial}{\partial \boldsymbol{\mu}} \ln \mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D}) = \frac{1}{2} \sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) = 0$$

iz čega kao procijenitelj najveće izglednosti slijedi

$$\hat{\boldsymbol{\mu}}_{\mathrm{ML}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)} \,. \tag{2.43}$$

Maksimizacija izraza (2.42) po $\boldsymbol{\Sigma}$ nešto je složenija, no daje očekivan rezultat:

$$\hat{\boldsymbol{\Sigma}}_{\mathrm{ML}} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}_{\mathrm{ML}})(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}_{\mathrm{ML}})^{\mathrm{T}} \,. \tag{2.44}$$

Procjenitelji $\hat{\boldsymbol{\mu}}_{\mathrm{ML}}$ i $\hat{\boldsymbol{\Sigma}}_{\mathrm{ML}}$ analogni su procjeniteljima $\hat{\mu}_{\mathrm{ML}}$ odnosno $\hat{\sigma}^2_{\mathrm{ML}}$ univarijatne Gaussove razdiobe. Također vrijede ista zapažanja što se tiče nepristranosti procjenitelja: procjenitelj $\hat{\boldsymbol{\mu}}_{\mathrm{ML}}$ je nepristran, dok procjenitelj $\hat{\boldsymbol{\Sigma}}_{\mathrm{ML}}$ podcjenuje i na malim uzorcima treba ga korigirati faktorom $N/(N-1)$.

Primjer 2.5 (Procjena parametara multivarijatne razdiobe) Raspolažemo uzorkom $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{8}$ za koji znamo da potječe iz multivarijatne normalne razdiobe:

$$\begin{array}{llll} \mathbf{x}^{(1)} = & (9.59, -0.75, & 0.60) & \mathbf{x}^{(5)} = (2.24, \quad 0.02, -4.72) \\ \mathbf{x}^{(2)} = & (2.30, \quad 0.37, & -2.90) & \mathbf{x}^{(6)} = (6.59, -0.20, -0.69) \\ \mathbf{x}^{(3)} = & (8.87, -0.84, & -0.28) & \mathbf{x}^{(7)} = (3.69, \quad 0.35, -1.84) \\ \mathbf{x}^{(4)} = & (3.99, \quad 1.92, & -0.13) & \mathbf{x}^{(8)} = (3.10, \quad 1.29, -0.59) \end{array}$$

Prema (2.43), ML-procjena vektora srednje vrijednosti je

$$\hat{\boldsymbol{\mu}} = \frac{1}{8}\sum_{i=1}^{8}\mathbf{x}^{(i)} \approx (5.05, 0.27, -1.32)\,.$$

Prema (2.44), ML-procjena kovarijacijske matrice je

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{8}\sum_{i=1}^{8}(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^{\mathrm{T}} \approx \begin{pmatrix} 7.47 & -1.63 & 3.20 \\ -1.63 & 0.79 & -0.01 \\ 3.20 & -0.01 & 2.68 \end{pmatrix}.$$

Kovarijacijska matrica nam govori da je varijanca najveća za prvu, najmanja za drugu varijablu. Prema (2.17), koeficijenti korelacije između varijabli su

$$\begin{aligned} \hat{\rho}_{X_1,X_2} &= \frac{\mathrm{Cov}(X_1,X_2)}{\sigma_{X_1}\sigma_{X_2}} = \frac{-1.63}{\sqrt{7.47 \times 0.79}} \approx -0.670 \\ \hat{\rho}_{X_1,X_3} &= \frac{\mathrm{Cov}(X_1,X_3)}{\sigma_{X_1}\sigma_{X_3}} = \frac{3.20}{\sqrt{7.47 \times 2.68}} \approx 0.715 \\ \hat{\rho}_{X_2,X_3} &= \frac{\mathrm{Cov}(X_2,X_3)}{\sigma_{X_2}\sigma_{X_3}} = \frac{-0.01}{\sqrt{0.79 \times 2.68}} \approx 0.007 \end{aligned}$$

Vidimo da su varijable X_1 i X_2 negativno korelirane, varijable X_1 i X_3 su pozitivno korelirane, dok varijable X_2 i X_3 gotovo da i nisu korelirane. Budući da su ovi koeficijenti računati na temelju uzorka, oni su također procjene i mogu varirati u ovisnosti o uzorku.

2.5 Bayesovski procjenitelj

Procjenitelj najveće izglednost najjednostavniji je način procjene parametara razdiobe. Međutim, njegov je najveći nedostatak sklonost prenaučenosti. Na primjer, ako je varijabla X Bernoullijeva, i ako se u uzorku nikada nije realizirala, najizglednija procjena za $\hat{\mu}_{\mathrm{ML}} = P(X = 1)$ jest 0. U mnogim situacijama ne možemo biti zadovoljni s takvom procjenom jer iskustveno znamo da ona nije realna i da je problem naprosto u tome što s uzorkom "nismo imali sreće". Vidjet ćemo također da procjena koja rezultira ništicom može biti loša za rad klasifikatora (npr. naivnog Bayesovog).

Uobičajen način rješavanja ovog problema jest uporaba **zaglađenih procjena** (engl. *smoothed estimates*). Zaglađena procjena preraspoređuje (zaglađuje) ukupnu masu vjerojatnosti tako da vrijednosti parametara koje bi inače imale vjerojatnost nula dobiju neku malu vjerojatnost (i obrnuto: da vrijednosti koje bi inače imale vjerojatnost 1 dobiju

nešto manju vjerojatnost). Na taj se način smanjuje prenaučenost modela.

2.5.1 Bayesovski i frekventistički pristup

Teorijski okvir zaglađene procjene parametara daje **bayesovska statistika** (engl. *Bayesian statistics*). Bayesovska statistika koristi teoriju vjerojatnosti kao alat za modeliranje nesigurnosti znanja. Prema bayesovskom shvaćanju teorije vjerojatnosti, jednakost $P(X = x) = 0.7$ odgovara izjavi *"vjerujem da $X = x$ s vjerojatnošću* 0.7*"*. Suprotno bayesovskom jest **frekventističko** shvaćanje teorije vjerojatnosti, koje vjerojatnost tumači kao vjerojatnost ishoda kod ponavljanja eksperimenata, odnosno kao relativnu frekvenciju. Prema frekventističkom shvaćanju, jednakost $P(X = x) = 0.7$ odgovara izjavi *"u 70% slučajeva vrijedi $X = x$"*.

U nekim slučajevima frekventističko tumačenje intuitivnije je od bayesovskog, dok je u nekim slučajevim obrnuto. Npr., vjerojatnost da bacajući dvije kocke dobijemo dvije šestice intuitivno je lakše tumačiti frekventistički, budući da je eksperiment moguće ponavljati. S druge strane, vjerojatnost da je neki planet naseljen inteligentnim bićima intuitivno je lakše tumačiti bayesovski, budući da tako izražavamo nesigurnost našeg znanja, a sâm eksperiment uopće nije ponovljiv.

2.5.2 Bayesovski procjenitelj

Osnovna zamisao bayesovskog procjenitelja jest kombinirati apriorno znanje o mogućim vrijednostima parametra θ sa znanjem koje proizlazi iz uzorka. (Kod metode najveće izglednosti tako nešto nije moguće – ondje koristimo samo znanje koje proizlazi iz uzorka.) Ako, prije nego što uopće pogledamo uzorak, imamo neko apriorno znanje o distribuciji parametara, to je znanje vrlo korisno i treba ga iskoristiti. To je osobito slučaj ako je uzorak malen, jer tada ML-procjenitelj lako može prenaučiti model.

Naravno, točna vrijednost parametra θ nije nam poznata (inače uopće ne bismo trebali raditi procjenu na temelju uzorka), ali imamo neku predodžbu o mogućim vrijednostima tog parametra. To znanje modeliramo kao "nesigurnost", na način da parametar θ tretiramo kao slučajnu varijablu i definiramo njezinu distribuciju, $p(\theta)$. Npr., možemo reći da je vrlo vjerojatno da $\mu = 0.5$, ali da je manje vjerojatno da $\mu = 0$, što možemo modelirati primjerice kao $p(\mu = 0.5) = 0.8$ i $p(\mu = 0) = 0.1$. Primijetite da se ovdje zapravo radi o distribuciji nad varijablom koja je parametar druge distribucije, tj. možemo govoriti o *distribuciji distribucije*.

Sada možemo upotrijebiti Bayesovo pravilo kako bismo kombinirali apriornu razdiobu $p(\theta)$ s izglednošću $p(\mathcal{D}|\theta)$, koju smo dobili na temelju uzorka (skupa primjera) $\mathcal{D}$:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta')p(\theta')\mathrm{d}\theta'} \tag{2.45}$$

čime smo dobili aposteriornu vjerojatnost za parametar θ za dani uzorak $\mathcal{D}$. **Bayesovski procjenitelj** definiran je kao očekivanje vrijednosti od θ s obzirom na distribuciju $p(\theta|\mathcal{D})$:

$$\hat{\theta}_{\mathrm{Bayes}} = \mathbb{E}[\theta|\mathcal{D}] = \int \theta\, p(\theta|\mathcal{D})\mathrm{d}\theta\,. \tag{2.46}$$

Ovaj integral može biti teško izračunljiv, ovisno o obliku apriorne distribucije $p(\theta)$. Također, ovisno o izboru distribucije $p(\theta)$, umnožak $p(\mathcal{D}|\theta)p(\theta)$ može poprimiti različite oblike. Najbolji oblik za $p(\theta)$ – za koji je integral analitički izračunljiv – jest onaj koji je istog oblika

kao aposteriorna gustoća $p(\theta|\mathcal{D})$. Ako je to ispunjeno, par distribucija $p(\theta)$ i $p(\theta|\mathcal{D})$ zovemo **konjugatnim distribucijama**. Nadalje, distribuciju $p(\theta)$ zovemo **konjugatnom apriornom distribucijom** (engl. *conjugate prior*) za izglednost $p(\mathcal{D}|\theta)$.

Sve razdiobe koje smo dosada razmatrali pripadaju tzv. **eksponencijalnoj familiji distribucija** i svaka od njih ima konjugatnu apriornu distribuciju. Konkretno, konjugatna apriorna distribucija za Gaussovu distribuciju je i sama Gaussova distribucija, dok je za multinomijalnu distribuciju konjugatna apriorna distibucija **Dirichletova distribucija**.

2.5.3 Laplaceovo zaglađivanje

U nastavku ćemo se usredotočiti na bayesovski procjenitelj za multinomijalnu varijablu, poznat pod nazivom **Laplaceovo zaglađivanje** (engl. *Laplace smoothing*). Parametar multinomijalne razdiobe, definirane s (2.23), jest $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$. Za apriornu razdiobu parametra $p(\boldsymbol{\mu})$ koristimo Dirichletovu razdiobu, koja je konjugatna apriorna distribucija za multinomijalnu distribuciju. Dirichletova razdioba

$$\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \mathrm{Dir}(\mu_1, \ldots, \mu_K|\alpha_1, \ldots, \alpha_k)$$

ima vektor parametara $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ i također je zvonolika (konkretan izraz za Dirichletovu distribuciju ovdje nas ne zanima). Parametri $\boldsymbol{\alpha}$ određuju oblik Dirichletove razdiobe i zapravo su hiperparametri našeg modela.

Uvrštavanje Dirichletove razdiobe u (2.45) i izračunavanjem očekivanja (2.46), za zaglađeni procjenitelj parametra μ_k naposlijetku dobivamo:

$$\hat{\mu}_{k,\mathrm{Bayes}} = \frac{N_k + \alpha_k}{N + \sum_{i=1}^{K} \alpha_i}\,.$$

Usporedimo ovaj izraz s izrazom (2.39) za ML-procjenitelj. Vrijednost N_k/N je relativna frekvencija koju, kao i kod ML-procjenitelja, izračunavamo na temelju uzorka. Međutim, bayesovski procjenitelj dodatno kombinira relativnu frekvenciju s apriornim znanjem. Interpretacija bayesovskog procjenitelja je sljedeća: prije nego što smo opazili uzorak $\mathcal{D}$, "virtualno" smo opazili uzorak veličine $\sum_{i=1}^{K} \alpha_k$, u kojemu se slučajna varijabla realizirala s vrijednošću k ukupno α_k puta. Za virtualni uzorak veličine 0, bayesovski procjenitelj degradira na ML-procjenitelj.

Možemo pojednostaviti Dirichletovu apriornu distribuciju na način da pretpostavimo $\alpha_i = \lambda$ za svaki $i = 1, \ldots, K$, tj. da koristimo apriornu distribuciju $\mathrm{Dir}(\mu_1, \ldots, \mu_K|\lambda, \ldots, \lambda)$. Tako dobivamo **Laplaceov (Lidstonov) procjenitelj**:

$$\hat{\mu}_{k,\mathrm{Bayes}} = \frac{N_k + \lambda}{N + K\lambda}\,.$$

Odabirom različitih vrijednosti za λ dobivamo različite procjenitelje. Najčešće se uzima:

- Laplaceovo pravilo ili *add-one rule* ($\lambda = 1$): $\hat{\mu}_{k,\mathrm{Bayes}} = \frac{N_k+1}{N+K}$,
- Jeffreys-Perksovo pravilo ($\lambda = 1/2$): $\hat{\mu}_{k,\mathrm{Bayes}} = \frac{N_k+1/2}{N+K/2}$,
- Schurmann-Grassbergerovo pravilo ($\lambda = 1/K$): $\hat{\mu}_{k,\mathrm{Bayes}} = \frac{N_k+1/K}{N+1}$.

Primjer 2.6 (Laplaceov procjenitelj) Neka je X multinomijalna varijabla s $K = 5$ mogućih stanja, $x \in \{0, 1, 2, 3, 4\}$. Iz ove razdiobe dobiven je uzorak veličine $N = 16$:

$$\mathbf{X} = (0, 1, 2, 0, 4, 1, 0, 0, 2, 2, 1, 4, 2, 1, 0, 2)\,.$$

Prema (2.39), ML-procjena parametra $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \mu_4, \mu_5)$ je

$$\hat{\boldsymbol{\mu}}_{\mathrm{ML}} \approx (0.3125, 0.25, 0.3125, 0.0, 0.125)\,.$$

Dobivena procjena je prenaučena: vjerojatnost ishoda $X = 3$ jednaka je nuli zbog toga što se u uzorku varijabla X nikada nije realizirala s vrijednošću 3. Zaglađeni procjenitelji daju drugačije procjene:

Laplace:	$\hat{\boldsymbol{\mu}} \approx (0.286, 0.238, 0.286, 0.048, 0.142)$
Jeffreys-Perks:	$\hat{\boldsymbol{\mu}} \approx (0.298, 0.243, 0.298, 0.027, 0.135)$
Schurmann-Grassberger:	$\hat{\boldsymbol{\mu}} \approx (0.306, 0.247, 0.306, 0.012, 0.129)$

Iz primjera je vidljivo da je Schurmann-Grassbergerov procjenitelj "najkonzervativniji" u smislu da događaju koji se u uzorku nije realizirao dodjeljuje najmanje vjerojatnosne mase.

Poglavlje 3

Bayesov klasifikator

Bayesov klasifikator je vjerojatno najpoznatiji klasifikacijski model i tipičan predstavnik generativnih modela. Klasifikacija primjera ostvaruje se pomoću Bayesovog pravila, koji nam za svaku klasu daje vjerojatnost da primjer pripada toj klasi. Bayesov klasifikator je parametarski model, što znači da pretpostavlja da se podatci pokoravaju nekoj teorijskoj vjerojatnosnoj razdiobi. Parametri razdiobe su nepoznati i potrebno ih je naučiti (procijeniti) na temelju podataka.

3.1 Bayesovo pravilo

Kod Bayesovog klasifikatora, klasifikacija primjera $\mathbf{x}$ temelji se na izračunu **aposteriorne vjerojatnosti** $P(Y = \mathcal{C}_j|X = \mathbf{x})$, tj. vjerojatnosti da primjer $\mathbf{x}$ pripada klasi $\mathcal{C}_j$. Tu vjerojatnost izračunavamo posredno, na temelju zajedničke gustoće $p(\mathbf{x}, \mathcal{C}_j)$, primjenom Bayesovog pravila:

$$P(\mathcal{C}_j|\mathbf{x}) = \frac{p(\mathbf{x}, \mathcal{C}_j)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathcal{C}_j)P(\mathcal{C}_j)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathcal{C}_j)P(\mathcal{C}_j)}{\sum_{k=1}^{K} p(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k)}\,. \tag{3.1}$$

Marginalnu vjerojatnost $P(\mathcal{C}_j)$ nazivamo **apriorna vjerojatnost klase** (engl. *class prior*), a uvjetnu gustoću $p(\mathbf{x}|\mathcal{C}_j)$ nazivamo **klasom uvjetovana gustoća** (engl. *class conditional density*) ili, općenitije, **izglednost klase** (engl. *class likelihood*). Ako je $\mathbf{x}$ diskretna varijabla, umjesto gustoća $p(\mathbf{x}|\mathcal{C}_j)$ i $p(\mathbf{x})$ koristimo odgovarajuće vjerojatnosti.

Svi izrazi koji se pojavljuju u (3.1) mogu se dobiti marginalizacijom odnosno normalizacijom zajedničke gustoće $p(\mathbf{x}, \mathcal{C}_j)$, koja uvijek sadržava potpunu informaciju o podatcima. Međutim, faktorizacija zajedničke gustoće u izglednost i apriornu vjerojatnost bitno pojednostavljuje modeliranje jer omogućava da zasebno, pomoću odabrane parametarske razdiobe, modeliramo izglednost svake klase (tj. razdiobu primjera unutar svake klase). Na primjer, za kontinuiranu varijablu $\mathbf{x}$ izglednost se uobičajeno modelira Gaussovom razdiobom.

Optimalna klasifikacijska odluka jest ona koja maksimizira aposteriornu vjerojatnost $P(\mathcal{C}_j|\mathbf{x})$. Drugim riječima, primjer $\mathbf{x}$ treba klasificirati u onu klasu $\mathcal{C}_j$ za koju je $P(\mathcal{C}_j|x)$ najveći. Takvu hipotezu nazivamo **maksimalna aposteriorna hipoteza** (MAP):

$$h(\mathbf{x}) = \operatorname*{argmax}_{\mathcal{C}_k} p(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k)\,. \tag{3.2}$$

Primijetite da je dovoljno maksimizirati samo brojnik izraza (3.1), budući da je marginalna gustoća $p(\mathbf{x})$ za sve klase $\mathcal{C}_j$ jednaka i služi samo normalizaciji na jedinični interval. Ako želimo vjerojatnosnu interpretaciju hipoteze, možemo definirati zasebnu hipotezu h_j za svaku klasu $\mathcal{C}_j$:

$$h_j(\mathbf{x}) = P(\mathcal{C}_j|\mathbf{x})\,.$$

Zanemarimo li nazivnik $p(\mathbf{x})$, hipotezu možemo još jednostavnije definirati kao

$$h_j(\mathbf{x}) = p(\mathbf{x}|\mathcal{C}_j)P(\mathcal{C}_j)$$

Ovime gubimo vjerojatnosnu interpretaciju jer vrijednost nije normalizirana na jedinični interval, no vrijednost $h_j(\mathbf{x})$ može se tumačiti kao pouzdanosti klasifikacije primjera $\mathbf{x}$ u klasu $\mathcal{C}_j$. U oba ova slučaja govorimo o **klasifikatoru s ocjenom pouzdanosti** (engl. *confidence-rated classifier*).

Primjer 3.1 (Bayesova klasifikacija) Razmotrimo problem klasifikacije u tri klase: $\mathcal{C}_1$, $\mathcal{C}_2$ i $\mathcal{C}_3$. Neka su apriorne vjerojatnosti tih klasa $P(\mathcal{C}_1) = P(\mathcal{C}_2) = 0.3$ i $P(\mathcal{C}_3) = 0.4$. Poznato je da za primjer $\mathbf{x}$ izglednosti iznose $p(\mathbf{x}|\mathcal{C}_1) = 0.9$ i $p(\mathbf{x}|\mathcal{C}_2) = p(\mathbf{x}|\mathcal{C}_3) = 0.4$. Izračunajmo maksimalnu aposteriornu hipotezu za svaku od klasa. U brojniku izraza (3.1) imamo

$$\begin{aligned} p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1) &= 0.9 \times 0.3 = 0.27 \\ p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2) &= 0.4 \times 0.3 = 0.12 \\ p(\mathbf{x}|\mathcal{C}_3)P(\mathcal{C}_3) &= 0.4 \times 0.4 = 0.16 \end{aligned}$$

Apriorna gustoća vjerojatnosti primjera $\mathbf{x}$ jest $p(\mathbf{x}) = \sum_{k=1}^{3} p(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k) = 0.55$. Za aposteriorne vjerojatnosti $P(\mathcal{C}_2|\mathbf{x})$ dobivamo

$$P(\mathcal{C}_1|\mathbf{x}) = \frac{0.27}{0.55} = 0.49 \qquad P(\mathcal{C}_2|\mathbf{x}) = \frac{0.12}{0.55} = 0.22 \qquad P(\mathcal{C}_3|\mathbf{x}) = \frac{0.16}{0.55} = 0.29$$

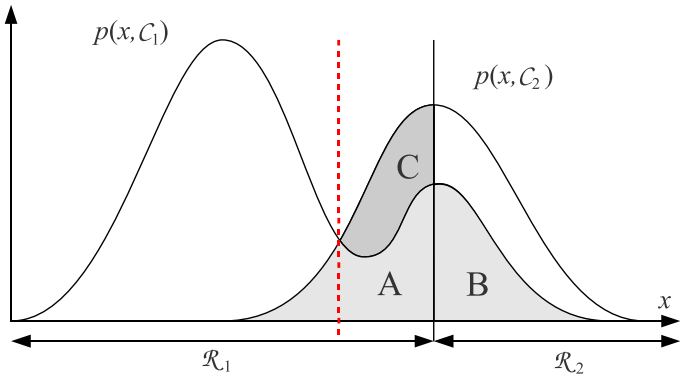
U ovom slučaju aposteriorna vjerojatnost najveća je za klasu $\mathcal{C}_1$, pa je maksimalna aposteriorna hipoteza $h(\mathbf{x}) = \mathcal{C}_1$.

3.1.1 Minimizacija pogreške klasifikacije

Premda je intuitivno jasno da je optimalna klasifikacijska odluka ona koja maksimizira aposteriornu vjerojatnost $P(\mathcal{C}_j|\mathbf{x})$, uvjerimo se da je to doista tako. Pretpostavimo da je naš cilj minimizirati broj pogrešaka klasifikacije. Ograničimo se na slučaj dvije klase. Pogreška klasifikacije nastupa ako se primjer $\mathbf{x} \in \mathcal{C}_2$ klasificira u klasu $\mathcal{C}_1$, ili obrnuto. Označimo sa $\mathcal{R}_1 \subseteq \mathcal{X}$ primjere koje naš klasifikator klasificira kao $\mathcal{C}_1$, tj. $\mathcal{R}_1 = \{\mathbf{x} \in \mathcal{X} \mid h_1(\mathbf{x}) = 1\}$, a sa $\mathcal{R}_2 = \mathcal{X}\backslash\mathcal{R}_1$ primjere koje naš klasifikator klasificira kao $\mathcal{C}_2$. Vjerojatnost pogreške je

$$P(\text{pogreška}) = p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) = \int_{\mathbf{x}\in\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\,\mathrm{d}\mathbf{x} + \int_{\mathbf{x}\in\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\,\mathrm{d}\mathbf{x}\,.$$

Postavlja se pitanje: kako odabrati područja $\mathcal{R}_1$ i $\mathcal{R}_2$, odnosno gdje postaviti granicu između klasa, a da vjerojatnost pogrešne klasifikacije bude najmanja? Očito je da je



Slika 3.1: Ilustracija zajedničkih gustoća vjerojatnosti $p(x, \mathcal{C}_1)$ i $p(x, \mathcal{C}_2)$ za prostor primjera $\mathcal{X} = \mathbb{R}$. Granice između područja odluke $\mathcal{R}_1$ i $\mathcal{R}_2$ predstavljena je okomitom linijom. Vjerojatnost pogreške klasifikacije jednaka je površini A+B+C. Površina A+B=konst., dok je površinu C moguće ukloniti ako se kao granica između klasa izabere iscrtkana linija, tj. vrijednost x za koju $p(x, \mathcal{C}_1) = p(x, \mathcal{C}_2)$. To je istovjetno odabiru klase za koju je aposteriorna vjerojatnost $P(\mathcal{C}_k|\mathbf{x})$ maksimalna.

granicu potrebno postaviti tako da se svaki primjer $\mathbf{x}$ klasificira u onu klasu za koju je vrijednost ispod integrala manja, jer time za taj primjer smanjujemo vjerojatnost pogreške. To pak znači da, ako je $p(\mathbf{x}, \mathcal{C}_1) > p(\mathbf{x}, \mathcal{C}_2)$, tada primjer $\mathbf{x}$ trebamo klasificirati u klasu $\mathcal{C}_1$ (tj. $\mathcal{R}_1$ treba namjestiti tako da je $\mathbf{x} \in \mathcal{R}_1$), jer je tada vjerojatnost pogreške manja i jednaka je $p(\mathbf{x}, \mathcal{C}_2)$. Budući da vrijedi $p(\mathbf{x}, \mathcal{C}_k) = P(\mathcal{C}_k|\mathbf{x})p(\mathbf{x})$, a da je faktor $p(\mathbf{x})$ zajednički obama pribrojnicima, to slijedi da je vjerojatnost pogreške minimalna ako se svaki primjer $\mathbf{x}$ klasificira u klasu za koju je aposteriorna vjerojatnost $P(\mathcal{C}_k|\mathbf{x})$ najveća. Opisana situacija prikazana je slikom 3.1 za jednodimenzijski prostor primjera $\mathcal{X} = \mathbb{R}$.

3.1.2 *Minimizacija rizika

U gornjim razmatranjima pretpostavili smo da je naš cilj minimizirati broj pogrešnih klasifikacija. Situacija je drugačija kada gubitci nisu jednoliki, odnosno kada matrica gubitka (L_{kj}) nije nula-jedan. Tada je potrebno minimizirati očekivanu vrijednost funkcije gubitka L. Očekivana vrijednost funkcije gubitka L naziva se **funkcija rizika** i definira kao

$$\mathbb{E}[L] = \sum_{k=1}^{K} \sum_{j=1}^{K} \int_{\mathbf{x} \in \mathcal{R}_j} L_{kj}\, p(\mathbf{x}, \mathcal{C}_k)\, \mathrm{d}\mathbf{x} \tag{3.3}$$

gdje je L_{kj} gubitak uslijed pogrešne klasifikacije primjera iz klase $\mathcal{C}_k$ u klasu $\mathcal{C}_j$. Integracija se provodi po regiji $\mathcal{R}_j \subseteq \mathcal{X}$, koja obuhvaća sve primjere koje klasifikator klasificira kao $\mathcal{C}_j$, tj. $\mathcal{R}_j = \{\mathbf{x} \in \mathcal{X} \mid h_j(\mathbf{x}) = 1\}$.

Primjer 3.2 (Očekivanje gubitka) Za neki binaran klasifikacijski problem dana je asimetrična matrica gubitka

$$L = \begin{pmatrix} 0 & 10 \\ 1 & 0 \end{pmatrix}.$$

Dan je (diskretan) prostor primjera $\mathcal{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}\}$. Pomoću generativnog probabilističkog modela izračunate su izglednosti klasa

$$P(\mathbf{x}^{(1)}|\mathcal{C}_1) = 0.75 \qquad P(\mathbf{x}^{(2)}|\mathcal{C}_1) = 0.1 \qquad P(\mathbf{x}^{(3)}|\mathcal{C}_1) = 0.15$$
$$P(\mathbf{x}^{(1)}|\mathcal{C}_2) = 0.25 \qquad P(\mathbf{x}^{(2)}|\mathcal{C}_2) = 0.55 \qquad P(\mathbf{x}^{(3)}|\mathcal{C}_2) = 0.2$$

Apriorne vjerojatnosti klasa neka su $P(\mathcal{C}_1) = 0.8$ i $P(\mathcal{C}_2) = 0.2$. Zajedničke vjerojatnosti $P(\mathbf{x}, \mathcal{C}_j) = P(\mathbf{x}|\mathcal{C}_j)P(\mathcal{C}_j)$ su

$$P(\mathbf{x}^{(1)}, \mathcal{C}_1) = 0.6 \qquad P(\mathbf{x}^{(2)}, \mathcal{C}_1) = 0.08 \qquad P(\mathbf{x}^{(3)}, \mathcal{C}_1) = 0.12$$
$$P(\mathbf{x}^{(1)}, \mathcal{C}_2) = 0.05 \qquad P(\mathbf{x}^{(2)}, \mathcal{C}_2) = 0.11 \qquad P(\mathbf{x}^{(3)}, \mathcal{C}_2) = 0.04$$

iz čega za aposteriorne vjerojatnosti dobivamo

$$P(\mathcal{C}_1|\mathbf{x}^{(1)}) = 0.92 \qquad P(\mathcal{C}_1|\mathbf{x}^{(2)}) = 0.42 \qquad P(\mathcal{C}_1|\mathbf{x}^{(3)}) = 0.75$$
$$P(\mathcal{C}_2|\mathbf{x}^{(1)}) = 0.08 \qquad P(\mathcal{C}_2|\mathbf{x}^{(2)}) = 0.58 \qquad P(\mathcal{C}_2|\mathbf{x}^{(3)}) = 0.25$$

Pretpostavimo da se sva tri primjera klasificiraju tako da se smanji broj pogrešaka klasifikacije, tj. svaki primjer klasificira se u aposteriorno najvjerojatniju klasu: primjeri $\mathbf{x}^{(1)}$ i $\mathbf{x}^{(3)}$ u klasu $\mathcal{C}_1$, a primjer $\mathbf{x}^{(2)}$ u klasu $\mathcal{C}_2$. Tada $\mathcal{R}_1 = \{\mathbf{x}^{(1)}, \mathbf{x}^{(3)}\}$ i $\mathcal{R}_2 = \{\mathbf{x}^{(2)}\}$. Očekivani gubitak je

$$\begin{aligned}
\mathbb{E}[L] &= \sum_{k=1}^{2}\sum_{j=1}^{2}\sum_{\mathbf{x}\in\mathcal{R}_j} L_{kj}P(\mathbf{x}, \mathcal{C}_k) = \sum_{k=1}^{2}\Big(\sum_{\mathbf{x}\in\mathcal{R}_1} L_{k,1}P(\mathbf{x}, \mathcal{C}_k) + \sum_{\mathbf{x}\in\mathcal{R}_2} L_{k,2}P(\mathbf{x}, \mathcal{C}_k)\Big) \\
&= L_{21}\sum_{\mathbf{x}\in\mathcal{R}_1} P(\mathbf{x}, \mathcal{C}_2) + L_{12}\sum_{\mathbf{x}\in\mathcal{R}_2} P(\mathbf{x}, \mathcal{C}_1) \\
&= L_{21}P(\mathbf{x}^{(1)}, \mathcal{C}_2) + L_{21}P(\mathbf{x}^{(3)}, \mathcal{C}_2) + L_{12}P(\mathbf{x}^{(2)}, \mathcal{C}_1) \\
&= 1 \times 0.05 + 1 \times 0.04 + 10 \times 0.08 = 0.89
\end{aligned}$$

U ovom slučaju očekivani gubitak nije minimalan jer klasifikacija primjera $\mathbf{x}^{(2)}$ u smislu minimizacije rizika nije optimalna. Ako bismo primjer $\mathbf{x}^{(2)}$ klasificirali u klasu $\mathcal{C}_1$, očekivani gubitak bio bi optimalan i iznosio bi

$$\mathbb{E}[L] = 1 \times 0.05 + 1 \times 0.04 + 1 \times 0.11 = 0.2\,.$$

Razlog tome jest što je, sukladno zadanoj matrici gubitka L, gubitak uslijed pogrešne klasifikacije u klasu $\mathcal{C}_2$ deset puta veći od gubitka uslijed pogrešne klasifikacije u klasu $\mathcal{C}_1$.

Želimo li dakle minimizirati očekivani gubitak $\mathbb{E}[L]$, svaki primjer $\mathbf{x}$ treba klasificirati u klasu $\mathcal{C}_j$ za koju je vrijednost $\sum_k L_{kj}\, p(\mathbf{x}, \mathcal{C}_k)$ minimalna. Budući da vrijedi $p(\mathbf{x}, \mathcal{C}_k) = P(\mathcal{C}_k|\mathbf{x})p(\mathbf{x})$, a da je faktor $p(\mathbf{x})$ zajednički svim klasama, optimalna klasifikacijska odluka jest ona koja minimizira izraz

$$R(\mathcal{C}_j|\mathbf{x}) = \sum_{k=1}^{K} L_{kj}P(\mathcal{C}_k|\mathbf{x}) \tag{3.4}$$

i to je **očekivani rizik** pri klasifikaciji primjera $\mathbf{x}$ u klasu $\mathcal{C}_j$. Ako dakle želimo optimalnu klasifikaciju u smislu minimizacije rizika, funkciju hipoteze možemo definirati kao

$$h(\mathbf{x}) = \operatorname*{argmin}_{\mathcal{C}_k} R(\mathcal{C}_k|\mathbf{x})\,. \tag{3.5}$$

ili, ako želimo klasifikaciju s ocjenom pouzdanosti, kao

$$h_j(\mathbf{x}) = -R(\mathcal{C}_j|\mathbf{x})\,. \tag{3.6}$$

Primjer 3.3 (Minimizacija rizika) Za neki višeklasni klasifikacijski problem dana je matrica gubitka

$$L = \begin{pmatrix} 0 & 1 & 5 \\ 1 & 0 & 5 \\ 10 & 100 & 0 \end{pmatrix}.$$

Ako su za neki primjer $\mathbf{x}$ temeljem probabilističkog modela izračunate aposteriorne vjerojatnosti $P(\mathcal{C}_1|\mathbf{x}) = 0.25$, $P(\mathcal{C}_2|\mathbf{x}) = 0.6$, $P(\mathcal{C}_3|\mathbf{x}) = 0.15$, odredimo u koju klasu $\mathcal{C}_j$ treba klasificirati primjer $\mathbf{x}$, a da rizik pogrešne klasifikacija bude minimalan. Trebamo minimizirati vrijednost (3.4) za $j \in \{1, 2, 3\}$:

$$\begin{aligned}
(j = 1): &\qquad R(\mathcal{C}_1|\mathbf{x}) = \textstyle\sum_k L_{k,1} P(\mathcal{C}_k|\mathbf{x}) = 1 \times 0.6 + 10 \times 0.15 = 2.10 \\
(j = 2): &\qquad R(\mathcal{C}_2|\mathbf{x}) = \textstyle\sum_k L_{k,2} P(\mathcal{C}_k|\mathbf{x}) = 1 \times 0.25 + 100 \times 0.15 = 15.25 \\
(j = 3): &\qquad R(\mathcal{C}_3|\mathbf{x}) = \textstyle\sum_k L_{k,3} P(\mathcal{C}_k|\mathbf{x}) = 5 \times 0.25 + 5 \times 0.6 = 4.25
\end{aligned}$$

Rizik je najmanji ako se primjer $\mathbf{x}$ klasificira u klasu $\mathcal{C}_1$. Ako međutim u obzir ne bismo uzimali rizik, ili ako bi gubitak bio jednolik (matrica nula-jedan), primjer $\mathbf{x}$ klasificirali bismo u klasu $\mathcal{C}_2$, budući da je aposteriorna vjerojatnost $P(\mathcal{C}_2|\mathbf{x})$ najveća.

3.1.3 *Kompenzacija neujednačene zastupljenosti klasa

U slučajevima kada broj pozitivnih primjera izrazito nadmašuje broj negativnih primjera (ili obrnuto), problematično je naučiti model koji dobro generalizira. Problem se kod Bayesovog klasifikatora može riješiti tako da se skup za učenje najprije umjetno uravnoteži, a zatim se, nakon učenja, aposteriorne vjerojatnosti $P(\mathcal{C}_j|\mathbf{x})$ skaliraju i normaliziraju uzevši u obzir stvarnu zastupljenost klasa.

Primjer 3.4 (Kompenzacija neujednačenosti klasa) Bayesov klasifikator koristimo za detekciju određene vrste raka na temelju medicinskih nalaza pacijenata. Neka je $\mathcal{C}$ klasa primjera u kojima je detektiran rak. Pretpostavimo da se dotična vrsta raka u slučajnom uzorku pacijenata pojavljuje u 1/1000 slučajeva, pa je najizglednija procjena $P(\mathcal{C}) = 0.001$. Ako bi se takav slučajan uzorak koristio kao skup za učenje, on bi bio izrazito neuravnotežen. U namjeri da to kompenziramo, pripremili smo umjetno uravnotežen skup za učenje u kojemu je od 1000 primjera njih 400 pozitivnih. Na tako uravnoteženom skupu vrijedi $P'(\mathcal{C}) = 0.4$.

Pretpostavimo da, nakon učenja na uravnoteženom skupu, model za neki primjer $\mathbf{x}$ izračunava aposteriornu vjerojatnost $P'(\mathcal{C}|\mathbf{x}) = 0.7$. Onda je stvarna aposteriorna vjerojatnost, ispravljena s obzirom na pravu zastupljenost klase $\mathcal{C}$, sljedeća:

$$P(\mathcal{C}|\mathbf{x}) = P'(\mathcal{C}|\mathbf{x}) \times \frac{P(\mathcal{C})}{P'(\mathcal{C})} = 0.7 \times \frac{0.001}{0.4} = 0.00175\,.$$

Uz pretpostavku simetričnog gubitka (matrice gubitka nula-jedan), primjer $\mathbf{x}$ ne bismo klasificirali u klasu $\mathcal{C}$.

3.2 Naivan Bayesov klasifikator

Razmotrimo sada Bayesov klasifikacijski model za slučaj diskretnih ulaznih varijabli. Pretpostavimo da raspolažemo skupom primjera za učenje $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$ te da je problem višeklasan, $y^{(i)} \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$. Naš model je

$$h(x_1, \ldots, x_n) = \operatorname*{argmax}_j P(x_1, \ldots, x_n|\mathcal{C}_j) P(y = \mathcal{C}_j)\,. \tag{3.7}$$

Trebamo dakle procjeniti parametre diskretnih razdioba $P(x_1, \ldots, x_n|\mathcal{C}_j)$ i $P(\mathcal{C}_j)$. Varijabla y je diskretna multinomijalna varijabla, pa za najizgledniju procjenu apriornih vjerojatnosti $P(\mathcal{C}_j)$ imamo

$$\hat{P}(\mathcal{C}_j) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = \mathcal{C}_j\} = \frac{N_j}{N} \tag{3.8}$$

tj. relativan udio primjera koji su označeni klasom $\mathcal{C}_j$. Ukupan broj ovakvih parametara je $K - 1$, gdje je K broj klasa (za K-tu klasu vjerojatnost je određena zbog ograničenja $\sum_j P(\mathcal{C}_j) = 1$).

Na isti način mogli bismo pokušati procjeniti vjerojatnost $P(x_1, \ldots, x_n|\mathcal{C}_j)$, tako da varijable $x_1, \ldots, x_n$ tretiramo zajednički, tj. da vektor $\mathbf{x} = (x_1, \ldots, x_n)$ tretiramo kao jednu multinomijalnu varijablu. Međutim, to rješenje u praksi ne funkcionira. Problem je što broj mogućih stanja varijable $\mathbf{x}$ raste eksponencijalno s dimenzijom n, a to znači da i broj parametara modela raste eksponencijalno. Npr., binaran vektor $\mathbf{x}$ može poprimiti 2^n različitih vrijednosti. Za svaki takav vektor trebat će procijeniti vjerojatnost pripadanja jednoj od K klasa, što znači da ukupno treba procijeniti $(2^n - 1)K$ parametara. Zbog velikog broja parametara, model će imati vrlo visoku varijancu i bit će sklon prenaučenosti. Zapravo, takav model samo pohranjuje vjerojatnosti viđenih primjera, pa savršeno klasificira primjere iz skupa zaučenje, dok svim neviđenim primjerima dodijeljuje vjerojatnost nula. Kako bi model mogao generalizirati, nužno je uvesti neke pretpostavke.

Prije no što uvedemo ikakvu pretpostavku, primijetimo da vjerojatnost $P(x_1, \ldots, x_n|\mathcal{C}_j)$ možemo faktorizirati primjenom pravila lanca. Općenito, **pravilo lanca** (engl. *chain rule*) glasi

$$\begin{aligned} P(x_1, \ldots, x_n) &= P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \cdots P(x_n|x_1, \ldots, x_{n-1}) \\ &= \prod_{k=1}^{n} P(x_k|x_1, \ldots, x_{k-1}). \end{aligned} \tag{3.9}$$

Dodavanjem uvjetne varijable $\mathcal{C}_j$, dobivamo:

$$P(x_1, \ldots, x_n|\mathcal{C}_j) = \prod_{k=1}^{n} P(x_k|x_1, \ldots, x_{k-1}, \mathcal{C}_j). \tag{3.10}$$

Pretpostavka koju sada možemo uvesti jest da su varijable međusobno **uvjetno nezavisne** za zadanu klasu, odnosno da vrijedi

$$P(x_i|x_j, \mathcal{C}) = P(x_i|\mathcal{C})\,. \tag{3.11}$$

Višestrukom primjenom ove jednakosti izraz (3.10) faktorizira se na

$$P(x_1, \ldots, x_n|\mathcal{C}_j) = \prod_{k=1}^{n} P(x_k|x_1, \ldots, x_{k-1}, \mathcal{C}_j) \overset{(3.11)}{=} \prod_{k=1}^{n} P(x_k|\mathcal{C}_j) \tag{3.12}$$

što daje model

$$h(x_1, \ldots, x_n) = \operatorname*{argmax}_j P(\mathcal{C}_j) \prod_{k=1}^{n} P(x_k|\mathcal{C}_j)\,. \tag{3.13}$$

Ovaj model nazivamo **naivan Bayesov klasifikator** (engl. *naïve Bayes classifer*).[1] Nazivamo ga naivnim jer pretpostavka o uvjetnoj nezavisnosti u praksi uglavnom ne vrijedi. Primjerice, u kontekstu klasifikacije dokumenata, ova pretpostavka kaže da, ako nam je poznato da dokument pripada klasi "Sport", vjerojatnost pojavljivanja riječi "nogomet" jednaka je neovisno o tome pojavljuje li se u istome tekstu riječ "lopta". Pretpostavka je sasvim sigurno pogrešna, no ipak se pokazuje da naivan Bayesov klasifikator u praksi vrlo dobro funkcionira.

Vjerojatnosti $P(x_k|\mathcal{C}_j)$ možemo jednostavno procjeniti metodom najveće izglednosti:

$$\hat{P}(x_k|\mathcal{C}_j) = \frac{\sum_{i=1}^{N} \mathbf{1}\{x_k^{(i)} = x_k \wedge y^{(i)} = \mathcal{C}_j\}}{\sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = \mathcal{C}_j\}} = \frac{N_{kj}}{N_j} \tag{3.14}$$

ili Laplaceovim procjeniteljem:

$$\hat{P}(x_k|\mathcal{C}_j) = \frac{\sum_{i=1}^{N} \mathbf{1}\{x_k^{(i)} = x_k \wedge y^{(i)} = \mathcal{C}_j\} + \lambda}{\sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = \mathcal{C}_j\} + K_k\lambda} = \frac{N_{kj} + \lambda}{N_j + K_k\lambda} \tag{3.15}$$

gdje je K_k broj mogućih vrijednosti značajke x_k.

Ukupan broj ovakvih vjerojatnosti koje moramo procijeniti je $\sum_{k=1}^{n}(K_k - 1)K$. Ako su značajke binarne, taj broj je nK. Broj parametara sada linearno ovisi o dimenziji n, a ne više eksponencijalno. Dakle, naivna pretpostavka u uvjetnoj nezavisnosti ulaznih varijabli omogućila nam je značajno smanjenje broja parametara.

Primjer 3.5 (Naivan Bayesov klasifikator) Naivan Bayesov klasifikator koristimo za klasifikaciju SF-filmova u kategoriju *Dobar film*. Koristimo sljedeće značajke:

$$\begin{aligned}
\text{Mjesto radnje} &= \{\text{svemir}, \text{Zemlja}, \text{drugdje}\} \\
\text{Glavni lik} &= \{\text{znanstvenica}, \text{dijete}, \text{kriminalac}, \text{policajac}\} \\
\text{Vrijeme radnje} &= \{\text{prošlost}, \text{budućnost}, \text{sadašnjost}\} \\
\text{Vanzemaljci} &= \{\text{da}, \text{ne}\}
\end{aligned}$$

[1] Od 1961., kada je ovaj model predložen, javlja se i pod drugim imenima: *idiot Bayes*, *simple Bayes*, *idependent Bayes*.

i	x_1 Mjesto radnje	x_2 Glavni lik	x_3 Vrijeme radnje	x_4 Vanzemaljci	y Dobar film
1	svemir	znanstvenica	sadašnjost	da	ne
2	Zemlja	kriminalac	budućnost	ne	ne
3	drugdje	dijete	prošlost	da	ne
4	svemir	znanstvenica	sadašnjost	ne	da
5	svemir	kriminalac	prošlost	ne	ne
6	Zemlja	dijete	prošlost	da	da
7	Zemlja	policajac	budućnost	da	ne
8	svemir	policajac	budućnost	ne	da

Model je binaran ($K = 2$), a broj vrijednosti pojedinih značajki je $K_1 = 3$, $K_2 = 4$, $K_3 = 3$ i $K_4 = 2$. Ukupan broj parametara ovog modela je:

$$K - 1 + \sum_{k=1}^{n} (K_k - 1)K = 1 + 2 \times (2 + 3 + 2 + 1) = 17\,.$$

Preglednosti radi, u nastavku navodimo redundantan skup od ukupno 26 parametara. ML-procjene za apriorne vjerojatnosti klasa su:

$$P(y = \text{da}) = \tfrac{1}{8}\textstyle\sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = \text{da}\} = \tfrac{3}{8} \qquad P(y = \text{ne}) = \tfrac{1}{8}\textstyle\sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = \text{ne}\} = \tfrac{5}{8} \tag{3.16}$$

ML-procjene za vjerojatnosti $P(x_k|\mathcal{C}_j)$ su:

$$\begin{array}{llll}
P(x_1 = \text{svemir}|y = \text{da}) & = 2/3 & P(x_1 = \text{svemir}|y = \text{ne}) & = 2/5 \\
P(x_1 = \text{Zemlja}|y = \text{da}) & = 1/3 & P(x_1 = \text{Zemlja}|y = \text{ne}) & = 2/5 \\
P(x_1 = \text{drugdje}|y = \text{da}) & = 0 & P(x_1 = \text{drugdje}|y = \text{ne}) & = 1/5 \\
 & & & \\
P(x_2 = \text{znanstvenica}|y = \text{da}) & = 1/3 & P(x_2 = \text{znanstvenica}|y = \text{ne}) & = 1/5 \\
P(x_2 = \text{dijete}|y = \text{da}) & = 1/3 & P(x_2 = \text{dijete}|y = \text{ne}) & = 1/5 \\
P(x_2 = \text{kriminalac}|y = \text{da}) & = 0 & P(x_2 = \text{kriminalac}|y = \text{ne}) & = 2/5 \\
P(x_2 = \text{policajac}|y = \text{da}) & = 1/3 & P(x_2 = \text{policajac}|y = \text{ne}) & = 1/5 \\
 & & & \\
P(x_2 = \text{prošlost}|y = \text{da}) & = 1/3 & P(x_2 = \text{prošlost}|y = \text{ne}) & = 2/5 \\
P(x_2 = \text{sadašnjost}|y = \text{da}) & = 1/3 & P(x_2 = \text{sadašnjost}|y = \text{ne}) & = 1/5 \\
P(x_2 = \text{budućnost}|y = \text{da}) & = 1/3 & P(x_2 = \text{budućnost}|y = \text{ne}) & = 2/5 \\
 & & & \\
P(x_2 = \text{da}|y = \text{da}) & = 1/3 & P(x_2 = \text{da}|y = \text{ne}) & = 3/5 \\
P(x_2 = \text{ne}|y = \text{da}) & = 2/3 & P(x_2 = \text{ne}|y = \text{ne}) & = 2/5
\end{array}$$

Razmotrimo sada klasifikaciju novog primjera $\mathbf{x}^{(1)} = (\text{svemir}, \text{dijete}, \text{sadašnjost}, \text{da})$:

$$
\begin{aligned}
P(\text{da}|\mathbf{x}^{(1)}) &\propto \prod_{k=1}^{n} P(x_k|y=\text{da})P(y=\text{da}) \\
&= P(\text{svemir}|\text{da})P(\text{dijete}|\text{da})P(\text{sadašnost}|\text{da})P(\text{da}|\text{da})P(\text{da}) \\
&= 2/3 \times 1/3 \times 1/3 \times 1/3 \times 3/8 = 0.009 \\
P(\text{ne}|\mathbf{x}^{(1)}) &\propto \prod_{k=1}^{n} P(x_k|y=\text{ne})P(y=\text{ne}) \\
&= P(\text{svemir}|\text{ne})P(\text{dijete}|\text{ne})P(\text{sadašnost}|\text{ne})P(\text{da}|\text{ne})P(\text{ne}) \\
&= 2/5 \times 1/5 \times 1/5 \times 3/5 \times 5/8 = 0.006
\end{aligned}
$$

MAP-hipoteza je:

$$h(\mathbf{x}^{(1)}) = \underset{\mathcal{C}\in\{\text{da,ne}\}}{\text{argmax}}\, P(\mathcal{C}|\mathbf{x}) = \text{da}$$

Što je s klasifikacijom primjera $\mathbf{x}^{(2)} = (\text{svemir}, \text{kriminalac}, \text{sadašnjost}, \text{ne})$?

$$
\begin{aligned}
P(\text{da}|\mathbf{x}^{(2)}) &\propto \prod_{k=1}^{n} P(x_k|y=\text{da})P(y=\text{da}) \\
&= P(\text{svemir}|\text{da})P(\text{kiminalac}|\text{da})P(\text{sadašnost}|\text{da})P(\text{ne}|\text{da})P(\text{da}) \\
&= 2/3 \times 0 \times 1/3 \times 2/3 \times 3/8 = 0 \\
P(\text{ne}|\mathbf{x}^{(2)}) &\propto \prod_{k=1}^{n} P(x_k|y=\text{da})P(y=\text{ne}) \\
&= P(\text{svemir}|\text{ne})P(\text{kriminalac}|\text{ne})P(\text{sadašnost}|\text{ne})P(\text{ne}|\text{ne})P(\text{ne}) \\
&= 2/5 \times 2/5 \times 1/5 \times 2/5 \times 5/8 = 0.008
\end{aligned}
$$

Zato što je $P(x = \text{kriminalac}|y = \text{da}) = 0$, aposteriorna vjerojatnost hipoteze $y = \text{da}$ je nula, neovisno o drugim vjerojatnostima u umnošku. Ovo je tipičan primjer prenaučenosti koji može nastupiti kod uporabe ML-procjena.

Primjer 3.6 (Zaglađeni naivan Bayesov klasifikator) Razmotrimo gornji skup za učenje i klasifikaciju primjer $\mathbf{x}^{(2)}$, ali uz zaglađivanje. Umjesto ML-procjena koristit ćemo Laplaceove procjene s $\lambda = 1$:

$$\hat{P}(x_k|\mathcal{C}_j) = \frac{N_{kj}+1}{N_j + K_k}\,.$$

Tako za procjene izglednosti dobivamo:

$$
\begin{array}{llll}
P(x_1 = \text{svemir}|y = \text{da}) & = \frac{2+1}{3+3} = 1/2 & P(x_1 = \text{svemir}|y = \text{ne}) & = \frac{2+1}{5+3} = 3/8 \\
P(x_1 = \text{kriminalac}|y = \text{da}) & = \frac{0+1}{3+4} = 1/7 & P(x_1 = \text{kriminalac}|y = \text{ne}) & = \frac{2+1}{5+4} = 1/3 \\
P(x_1 = \text{sadašnjost}|y = \text{da}) & = \frac{1+1}{3+3} = 1/3 & P(x_1 = \text{sadašnjost}|y = \text{ne}) & = \frac{1+1}{5+3} = 1/4 \\
P(x_1 = \text{ne}|y = \text{da}) & = \frac{2+1}{3+2} = 3/5 & P(x_1 = \text{ne}|y = \text{ne}) & = \frac{2+1}{5+2} = 3/7
\end{array}
$$

Za posteriorne vjerojatnosti sada dobivamo:

$$\begin{aligned}
P(\text{da}|\mathbf{x}^{(2)}) &= \prod_{k=1}^{n} P(x_k|y=\text{da})P(y=\text{da}) \\
&= P(\text{svemir}|\text{da})P(\text{kiminalac}|\text{da})P(\text{sadašnost}|\text{da})P(\text{ne}|\text{da})P(\text{da}) \\
&= 1/2 \times 1/7 \times 1/3 \times 3/5 \times 3/8 = 0.0054 \\
P(\text{ne}|\mathbf{x}^{(2)}) &= \prod_{k=1}^{n} P(x_k|y=\text{da})P(y=\text{ne}) \\
&= P(\text{svemir}|\text{ne})P(\text{kriminalac}|\text{ne})P(\text{sadašnost}|\text{ne})P(\text{ne}|\text{ne})P(\text{ne}) \\
&= 3/8 \times 1/3 \times 1/4 \times 3/7 \times 5/8 = 0.0084
\end{aligned}$$

Aposteriorno najvjerojatnija klasifikacija i dalje je $y = \text{ne}$, međutim razlika između aposteriornih vjerojatnosti sada je nešto manja.

3.3 Polunaivan Bayesov klasifikator

Zaključili smo da procjena za $P(x_1, \ldots, x_n|\mathcal{C}_j)$ izravno za vektor $\mathbf{x} = (x_1, \ldots, x_n)$ nema smisla jer ne omogućava generalizaciju. S druge strane, pretpostavka naivnog Bayesa o uvjetnoj nezavisnosti varijabli x_i s obzirom na klasu $\mathcal{C}_j$ vrlo je radikalna i u praksi nije zadovoljena. Postavlja se pitanje: možemo li napraviti model koji bi bio između ove dvije krajnosti? Takav model neke bi varijable tretirao kao uvjetno nezavisne, dok bi druge tretirao zajednički. Npr. umjesto faktorizacije:

$$P(\mathcal{C}_j|x_1, x_2, x_3) \propto P(x_1|\mathcal{C}_j)P(x_2|\mathcal{C}_j)P(x_3|\mathcal{C}_j)P(\mathcal{C}_j)$$

mogli bismo faktorizirati na sljedeći način:

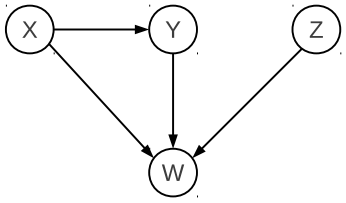
$$P(\mathcal{C}_j|x_1, x_2, x_3) \propto P(x_1|\mathcal{C}_j)P(x_2, x_3|\mathcal{C}_j)P(\mathcal{C}_j)$$

ili, ekvivalentno:

$$P(\mathcal{C}_j|x_1, x_2, x_3) \propto P(x_1|\mathcal{C}_j)P(x_2|\mathcal{C}_j)P(x_3|x_2, \mathcal{C}_j)P(\mathcal{C}_j)$$

pogotovo ukoliko se pokaže da ne vrijedi $x_2 \perp x_3|\mathcal{C}_j$, odnosno da varijable x_2 i x_3 nisu uvjetno nezavisne za klasu $\mathcal{C}_j$.

Ovakav model, kod kojega za neke varijable pretpostavljamo uvjetnu nezavisnost, dok za druge varijable to ne pretpostavljamo i modeliramo ih zajednički, nazivamo **polunaivan Bayesov klasifikator** (engl. *seminaïve Bayes classifier*). Očito je da će polunaivan model točnije modelirati podatke i davati točnije procjene. S druge strane, polunaivan model je složeniji od naivnog modela, odnosno ima više parametara i njegovo je učenje složenije. Naime, kada varijable tretiramo zajednički, broj parametara eksponencijalno ovisi o broju varijabli, a ne više linearno. Npr., za faktore $P(x_2|\mathcal{C}_j)P(x_3|\mathcal{C}_j)$ treba procijeniti $K(K_2 - 1) + K(K_3 - 1)$ parametara, dok za faktor $P(x_2, x_3|\mathcal{C}_j)$ treba procijeniti $K(K_2K_3 - 1)$ parametara. Posljedično, polunaivan model ima manju pristranost i veću varijancu, pa ga je lakše prenaučiti. Taj se problem međutim u načelu može izbjeći pravilnim odabirom modela, primjerice unakrsnom provjerom.



Slika 3.2: Bayesova mreža za četiri varijable.

Prije nego što se uopće krene učiti model polunaivnog Bayesovog klasifikatora, potrebno je odrediti koje varijable treba tretirati zajednički, a koje se mogu tretirati kao uvjetno nezavisne za danu klasu. Kada je broj varijabli malen, moguće je iscrpno ispitati sve moguće modele.[2] Za realne probleme (modele s desetak ili više varijabli) tako nešto nije izvedivo, već se moramo osloniti na heurističke metode optimizacije. Nekoliko takvih postupaka razmotrit ćemo u nastavku.

3.3.1 Bayesova mreža

Za daljnje razmatranje bit će nam korisno Bayesov klasifikator grafički prikazati kao usmjereni graf. Općenito, modele koji koriste graf kako bi na sažet način prikazali zajedničku distribuciju nazivamo **probabilistički grafički modeli** (engl. *probabilistic graphical models*). Posebice, grafički model kod kojega je graf aciklički i usmjeren nazivamo **Bayesova mreža** (engl. *Bayesian networks*).[3] Čvorovi Bayesove mreže predstavljaju slučajne varijable, a lukovi predstavljaju zavisnosti između varijabli. Ako varijabla X zavisi o varijabli o Y, tada crtamo lûk od čvora Y do čvora X.

Na slici 3.2 prikazana je jednostavna Bayesova mreža za četiri varijable. Ova mreža predstavlja grafički zapis zajedničke distribucije $P(X, Y, Z, W)$, faktorizirane kao:

$$P(X, Y, Z, W) = P(X)P(Y|X)P(Z)P(W|X, Y, Z)\,.$$

Nepostojanje određenih lukova upućuje na (uvjetnu) nezavisnost između varijabli. Tako su varijable X i Z nezavisne od svih drugih varijabli, varijabla Y ovisi samo o varijabli X (odnosno varijabla Y uvjetno je nezavisna od drugih varijabli, ako je dana varijabla X), dok varijabla W ovisi o sve tri varijable.

Općenito, Bayesova mreža sažeto zapisuje funkciju gustoće vjerojatnosti $p(\mathbf{x})$ kao:

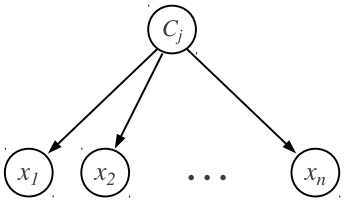
$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i|\mathrm{pa}(x_i))$$

gdje je $\mathrm{pa}(x_i)$ skup čvorova roditelja čvora x_i.

Na slici 3.3 prikazana je Bayesova mreža za naivan Bayesov klasifikator. Između varijabli x_i ne postoje izravne zavisnosti, no postoje zavisnosti između klase $\mathcal{C}_j$ i svake od varijabli x_i. Modeli polunaivnog Bayesovog klasifikatora razlikovat će se od ovog modela po tome što će neke od varijabli x_i biti združene u zajednički čvor, ili po tome što će između nekih od varijabli x_i postojati lukovi koji modeliraju zavisnost.

[2] Broj mogućih združivanja jednak je broju particija nad n varijabli i dan je Bellovim brojem B_n. Primjerice $B_3 = 5$, $B_5 = 52$, $B_{10} = 115975$.

[3] Također: *mreža vjerovanja* (engl. *belief network*) i *usmjeren aciklički grafički model*.



Slika 3.3: Naivan Bayesov klasifikator prikazan kao Bayesova mreža.

3.3.2 Algoritam FSSJ

Algoritam **unaprijednog slijednog odabira i združivanja** (engl. *forward sequential selection and joining*, FSSJ) (Pazzani, 1997) izgrađuje model polunaivnog Bayesovog klasifikatora na način da minimizira njegovu pogrešku. Algoritam kreće s nepovezanom Bayesovom mrežom. U svakoj idućoj iteraciji algoritam nastoji neku od varijabli x_i ili povezati s čvorem $\mathcal{C}_j$, ili je dodati u već postojeći čvor, ovisno o tome što daje manju pogrešku generalizacije. Riječ je zapravo o pohlepnom pretraživanju prostora stanja metodom usponom na vrh, pri čemu se kao kriterij optimizacije koristi pogreška generalizacije.

Algoritam FSSJ

1. Inicijaliziraj skup varijabli x_i koje se koriste u modelu na prazan skup. Graf je početno nepovezan, pa

   $$P(x_1, \ldots, x_n, \mathcal{C}_j) = P(x_1) \cdots P(x_n) P(\mathcal{C}_j)$$

   odnosno

   $$P(\mathcal{C}_j | x_1, \ldots, x_n) = P(\mathcal{C}_j)\,.$$

   Klasificiraj sve primjere iz skupa za provjeru u klasu $\mathcal{C}^*$ koja je najzastupljenija u skupu za učenje, $\mathcal{C}^* = \operatorname{argmax}_j P(\mathcal{C}_j)$.

2. Za varijablu x_i koja još nije uključena u model, razmotri sljedeće operacije:

   a) Uključivanje varijable x_i u model tako da se ona doda kao uvjetno nezavisna u odnosu na ostale varijable za danu klasu $\mathcal{C}_j$, tj. dodavanje lûka $(\mathcal{C}_j, x_i)$.

   b) Uključivanje varijable u model tako da se ona doda u zajednički čvor (tzv. superčvor) s nekom već uključenom varijablom.

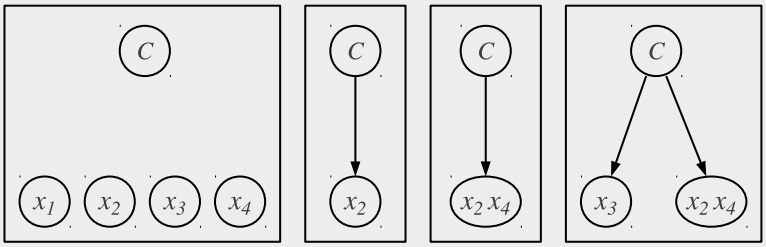
3. Izaberi varijablu i operaciju koja minimizira pogrešku generalizacije.

4. Ponavljaj od koraka 2 sve dok nema novih poboljšanja pogreške generalizacije.

Opisani algoritam je pohlepan jer nikada ne preispituje svoje odluke. Zbog toga nema garancije da je pronađeni model doista optimalan. (Teoretski je moguće da model koji izgradi algoritam FSSJ bude lošiji od modela naivnog Bayesovog klasifikatora.)

Treba naglasiti da je bitno da se optimizacija provodi unakrsnom provjerom, odnosno temeljem pogreške generalizacije (mjerene na skupu za provjeru), a ne temeljem empirijske pogreške (mjerene na skupu za učenje). Naime, minimalnu empirijsku pogrešku ostvario bi potpuno nefaktoriziran model (model u kojem je svih n varijabli združeno), ali, kao što smo već smo naglasili, takav bi model imao veliku pogrešku generalizacije.

Primjer 3.7 (Algoritam FSSJ) Razmotrimo izgradnju polunaivnog Bayesovog klasifikatora za četiri ulazne varijable: x_1, x_2, x_3, x_4. Jedan mogući slijed koraka izgradnje klasifikatora je ovaj:



Modelu odgovara aposteriorna razdioba

$$P(\mathcal{C}|x_1, x_2, x_3, x_4) \propto P(x_3|\mathcal{C})P(x_2, x_4|\mathcal{C})P(\mathcal{C})\,.$$

Označimo sa $E(x_i, [x_k, x_l])$ pogrešku generalizacije modela koji varijable x_k i x_l tretira združeno i uvjetno nezavisno od varijable x_i. Na temelju slijeda koraka algoritma FSSJ, možemo zaključiti da za vrijedi:

$$\begin{aligned}
E(x_2) &\leqslant E(x_i),\ i = 1, 3, 4\\
E([x_2, x_4]) &\leqslant E(x_2, x_i),\ i = 1, 3, 4\\
E([x_2, x_4]) &\leqslant E([x_2, x_i]),\ i = 1, 3\\
E([x_2, x_4], x_3) &\leqslant E([x_2, x_4], x_1)\\
E([x_2, x_4], x_3) &\leqslant E([x_2, x_4, x_i]),\ i = 1, 3\\
E([x_2, x_4], x_3) &\leqslant E(x_1, [x_2, x_4], x_3)\\
E([x_2, x_4], x_3) &\leqslant E([x_1, x_2, x_4], x_3)\\
E([x_2, x_4], x_3) &\leqslant E([x_2, x_4], [x_1, x_3])
\end{aligned}$$

Primijetite da algoritam u konačan model nije uključio varijablu x_1, jer njezino uključivanje u model ne bi dovelo do smanjenja pogreške generalizacije. Algoritam FSSJ dakle, osim izgradnje modela, ujedno provodi i **odabir značajki** (engl. *feature selection*).

Također primijetite da je algoritam FSSJ pohlepan. Npr., u ovom primjeru algoritam je provjerio i odbacio samo 15 modela od ukupno 52 moguća.[4] Možda se globalni minimum doseže sa $E([x_1, x_4], x_2, x_3)$, no taj model algoritam FSSJ nije provjeravao.

3.3.3 Klasifikator TAN

TAN (engl. *tree augmented naive Bayes*) (Friedman et al., 1997) je polunaivan Bayesov klasifikator čija se izgradnja temelji na mjeri uzajamne informacije među varijablama. **Uzajamna informacija** (engl. *mutual information*) između slučajnih varijabli X i Y defi-

[4] Budući da je moguće ne uključiti neku od varijabli, ukupan broj modela jednak je Bellovom broju B_{n+1} (jedna dodatna particija je za sve neuključene čvorove).

nirana je kao:

$$I(X,Y) = \sum_{j=1}^{K_Y} \sum_{i=1}^{K_X} P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)} \,. \tag{3.17}$$

U okviru teorije informacije, $I(X,Y)$ se tumači kao količina informacije koju varijable X i Y međusobno dijele, odnosno u kojoj mjeri poznavanje vrijednosti jedne od varijabli smanjuje nesigurnost o vrijednosti druge varijable. Mjera uzajamne informacije kvantificira stupanj stohastičke zavisnosti među varijablama i to je veća što je omjer između $P(X,Y)$ i $P(X)P(Y)$ manji; varijable X i Y su nezavisne akko $I(X,Y) = 0$. **Uvjetna uzajamna informacija** (engl. *conditional mutual information*) definirana je kao

$$I(X,Y|Z) = \sum_{k=1}^{K_Z} P(z_k) I(X,Y|z_k) = \sum_{k=1}^{K_Z} \sum_{j=1}^{K_Y} \sum_{i=1}^{K_X} P(x_i, y_j, z_k) \log \frac{P(x_i, y_j|z_k)}{P(x_i|z_k)P(y_j|z_k)} \,. \tag{3.18}$$

Klasifikator TAN zasniva se na ideji da je varijable za koje je mjera uvjetne uzajamne informacije $I(x_i, x_j|\mathcal{C}_j)$ najveća potrebno modelirati kao zavisne. Počevši od nepovezane Bayesove mreže, algoritam nastoji povezati parove čvorova s najvećom mjerom uvjetne uzajamne informacije, osim ako bi takvo povezivanje narušilo svojstvo acikličnosti grafa.

Algoritam izgradnje klasifikatora TAN

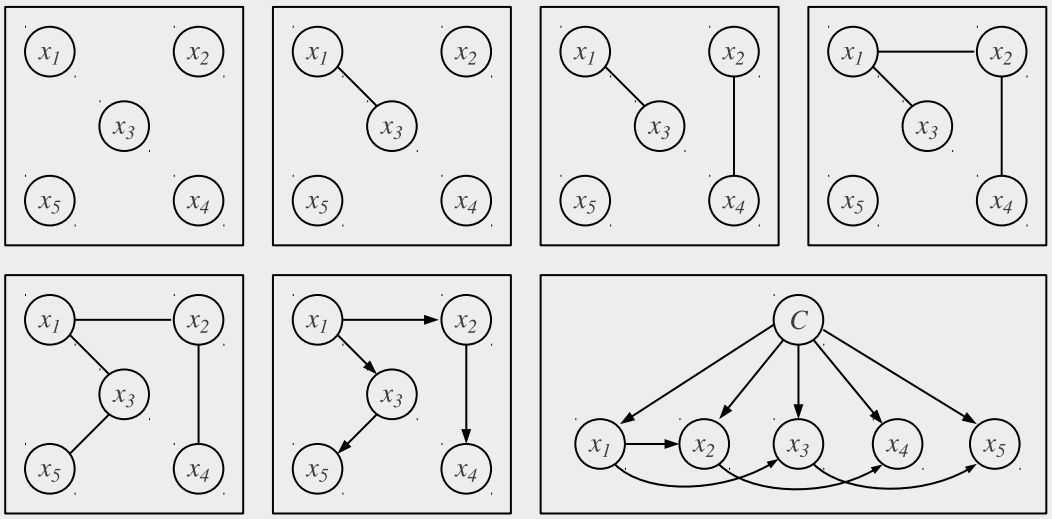
1. Izračunaj $I(x_i, x_j|\mathcal{C})$ za $i < j$, $i = 1, \ldots, n$ i sortiraj silazno.
2. Izgradi nepovezanu Bayesovu mrežu s čvorovima $x_1, \ldots, x_n$.
3. Razmotri par (x_i, x_j) s najvećom vrijednošću $I(x_i, x_j|\mathcal{C})$ i dodaj (neusmjereni) brid (x_i, x_j) ako time ne nastaje ciklus. Inače preskoči taj par i razmatraj idući par u listi.
4. Ponavljaj korak 3 dok ne izgradiš $n - 1$ bridova.
5. Pretvori neusmjeren graf u usmjeren graf tako da nasumično odabereš jedan čvor kao korijen.
6. Dodaj čvor $\mathcal{C}$ i poveži ga lukovima sa svim ulaznim varijablama.

Prva četiri koraka moguće je formulirati i na ovaj način: izgradi neusmjeren potpuno povezan težinski graf, kod kojega je težina brida (x_i, x_j) jednaka $I(x_i, x_j|\mathcal{C}_j)$, a zatim izračunaj razapinjući put maksimalne težine.

Primjer 3.8 (Klasifikator TAN) Izgradimo klasifikator TAN za ulazne varijable $x_1, \ldots, x_5$. Pretpostavimo da na skupu za učenje vrijedi:

$$I(x_1, x_3|\mathcal{C}) > I(x_2, x_4|\mathcal{C}) > I(x_1, x_2|\mathcal{C}) > I(x_3, x_4|\mathcal{C}) > I(x_1, x_4|\mathcal{C}) >$$
$$I(x_3, x_5|\mathcal{C}) > I(x_1, x_5|\mathcal{C}) > I(x_2, x_3|\mathcal{C}) > I(x_2, x_5|\mathcal{C}) > I(x_4, x_5|\mathcal{C}) \,.$$

Iz ovih je odnosa vidljivo da su, uz danu klasu $\mathcal{C}_j$, varijable x_1 i x_3 na skupu za učenje najmanje, a varijable x_4 i x_5 najviše uvjetno nezavisne. Koraci izgradnje klasifikatora TAN su:



Kao korijenski čvor izabran je čvor x_1. Dobiveni model odgovara aposteriornoj razdiobi

$$P(\mathcal{C}|x_1, x_2, x_3, x_4, x_5) \propto P(x_1|\mathcal{C})P(x_2|x_1, \mathcal{C})P(x_3|x_1, \mathcal{C})P(x_4|x_2, \mathcal{C})P(x_5|x_3, \mathcal{C})P(\mathcal{C})\,.$$

Opisani postupak zaslužuje nekoliko komentara. Nasumičan odabir korijenskog čvora na prvi se pogled može činiti kao nepoželjna nedeterminističnost. Međutim, odabir korijenskog čvora je nebitan jer nema utjecaja na konačan oblik zajedničke distribucije – zajednička distribucija može se faktorizirati na više načina budući da vrijedi $P(X|Y)P(Y) = P(Y|X)P(X)$. Treba međutim naglasiti da se vrijednosti $I(x_i, x_j|\mathcal{C}_j)$ računaju na temelju skupa za učenje $\mathcal{D}$ (npr. temeljem ML-procjena), pa se dakle radi o procjeni uvjetne uzajamne informacije, a ne o pravoj vrijednosti. Kao i uvijek kad je riječ o valjanim procjeniteljima, procjena će biti to točnija što je uzorak (odnosno broj primjera) veći.

Koja je razlika između modela izgrađenog algoritmom FSSJ, kod kojega se čvorovi združuju, i modela TAN, kod kojega se čvorovi povezuju? Razlika je u složenosti modela, odnosno u broju parametara. Klasifikator TAN može modelirati zavisnosti između pojedinačnih parova varijabli za koje zavisnost ne vrijedi tranzitivno, dok klasifikator izgrađen algoritmom FSSJ to ne može. Zbog toga će broj parametara modela TAN biti manji (v. primjer 3.9).

Primjer 3.9 (Broj parametara Bayesovog klasifikatora) Želimo izgraditi klasifikator s pet ulaznih varijabli, $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$. Pretpostavimo da sve varijable imaju tri moguće vrijednosti, a da je klasifikator binaran. Nadalje pretpostavimo da su sve varijable uvjetno nezavisne uz danu klasu $\mathcal{C}_j$, osim parova varijabli x_1 i x_2 te varijabli x_2 i x_3, koje su uvjetno zavisne uz danu klasu $\mathcal{C}_j$.

Izračunajmo ukupan broj parametara različitih varijanti diskretnih Bayesovih klasifikatora. Broj parametara ovisi o načinu faktorizacije zajedničke distribucije $P(\mathbf{x}, \mathcal{C}_j)$. Kod naivnog Bayesovog klasifikatora imamo

$$P(\mathbf{x}, \mathcal{C}_j) = P(\mathcal{C}_j) \prod_{i=1}^{5} P(x_i|\mathcal{C}_j)$$

pa je broj parametara jednak $(2-1)+5\times 2\times(3-1)=21$. Naivan Bayesov klasifikator ima najmanje parametara, ali u potpunosti zanemaruje zavisnost koja postoji između varijabli.

Polunaivan Bayesov klasifikator izgrađen algoritmom FSSJ može uzeti u obzir navedene zavisnosti između varijabli jedino tako da u jedan superčvor združi varijable x_1, x_2 i x_3:

$$P(\mathbf{x},\mathcal{C}_j) = P(x_1,x_2,x_3|\mathcal{C}_j)P(x_4|\mathcal{C}_j)P(x_5|\mathcal{C}_j)P(\mathcal{C}_j)$$

što daje ukupno $2\times(3^3-1)+2\times 2\times(3-1)+(2-1)=61$ parametar.

Klasifikator TAN zavisnost može modelirati s manje parametara, budući da može zasebno modelirati zavisnosti između parova varijabli x_1 i x_2 te x_2 i x_3:

$$P(\mathbf{x},\mathcal{C}_j) = P(x_1|x_2,\mathcal{C}_j)P(x_2|x_3,\mathcal{C}_j)P(x_3|\mathcal{C}_j)P(x_4|\mathcal{C}_j)P(x_5|\mathcal{C}_j)P(\mathcal{C}_j)$$

što daje ukupno $2\times 2\times 3\times(3-1)+2\times 2\times(3-1)+(2-1)=37$ parametara.

Općenito, za faktor $P(X_1,\ldots,X_i|X_{i+1},\ldots,X_n)$, kod kojeg svaka varijabla ima K stanja, treba procijeniti $\mathcal{O}(K^n)$ parametara. Ako taj broj želimo smanjiti, faktori trebaju biti što kraći (bolje je imati veći broj kraćih faktora nego manji broj duljih faktora).

3.3.4 Klasifikator k-DB

Kod klasifikatora TAN svaki čvor može imati samo jednog roditelja, odnosno svaka varijabla može, pored zavisnosti o klasi, biti uvjetno zavisna samo o jednoj ulaznoj varijabli. Bayesov model koji omogućava veći broj ovisnosti naziva se **klasifikator k-DB** (engl. *k-limited dependence bayesian classifiers*) (Sahami, 1996). Klasifikator k-DB može modelirati zavisnost varijable o najviše k drugih varijabli (ne računajući varijablu $\mathcal{C}_j$). Odabir čvorova između kojih se modeliraju zavisnosti provodi se, kao i kod klasifikatora TAN, temeljem mjere uvjetne uzajamne informacije $I(x_i,x_j|\mathcal{C}_j)$. Dodatno, varijable se u model uključuju redom prema relevantnosti, izračunate pomoću mjere uzajamne informacije $I(x_i,\mathcal{C}_j)$ između ulazne varijable i varijable klase. Model k-DB zavisnosti među varijablama može modelirati točnije nego model TAN, uz cijenu većeg broja parametara i veće sklonosti prenaučenosti.

Algoritam izgradnje klasifikatora k-DB

1. Izračunaj $I(x_i,\mathcal{C}_j)$ i $I(x_i,x_j|\mathcal{C}_j)$ za svaki par varijabli. Sortiraj varijable silazno po $I(x_i,\mathcal{C}_j)$.

2. Za varijablu x_i koja je prva u listi:

   a) Dodaj varijablu x_i u model i izbaci je iz liste.

   b) Postavi čvor $\mathcal{C}_j$ za roditelja čvora x_i.

   c) Od varijabli x_j koje su već uključene u model, njih k (ili manje, ako ih nema toliko) koje imaju najveću vrijednost $I(x_i,x_j|\mathcal{C}_j)$ postavi kao čvorove roditelje od x_i.

3. Ponavljaj prethodni korak dok lista nije prazna.

Budući da su svi lukovi uvijek usmjereni prema čvoru koji se u tekućoj iteraciji dodaje, a nikad od njega, konačan graf neće sadržavati usmjerene cikluse.

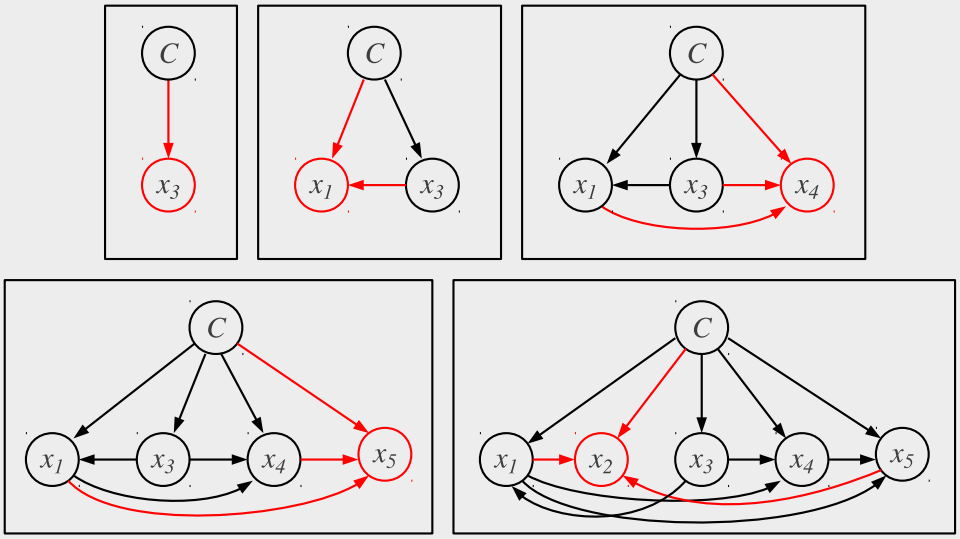
Primjer 3.10 (Klasifikator 2-DB) Izgradimo klasifikator 2-DB nad ulaznim varijablama $x_1, \ldots, x_5$. Neka su uzajamne informacije, procijenjene na skupu za učenje $\mathcal{D}$, takve da vrijedi:

$$I(x_3, \mathcal{C}) > I(x_1, \mathcal{C}) > I(x_4, \mathcal{C}) > I(x_5, \mathcal{C}) > I(x_2, \mathcal{C})$$

tj. varijabla $\mathcal{C}_j$ najviše ovisi o varijabli x_3, a najmanje o varijabli x_2. Ovim će redoslijedom varijable biti uključivane u model. Neka su uvjetne uzajamne informacije takve da vrijedi:

$$I(x_3, x_4|\mathcal{C}) > I(x_2, x_5|\mathcal{C}) > I(x_1, x_3|\mathcal{C}) > I(x_1, x_2|\mathcal{C}) > I(x_2, x_4|\mathcal{C}) > I(x_2, x_3|\mathcal{C}) > I(x_1, x_4|\mathcal{C}) > I(x_4, x_5|\mathcal{C}) > I(x_1, x_5|\mathcal{C}) > I(x_3, x_5|\mathcal{C})$$

tj. uz danu klasu $\mathcal{C}$, varijable x_3 i x_4 su najviše, a varijable x_3 i x_5 najmanje zavisne. Koraci izgradnje klasifikatora k-DB su (lukovi i čvorovi koji se dodaju obojani su crveno):



Dobiveni model odgovara aposteriornoj razdiobi

$$P(\mathcal{C}|x_1, x_2, x_3, x_4, x_5) \propto P(x_1|x_3, \mathcal{C})P(x_2|x_1, x_5, \mathcal{C})P(x_3|\mathcal{C})P(x_4|x_1, x_3, \mathcal{C}) P(x_5|x_1, x_4, \mathcal{C})P(\mathcal{C})\,.$$

3.3.5 Bayesova mreža kao klasifikator

Modeli polunaivnog Bayesovog klasifikatora koje smo ovdje opisali temelje se na ideji proširenja naivnog modela tako da se u obzir uzmu neke od zavisnosti među varijablama. Problemu klasifikacije moguće je pristupiti i obrnuto, na način da se najprije izgradi potpuna Bayesova mreža koja modelira neki problem, a zatim se lokalni dijelovi te mreže koriste za klasifikaciju. Može se pokazati da vrijednost svake varijable u Bayesovoj mreži ovisi samo o roditeljskim čvorovima, njihovoj djeci, i roditeljima čvorova djece. Taj se

skup čvorova naziva **Markovljev omotač** (engl. *Markov blanket*). Klasifikacija se svodi na zaključivanje nad čvorovima koji čine Markovljev omotač, za što se može upotrebiti bilo koji od algoritama zaključivanja nad Bayesovom mrežom. Za izgradnju mreže možemo upotrijebiti neki od algoritama za učenje strukture mreže (npr. algoritmi PC ili K2).

3.4 Bayesov klasifikator za kontinuirane varijable

Ako su varijable kontinuirane, izglednosti klasa tipično se modeliraju Gaussovom razdiobom. Osnovni razlog za to je analitička jednostavnost. Pored toga, mnogi prirodni fenomeni doista se pokoravaju normalnoj razdiobi u smislu da se primjere jedne klase može tretirati kao vrijednosti koje blago odstupaju od neke srednje vrijednosti. Treba naravno napomenuti da kontinuirane varijable ne moraju uvijek biti normalno distribuirane i da postoje statistički testovi kojima se može utvrditi je li to doista slučaj (npr. Kolmogorov-Smirnovov test ili Shapiro-Wilkov test). Unatoč tome, u većini slučajeva normalna je razdioba dovoljno dobra aproksimacija stvarnih podataka, pod uvjetom da primjeri iz pojedine klase oblikuju jednu grupu (u protivnom treba koristiti model Gaussove mješavine).

Razmotrimo prvo jednodimenzijski (univarijatan) Bayesov klasifikator, a zatim višedimenzijski (multivarijatan) Bayesov klasifikator za kontinuirane varijable. Jednodimenzijski klasifikator nije u praksi zanimljiv, ali razmatramo ga radi boljeg razumijevanja višedimenzijskog klasifikatora.

3.4.1 Jednodimenzijski Bayesov klasifikator

Kod jednodimenzijskog (univarijatnog) Bayesovog klasifikatora, izglednost svake klase $\mathcal{C}_j$ modeliramo jednodimenzijskom Gaussovom gustoćom $\mathcal{N}(\mu_j, \sigma_j^2)$:

$$p(x|\mathcal{C}_j) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left\{ -\frac{(x-\mu_j)^2}{2\sigma_j^2} \right\}. \tag{3.19}$$

Model možemo definirati kao

$$h_j(x) = p(x|\mathcal{C}_j)P(\mathcal{C}_j)$$

što je, budući da nas zanima samo maksimizacija, istovjetno s

$$h_j(x) = \ln p(x|\mathcal{C}_j) + \ln P(\mathcal{C}_j)\,. \tag{3.20}$$

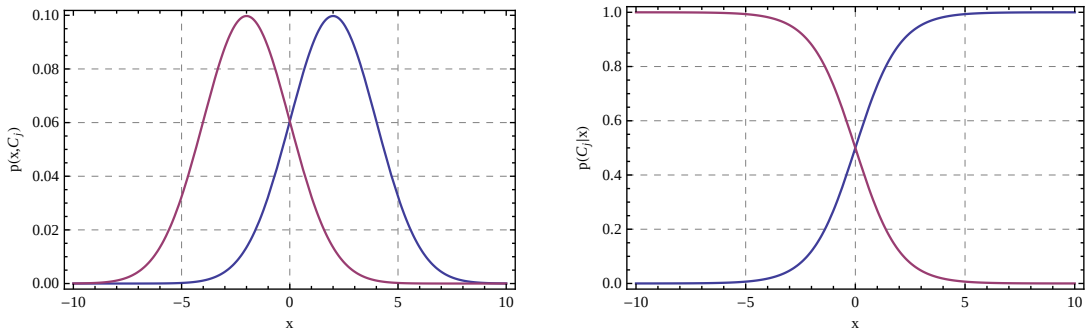
Uvrštenjem (3.19) u (3.20) dobivamo

$$h_j(x|\boldsymbol{\theta}_j) = -\frac{1}{2}\ln 2\pi - \ln\sigma_j - \frac{(x-\mu_j)^2}{2\sigma_j^2} + \ln P(\mathcal{C}_j) \tag{3.21}$$

čime je definiran parametarski model koji se sastoji od po tri parametra za svaku klasu, $\boldsymbol{\theta}_j = (\mu_j, \sigma_j, P(\mathcal{C}_j))$.

Raspolažemo skupom primjera za učenje $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$, gdje $y^{(i)} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$. Na temelju primjera iz skupa $\mathcal{D}$, parametre μ_j i σ_j^2 možemo procijeniti metodom najveće izglednosti, zasebno za svaku klasu $\mathcal{C}_j$. Tako dobivamo:

$$\hat{\mu}_j = \frac{1}{N_j}\sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = \mathcal{C}_j\} x^{(i)} \qquad \hat{\sigma}_j^2 = \frac{1}{N_j}\sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = \mathcal{C}_j\}(x^{(i)} - \hat{\mu}_j)^2 \tag{3.22}$$



(a) zajednička gustoća $p(x, \mathcal{C}_j)$ (b) aposteriorne vjerojatnosti $p(\mathcal{C}_j|x)$

Slika 3.4: Dvoklasni Gaussov model za klase jednakih varijanci i jednakih apriornih vjerojatnosti: $p(x|\mathcal{C}_1) \sim \mathcal{N}(-2, 4)$ i $p(x|\mathcal{C}_2) \sim \mathcal{N}(2, 4)$.

dok za procjenu apriornih vjerojatnosti $P(\mathcal{C}_j)$ koristimo (3.8). Uvrštavanjem ovih procjena u (3.21) dobivamo model

$$h_j(x) = -\ln \hat{\sigma}_j - \frac{(x - \hat{\mu}_j)^2}{2\hat{\sigma}_j^2} + \ln \hat{P}(\mathcal{C}_j) \tag{3.23}$$

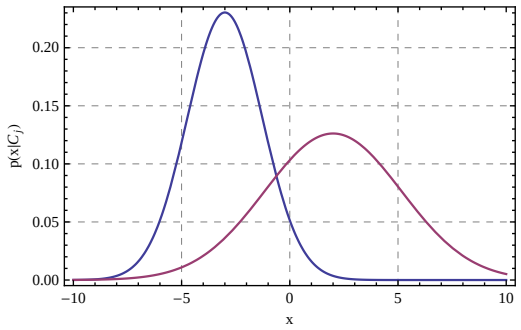
pri čemu smo prvi pribrojnik iz (3.21) zanemarili, budući da je za sve funkcije h_j identičan.

Na slici 3.4 prikazan je model za slučaj dviju klasa jednakih varijanci, $\hat{\sigma}_1^2 = \hat{\sigma}_2^2$, i jednakih apriornih vjerojatnosti, $\hat{P}(\mathcal{C}_1) = \hat{P}(\mathcal{C}_2)$. U ovom slučaju granica između područja nalazi se točno na polovici između srednjih vrijednosti $\hat{\mu}_1$ i $\hat{\mu}_2$, budući da tada, prema (3.21), vrijedi:

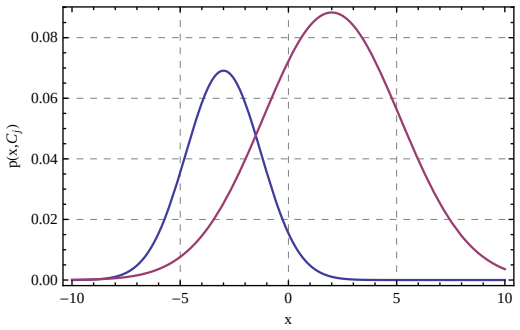
$$\begin{aligned} h_1(x) &= h_2(x) \\ (x - \hat{\mu}_1)^2 &= (x - \hat{\mu}_2)^2 \\ x &= \frac{1}{2}(\hat{\mu}_1 + \hat{\mu}_2) \end{aligned}$$

Na slici 3.5 prikazan je model za općenitiji slučaj dviju klasa s različitim srednjim vrijednostima, varijancama i apriornim vjerojatnostima. Zato što su varijance različite, funkcije aposteriorne vjerojatnosti presijecaju u dvije točke, pa postoje dvije granice između klasa. Također, zato što su apriorne vjerojatnosti klasa različite, granica područja primiče se klasi s manjom apriornom vjerojatnošću.

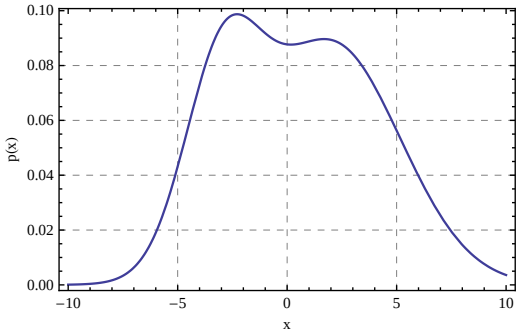
Treba naglasiti da smo, definiravši hipotezu kao (3.20), izgubili vjerojatnosnu interpretaciju hipoteze. To znači da vrijednost hipoteze $h_j(x)$ ne možemo tumačiti kao vjerojatnost da primjer x pripada klasi $\mathcal{C}_j$. To je vidljivo iz usporedbe slika 3.5d i 3.5b. Primjerice, vjerojatnost da primjer $x = 5$ pripada klasi $\mathcal{C}_2$ približno je jednaka vjerojatnosti da primjer $x = 10$ pripada istoj toj klasi. Međutim, nenormalizirana vrijednost, odnosno vrijednost zajedničke gustoće $p(x, \mathcal{C}_1)$, mnogo je manja za $x = 10$ nego za $x = 5$. Vidimo dakle da, ako hipoteza nije normalizirana, ne možemo uspoređivati rezultate klasifikacija različitih primjera. Međutim, možemo uspoređivati pouzdanosti klasifikacije pojedinačnog primjera u različite klase. Tako možemo zaključiti da je klasifikacija primjera $x = 5$ u klasu $\mathcal{C}_2$ znatno pouzdanija nego klasifikacija istog tog primjera u klasu $\mathcal{C}_1$.



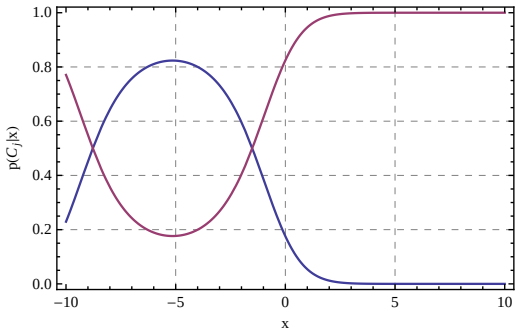
(a) izglednosti klasa $p(x|\mathcal{C}_j)$



(b) zajednička gustoća $p(x, \mathcal{C}_j)$



(c) gustoća primjera $p(x)$



(d) aposteriorne vjerojatnosti $P(\mathcal{C}_j|x)$

Slika 3.5: Dvoklasni Gaussov model. Klase su modelirane izglednostima $p(x|\mathcal{C}_1) \sim \mathcal{N}(-3, 3)$ i $p(x|\mathcal{C}_2) \sim \mathcal{N}(2, 10)$. Apriorna vjerojatnosti klasa su $P(\mathcal{C}_1) = 0.3$ i $P(\mathcal{C}_2) = 0.7$.

Primjer 3.11 (Jednodimenzijski Bayesov klasifikator) Želimo naučiti univarijatni Bayesov klasifikator za klasifikaciju primjera u tri klase, $\mathcal{C}_1$, $\mathcal{C}_2$ i $\mathcal{C}_3$. Prostor primjera je $\mathcal{X} = \mathbb{R}$, a skup primjera za učenje je

$$\mathcal{D} = \big\{(-1.52, \mathcal{C}_2), (1.28, \mathcal{C}_1), (0.56, \mathcal{C}_1), (4.15, \mathcal{C}_3), (3.36, \mathcal{C}_1), (-7.59, \mathcal{C}_2), (-1.14, \mathcal{C}_1), (4.05, \mathcal{C}_3), (4.09, \mathcal{C}_1), (5.24, \mathcal{C}_3), (3.72, \mathcal{C}_3), (-1.07, \mathcal{C}_1), (0.09, \mathcal{C}_2), (3.68, \mathcal{C}_3), (2.49, \mathcal{C}_3)\big\}.$$

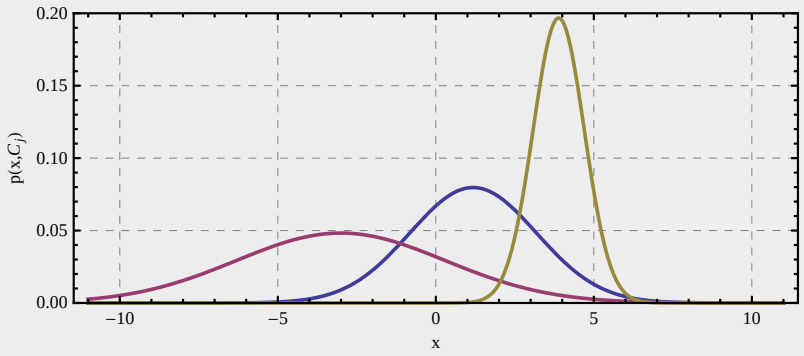
Primjeri poprimaju kontinuirane vrijednosti, pa izglednosti klasa modeliramo Gaussovim gustoćama. Procjenu parametara gustoća $p(x|\mathcal{C}_j)$, $j = 1, 2, 3$, računamo prema (3.22):

$$\hat{\mu}_1 = 1.18 \qquad \hat{\sigma}_1^2 = 4.01$$
$$\hat{\mu}_2 = -3.01 \qquad \hat{\sigma}_2^2 = 10.94$$
$$\hat{\mu}_3 = 3.89 \qquad \hat{\sigma}_3^2 = 0.66$$

(Primijetite da su procjene $\hat{\sigma}_j^2$ pristrane i da ih se može ispraviti tako da ih se pomnoži s $N/(N-1)$.) Procjenu apriornih vjerojatnosti klasa računamo prema (3.8) i dobivamo

$$\hat{P}(\mathcal{C}_1) = 0.4 \qquad \hat{P}(\mathcal{C}_2) = 0.2 \qquad \hat{P}(\mathcal{C}_3) = 0.4$$

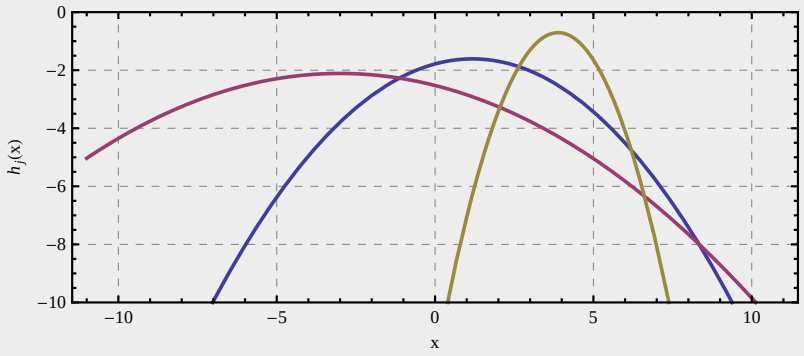
Zajedničke gustoće $p(x, \mathcal{C}_j) = p(x|\mathcal{C}_j)P(\mathcal{C}_j)$ izgledaju ovako



Uvrštavanjem u (3.23) za klase $\mathcal{C}_1$, $\mathcal{C}_2$ i $\mathcal{C}_3$ dobivamo sljedeće hipoteze:

$$\begin{aligned} h_1(x) &= -0.12x^2 + 0.29x - 1.79 \\ h_2(x) &= -0.05x^2 - 0.28x - 2.53 \\ h_3(x) &= -0.76x^2 + 5.91x - 12.19 \end{aligned}$$

koje izgledaju ovako:



Granice između klasa mogu se odrediti rješavanjem (ne)jednadžbi $h_j(x) = h_k(x) \gtrless h_l(x)$ $(j \neq k,\, j \neq l,\, k \neq l)$.

3.4.2 Višedimenzijski Bayesov klasifikator

Kod višedimenzijskog slučaja, $\mathcal{X} = \mathbb{R}^n$, izglednosti klasa modeliramo multivarijatnom Gaussovom gustoćom $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$:

$$p(\mathbf{x}|\mathcal{C}_j) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_j|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^{\mathrm{T}} \boldsymbol{\Sigma}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j) \right\}. \tag{3.24}$$

Vektor $\boldsymbol{\mu}_j$ opisuje prototipnu vrijednost primjera u klasi $\mathcal{C}_j$, dok matrica kovarijacije opisuje količinu šuma u svakoj varijabli te korelaciju između izvora šuma. Klasama odgovaraju hiperelipsoidi čije je usmjerenje određeno faktorima korelacije. Uvrštavanjem u model

$$h_j(\mathbf{x}) = \ln p(\mathbf{x}|\mathcal{C}_j) + \ln P(\mathcal{C}_j) \tag{3.25}$$

dobivamo

$$h_j(\mathbf{x}) = -\frac{n}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j) + \ln P(\mathcal{C}_j)\,. \tag{3.26}$$

Ovaj model ukupno ima Kn parametara za srednje vrijednosti te $\frac{Kn}{2}(n+1)$ parametara za matrice kovarijacije (prisjetite se da je kovarijacijska matrica simetrična pa je dovoljno pohraniti polovicu od ukupno n^2 parametara).

Raspišimo (3.26) kako bismo utvrdili je li model linearan ili nelinearan. Primijetite da je prvi pribrojnik jednak za sve klase, pa ga možemo zanemariti. Sređivanjem izraza dobivamo

$$h_j(\mathbf{x}) = -\frac{1}{2}\ln|\boldsymbol{\Sigma}_j| - \frac{1}{2}(\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}\mathbf{x} - 2\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j + \boldsymbol{\mu}_j^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j) + \ln P(\mathcal{C}_j)\,. \tag{3.27}$$

Primijetite da vrijedi $\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j = \boldsymbol{\mu}_j^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x}$, jer se radi o skalarnoj vrijednosti. Izraz $\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x}$ je kvadratna forma, pa funkcija $h_j(\mathbf{x})$ kvadratno ovisi o $\mathbf{x}$. Posljedično, granica između dviju klasa, $\mathcal{C}_j$ i $\mathcal{C}_k$, koju možemo izvesti rješavanjem jednadžbe $h_j(\mathbf{x}) = h_k(\mathbf{x})$, bit će paraboloid (slika 3.6a). Model je dakle nelinearan.

Za dani skup za učenje $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$ parametre modela procjenjujemo zasebno za svaku klasu $\mathcal{C}_j$. Ako koristimo metodu najveće izglednosti, procjene su:

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

Opisani model može vrlo točno modelirati podatke, ali je njegov nedostatak velik broj parametara. Ako je skup za učenje malen, vrlo je teško načiniti pouzdanu procjenu tih parametara. Pored toga, velik broj parametara često predstavlja i praktičan problem u smislu računalnih resursa. Kako je broj parametara $\mathcal{O}(n^2)$, problem je osobito izražen kod velikih dimenzija. Zbog toga ćemo u nastavku razmotriti niz mogućih pojednostavljenja ovog modela.

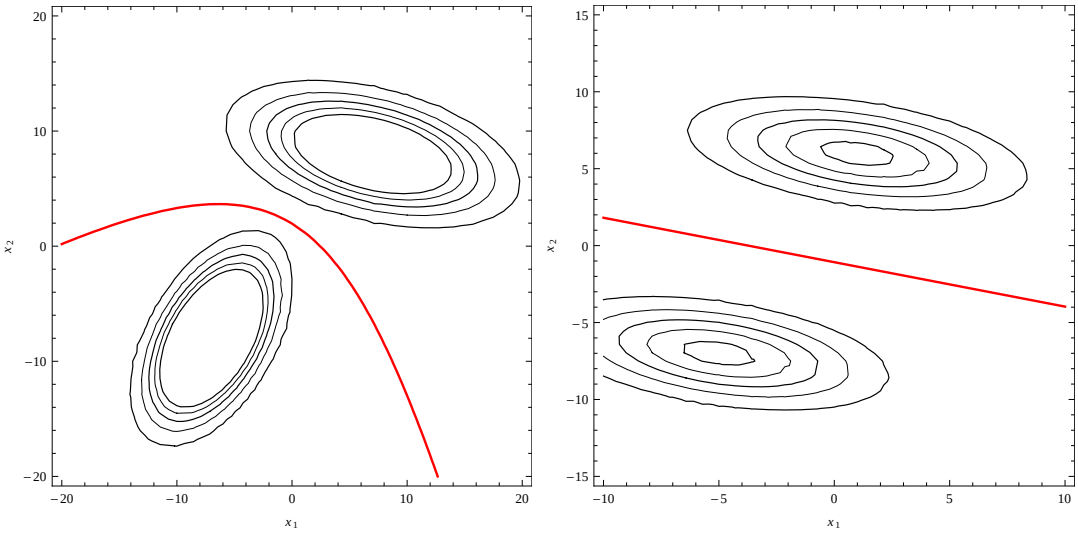
1. pojednostavljenje: dijeljena kovarijacijska matrica

Model se može pojednostaviti ako se pretpostavi da je kovarijacijska matrica jednaka za svaku klasu, odnosno da je dijeljena između klasa. Takvu matricu možemo procijeniti kao kao težinski zbroj pojedinačnih kovarijacijskih matrica:

$$\hat{\boldsymbol{\Sigma}} = \sum_j \hat{P}(\mathcal{C}_j)\hat{\boldsymbol{\Sigma}}_j\,. \tag{3.28}$$

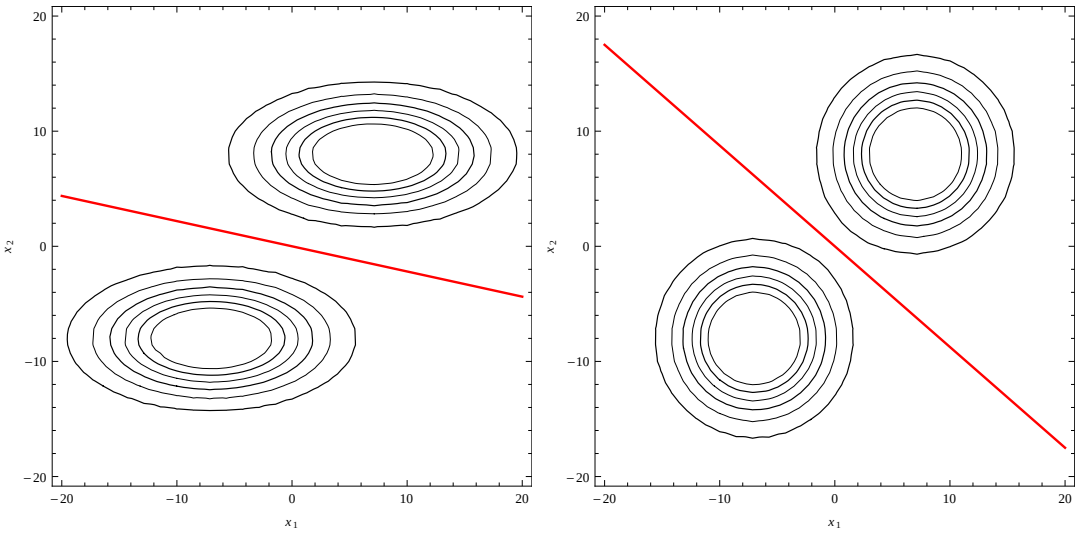
U tom slučaju kovarijacijska matrica $\boldsymbol{\Sigma}$ identična je za sve klase, pa za fiksirani $\mathbf{x}$ vrijedi

$$\underbrace{\frac{1}{2}\ln|\boldsymbol{\Sigma}|}_{=\text{konst.}} - \frac{1}{2}(\underbrace{\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x}}_{=\text{konst.}} - 2\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j + \boldsymbol{\mu}_j^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j) + \ln P(\mathcal{C}_j)$$



(a) Različite kov. matrice

(b) Dijeljena kov. matrica



(c) Dijeljena dijagonalna kov. matrica

(d) Dijeljena izotropna kov. matrica

Slika 3.6: Granica između dviju klasa za bivarijatni Gaussov model.

pa model (3.27) degenerira u

$$h_j(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j - \frac{1}{2}\boldsymbol{\mu}_j^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j + \ln P(\mathcal{C}_j)\,. \tag{3.29}$$

Primijetite da je iščeznuo član koji kvadratno ovisi o $\mathbf{x}$, pa su granice između klasa sada linearne (slika 3.6b). U slučaju da klase nisu linearno razdvojive, ovaj će model manje točno modelirati podatke. Model ima ukupno Kn parametara za srednje vrijednosti te $\frac{n}{2}(n+1)$ parametara za dijeljenu kovarijacijsku matricu. Broj parametara još uvijek kvadratno ovisi o n, što može biti problematično kod visokih dimenzija.

2. pojednostavljenje: dijagonalna kovarijacijska matrica

Daljnje pojednostavljenje modela moguće je uz pretpostavku da varijable nisu korelirane odnosno da su nezavisne.[5] U tom slučaju koristimo dijagonalnu kovarijacijsku matricu, $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_i^2)$. Matrica $\boldsymbol{\Sigma}^{-1}$ onda je također dijagonalna te vrijedi $\boldsymbol{\Sigma}^{-1} = \mathrm{diag}(1/\sigma_i^2)$ i $|\boldsymbol{\Sigma}| = \prod_i \sigma_i$. Multivarijatna Gaussova gustoća (3.24) degenerira u produkt univarijatnih Gaussovih razdiobi:

$$\begin{aligned} p(\mathbf{x}|\mathcal{C}_j) &= \frac{1}{(2\pi)^{n/2}\prod_{i=1}^n \sigma_i} \exp\Big\{ -\frac{1}{2}\sum_{i=1}^n \Big(\frac{x_i - \mu_{ij}}{\sigma_i}\Big)^2 \Big\} \\ &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_i} \exp\Big\{ -\frac{1}{2}\Big(\frac{x_i - \mu_{ij}}{\sigma_i}\Big)^2 \Big\}\,. \end{aligned} \tag{3.30}$$

Model koji smo dobili jest **naivan Bayesov klasifikator** za kontinuirane varijable. (Primijetite da za $n = 1$ dobivamo (3.19)). Uvrštavanjem (3.30) u (3.29) i zanemarivanjem pribrojnika koji su za sve klase jednaki dobivamo

$$h_j(\mathbf{x}) = -\frac{1}{2}\sum_{i=1}^n \Big(\frac{x_i - \mu_{ij}}{\sigma_i}\Big)^2 + \ln P(\mathcal{C}_j)\,. \tag{3.31}$$

Izraz $((x_i - \mu_{ij})/\sigma_i)^2$ odgovara udaljenosti između x_i i μ_{ij} izraženoj u jedinicama standardne devijacije. Geometrijski gledano, prvi pribrojnik u (3.31) jednak je kvadratu **normalizirane euklidske udaljenosti** između vrijednosti x_i i vrijednosti μ_{ij} (srednje vrijednosti dimenzije i za klasu $\mathcal{C}_j$). Normalizirana euklidska udaljenost neosjetljiva je na razlike u varijanci između pojedinih dimenzija. Varijance općenito mogu biti različite, pa klasama odgovaraju hiperelipsoidi, koji su međutim poravnati s osima jer su varijable nekorelirane (slika 3.6c). Broj parametara modela jest Kn za srednje vrijednosti klasa te n za varijance, što predstavlja smanjenje sa $\mathcal{O}(n^2)$ na $\mathcal{O}(n)$.

3. pojednostavljenje: izotropna kovarijacijska matrica

Moguća su daljnja pojednostavljenja modela. Možemo pretpostaviti da su varijance iste za svaku dimenziju, $\sigma_i = \sigma$, odnosno da je matrica kovarijacije izotropna, $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$.

$$h_j(\mathbf{x}) = -\frac{1}{2\sigma^2}\underbrace{\sum_{i=1}^n (x_i - \mu_{ij})^2}_{\|\mathbf{x}-\boldsymbol{\mu}\|^2} + \ln P(\mathcal{C}_j)\,. \tag{3.32}$$

[5] Nekoreliranost ne implicira nezavisnost, ali budući da u modelu ni na koji drugi način ne modeliramo zavisnost, ukidanje koreliranosti znači zapravo da pretpostavljamo nezavisnost.

Tablica 3.1: Složenost modela u ovisnosti o pretpostavkama o izglednostima klasa.

Pretpostavka	Kovarijacijska matrica	Broj parametara	
Različite, hiperelipsoidi	$\boldsymbol{\Sigma}_j$	$Kn(n+1)/2 + Kn$	$\mathcal{O}(n^2)$
Dijeljena, hiperelipsoidi	$\boldsymbol{\Sigma}$	$n(n+1)/2 + Kn$	$\mathcal{O}(n^2)$
Različite, poravnati hiperelipsoidi	$\boldsymbol{\Sigma}_j = \mathrm{diag}(\sigma_{i,j}^2)$	$2Kn$	$\mathcal{O}(n)$
Dijeljena, poravnati hiperelipsoidi	$\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_i^2)$	$n + Kn$	$\mathcal{O}(n)$
Različite, hipersfere	$\boldsymbol{\Sigma}_j = \sigma_j^2\mathbf{I}$	$K + Kn$	$\mathcal{O}(n)$
Dijeljena, hipersfere	$\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$	$1 + Kn$	$\mathcal{O}(n)$

Nomalizirana euklidska udaljenost svodi se na običnu euklidsku udaljenost, a klase su hipersfere sa središtima u $\boldsymbol{\mu}_j$ (slika 3.6d).

4. pojednostavljenje: jednake apriorne vjerojatnosti klasa

Konačno, model možemo pojednostaviti tako da pretpostavimo da su apriorne vjerojatnosti klasa jednake. Tada za model dobivamo

$$h_j(\mathbf{x}) = -\|\mathbf{x} - \boldsymbol{\mu}_j\|^2$$

gdje $\|\cdot\|$ označava normu vektora.[6] Model primjer $\mathbf{x}$ klasificira u klasu $\mathcal{C}_j$ s najbližom srednjom vrijednošću $\boldsymbol{\mu}_j$. Kao i ranije, izraz možemo raspisati kako bismo utvrdili kakav je oblik granice između klasa:

$$h_j(\mathbf{x}) = -\|\mathbf{x} - \boldsymbol{\mu}_j\|^2 = -(\mathbf{x} - \boldsymbol{\mu}_j)^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu}_j) = -(\mathbf{x}^{\mathrm{T}}\mathbf{x} - 2\mathbf{x}^{\mathrm{T}}\boldsymbol{\mu}_j + \boldsymbol{\mu}_j^{\mathrm{T}}\boldsymbol{\mu}_j)\,.$$

Prvi pribrojnik zajednički je svim klasama, pa ga možemo zanemariti. Granica između dviju klasa dakle je linearna

$$h_j(\mathbf{x}) = \mathbf{w}_j^{\mathrm{T}}\mathbf{x} + w_{j0}$$

s težinama $\mathbf{w}_j = \boldsymbol{\mu}_j$ i slobodnim članom $w_{j0} = -\frac{1}{2}\boldsymbol{\mu}_j^{\mathrm{T}}\boldsymbol{\mu}_j = -\frac{1}{2}\|\boldsymbol{\mu}_j\|^2$.

Opisali smo niz Bayseovih modela temeljenih na Gaussovoj gustoći. Pregled modela dan je u tablici 3.1. Kod najsloženijeg modela koristili smo zasebnu kovarijacijsku matricu za svaku klasu. Prijelaz s nelinearnog modela na linearan ostvarili smo uporabom dijeljene kovarijacijske matrice. Daljnja pojednostavljenja ostvarili smo uvođenjem dodatnih pretpostavki o izglednostima klasa, i to uporabom dijagonalne ili izotropne kovarijacijske matrice. Moguće su i druge kombinacije, npr. korištenje dijagonalnih, ali nedijeljenih kovarijacijskim matrica. Uvođenje dodatnih pretpostavki uvijek dovodi do smanjenja broja parametara, što za posljedicu ima povećanje pristranosti i smanjenje varijance modela. Za odabir optimalnog modela može se koristiti npr. unakrsna provjera.

Primjer 3.12 (Bayesov višedimenzijski klasifikator s kontinuiranim ulazima) Želimo izgraditi Bayesov klasifikator za klasifikaciju primjera iz prostora $\mathbb{R}^3$ u klase $\mathcal{C}_1$ i $\mathcal{C}_2$. Metodom najveće izglednosti dobili smo sljedeće procjene parametara na skupu za

[6] *Normu* ili duljinu vektora $\mathbf{x}$ definiramo kao $\|\mathbf{x}\| = \sqrt{\mathbf{x}^{\mathrm{T}}\mathbf{x}}$.

učenje:

$$\hat{\boldsymbol{\mu}}_1 = (1, -5, 0) \qquad \hat{\boldsymbol{\Sigma}}_1 = \begin{pmatrix} 1 & -0.75 & 0.2 \\ -0.75 & 6.25 & 1.5 \\ 0.2 & 1.5 & 4 \end{pmatrix} \qquad \hat{P}(\mathcal{C}_1) = 0.4$$

$$\hat{\boldsymbol{\mu}}_2 = (7, 2, 3) \qquad \hat{\boldsymbol{\Sigma}}_2 = \begin{pmatrix} 4 & 0.7 & 2 \\ 0.7 & 12.25 & -3.5 \\ 2 & -3.5 & 16 \end{pmatrix} \qquad \hat{P}(\mathcal{C}_2) = 0.6$$

Razmotrimo četiri modela, od nasloženijeg do najjednostavnijeg. Najsloženiji model je onaj definiran izrazom (3.27). Taj model ukupno ima 19 parametara (12 za kovarijacijsku matricu, 6 za vektore srednjih vrijednosti i 1 za apriornu vjerojatnost jedne od klasa). Uvrštavanjem gornjih procjena u (3.27) i izračunom determinanti i inverza kovarijacijskih matrica, dobivamo konkretne izraze za hipoteze dviju klasa:

$$\begin{aligned} h_1(\mathbf{x}) = &- 0.57x_1^2 - 0.10x_2^2 - 0.14x_3^2 - 0.17x_1x_2 + 0.12x_1x_3 + 0.08x_2x_3 \\ &+ 0.32x_1 - 0.83x_2 + 0.30x_3 - 4.70 \\ h_2(\mathbf{x}) = &- 0.14x_1^2 - 0.04x_2^2 - 0.04x_3^2 + 0.03x_1x_2 + 0.04x_1x_3 - 0.02x_2x_3 \\ &+ 1.70x_1 + 0.06x_2 - 0.02x_3 - 9.90 \end{aligned}$$

Hipoteze kvadratno ovise o ulaznim varijablama. Granica između dviju klasa je krivulja za koju $h_1(\mathbf{x}) = h_2(\mathbf{x})$, koja će također biti kvadratna funkcija od $\mathbf{x}$. Jednostavniji model je model s dijeljenom kovarijacijskom matricom. Prema (3.28), dijeljena matrica jednaka je

$$\hat{\boldsymbol{\Sigma}} = 0.4\hat{\boldsymbol{\Sigma}}_1 + 0.6\hat{\boldsymbol{\Sigma}}_2 = \begin{pmatrix} 2.8 & 0.12 & 1.28 \\ 0.12 & 9.85 & -1.5 \\ 1.28 & -1.5 & 11.2 \end{pmatrix}.$$

Ukupan broj parametara modela je 13. Uvrštavanjem u (3.29) dobivamo sljedeće hipoteze:

$$\begin{aligned} h_1(\mathbf{x}) &= 0.44x_1 - 0.53x_2 - 0.12x_3 - 2.50 \\ h_2(\mathbf{x}) &= 2.50x_1 + 0.17x_2 + 0.01x_3 - 9.40 \end{aligned}$$

Zbog dijeljenja kovarijacijske matrice iščezli su kvadratni članovi, pa su dobivene hipoteze linearne. Još jednostavniji model dobit ćemo ako uvedemo naivnu pretpostavku o nekoreliranosti varijabli, odnosno ako koristimo dijagonalnu dijeljenu kovarijacijsku matricu:

$$\hat{\boldsymbol{\Sigma}}_{diag} = \begin{pmatrix} 2.8 & 0 & 0 \\ 0 & 9.85 & 0 \\ 0 & 0 & 11.2 \end{pmatrix}.$$

Ovaj model ima 10 parametara. Uvrštavanjem u (3.31) dobivamo hipoteze:

$$\begin{aligned} h_1(\mathbf{x}) &= 0.36x_1 - 0.51x_2 - 2.4 \\ h_2(\mathbf{x}) &= 2.50x_1 + 0.20x_2 + 0.27x_3 - 9.9 \end{aligned}$$

Konačno, model možemo pojednostaviti korištenjem izotropne kovarijacijske matrice, odnosno uprosječivanjem varijanci svih varijabli:

$$\hat{\boldsymbol{\Sigma}}_{iso} = \begin{pmatrix} 7.88 & 0 & 0 \\ 0 & 7.88 & 0 \\ 0 & 0 & 7.88 \end{pmatrix}.$$

Ovaj model ima 8 parametara. Uvrštavanjem u (3.32) dobivamo hipoteze:

$$h_1(\mathbf{x}) = 0.13x_1 - 0.63x_2 - 2.6$$
$$h_2(\mathbf{x}) = 0.89x_1 + 0.25x_2 + 0.38x_3 - 4.4$$

Opisani modeli definiraju različite granice između dviju klasa, pa će neke primjere različito klasificirati. Na primjer, posljednji će model primjer $\mathbf{x} = (1, 3.5, 2)$ klasificirati u klasu $\mathcal{C}_2$, dok će ga složeniji modeli klasificirati u klasu $\mathcal{C}_1$.

Moguće je također izgraditi model koji djelomično kombinira više pretpostavki. Naime, kovarijacijsku matricu klase $\mathcal{C}_j$ možemo napisati kao težinsku kombinaciju triju osnovnih slučajeva:

$$\boldsymbol{\Sigma}'_j = \alpha\sigma^2\mathbf{I} + \beta\boldsymbol{\Sigma} + (1 - \alpha - \beta)\boldsymbol{\Sigma}_j, \qquad \alpha \geqslant 0,\ \beta \geqslant 0,\ 0 \leqslant \alpha + \beta \leqslant 1\,.$$

Za $\alpha = \beta = 0$ dobivamo najopćenitiji slučaj nelinearnog klasifikatora. Za $\alpha = 0$ i $\beta = 1$ kovarijacijska je matrica dijeljena, dok je za $\alpha = 1$ i $\beta = 0$ kovarijacijska matrica dijagonalna; u oba je slučaja klasifikator linearan. Između ovih krajnosti nalazi se čitav niz hibridnih modela. Optimalne vrijednosti hiperparametara α i β treba naravno odrediti unakrsnom provjerom.

Poglavlje 4

Linearni diskriminativni modeli

Diskriminativni modeliraju izravno modeliraju granicu između klasa, za razliku od generativnih modela, poput Bayesovog klasifikatora, koji tu ganicu modeliraju posredno preko zajedničke gustoće vjerojatnosti. Osnovna prednost diskriminativnih modela jest jednostavnost u smislu manjeg broja parametara. Unatoč tome, diskriminativni modeli nerijetko daju bolje rezultate od generativnih. U nastavku ćemo se usredotočiti na linearne diskriminativne modele, dakle modele kod kojih je granica između klasa hiperravnina.

4.1 Poopćeni linearni model

Razmotrimo vezu između linearne regresije i linearnog klasifikacijskog modela. Kod linearne regresije hipotezu smo definirali kao linearnu kombinaciju ulaznih značajki:

$$h(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

gdje je $\mathbf{w} \in \mathbb{R}^n$ **vektor težina**, a w_0 je **pomak** (engl. *bias*). Kod klasifikacije izlazi trebaju biti diskretne vrijednosti, odnosno, općenitije, aposteriorne vjerojatnosti klasa u intervalu $[0, 1]$. To možemo ostvariti tako da poopćimo linearan regresijski model uvođenjem nelinearne funkcije $f(\cdot)$ koja transformira izlaz linearne funkcije:

$$h(\mathbf{x}) = f(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)\,. \tag{4.1}$$

Funkcija f naziva se **aktivacijska funkcija** i njezina je svrha linearnu funkciju "spljoštiti" na jedinični interval. Granice između klasa su hiperravnine, odnosno točke za koje $h(\mathbf{x}) =$ konst., a to ujedno znači $\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 =$ konst. Dakle, granice između klasa linearne su funkcije od $\mathbf{x}$, čak i kada je funkcija f nelinearna. Zbog toga modele opisane s (4.1) nazivamo **poopćeni linearni modeli** (engl. *generalized linear models*).

4.1.1 Geometrija linearnog modela

Razmotrimo opet najjednostavniji slučaj:

$$h(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\,.$$

Primjer $\mathbf{x}$ klasificiramo u $\mathcal{C}_1$ ako $h(\mathbf{x}) \geqslant 0$, a inače u klasu $\mathcal{C}_2$. Granica između klasa je $(n-1)$-dimenzijska hiperravnina definirana jedadžbom $h(\mathbf{x}) = 0$. Hiperravnina dijeli

ulazni prostor u dva poluprostora: regiju $\mathcal{R}_1$ za klasu $\mathcal{C}_1$ i regiju $\mathcal{R}_2$ za klasu $\mathcal{C}_2$. Za dvije točke, $\mathbf{x}_1$ i $\mathbf{x}_2$, koje leže na hiperravnini, vrijedi:

$$h(\mathbf{x}_1) = h(\mathbf{x}_2) = 0 \quad \Rightarrow \quad \mathbf{w}^{\mathrm{T}}(\mathbf{x}_1 - \mathbf{x}_2) = 0$$

iz čega je očito da je $\mathbf{w}$ normala hiperravnine. Nadalje, ako je $\mathbf{x}$ točka na hiperravnini, onda $h(\mathbf{x}) = 0$ i vrijedi:

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0 \quad \Rightarrow \quad \frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}\,.$$

Vrijednost $\mathbf{w}^{\mathrm{T}}\mathbf{x}/\|\mathbf{w}\|$ je skalarna projekcija vektora $\mathbf{x}$ na jedinični vektor $\mathbf{w}/\|\mathbf{w}\|$ i odgovara udaljenosti ravnine od ishodišta. Vidimo dakle da parametar w_0 određuje položaj hiperravnine u prostoru.

Pokažimo još da je vrijednost $h(\mathbf{x})$ proporcionalna predznačenoj udaljenosti d točke $\mathbf{x}$ od hiperaravnine. Neka je $\mathbf{x}$ proizvoljno odabrana točka i neka je $\mathbf{x}_\perp$ njezina ortogonalna projekcija na hiperravninu. Onda imamo:

$$\mathbf{x} = \mathbf{x}_\perp + d\frac{\mathbf{w}}{\|\mathbf{w}\|}\,.$$

Množenjem obiju strana s $\mathbf{w}^{\mathrm{T}}$ i zatim dodavanjem w_0 dobivamo

$$\underbrace{\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0}_{=h(\mathbf{x})} = \underbrace{\mathbf{w}^{\mathrm{T}}\mathbf{x}_\perp + w_0}_{=h(\mathbf{x}_\perp)=0} + d\,\underbrace{\frac{\mathbf{w}^{\mathrm{T}}\mathbf{w}}{\|\mathbf{w}\|}}_{=\|\mathbf{w}\|}$$

iz čega za udaljenost između točke $\mathbf{x}$ i hiperravnine dobivamo

$$d = \frac{h(\mathbf{x})}{\|\mathbf{w}\|}\,. \tag{4.2}$$

Ovi su odnosi prikazani na slici 4.1 za slučaj $n = 2$.

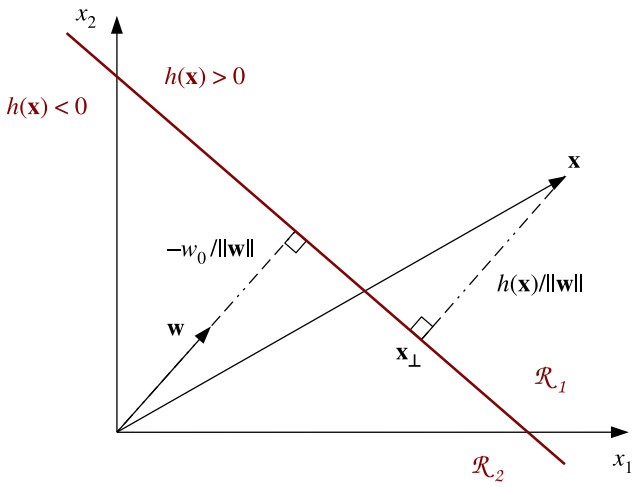
4.1.2 Višeklasna klasifikacija

Kod diskriminativnih modela problemu višeklasne klasifikacije ($K > 2$) može se pristupiti na tri načina.

1. Prvi je da se problem svede na $K - 1$ dvoklasnih klasifikacijskih problema, tako da svaki binarni klasifikator h_j odjeljuje primjere klase $\mathcal{C}_j$ od primjera svih drugih klasa. Tako postavljen klasifikacijski problem nazivamo **jedan-naspram-ostali** (engl. *one-vs-rest*).[1] Primjer klasificiramo u klasu $\mathcal{C}_j$ ako $h_j(\mathbf{x}) \geqslant 0$. Problem nastupa onda kada više binarnih klasifikatora primjer klasificira pozitivno, jer tada ne možemo jednoznačno odrediti klasu. Ovo je prikazano slikom 4.2a za slučaj $K = 3$ klasa.

2. Druga mogućnost je klasifikacija tipa **jedan-naspram-jedan** (engl. *one-vs-one*), kod koje je potrebno naučiti $\binom{K}{2}$ binarnih klasifikatora, po jedan za svaki par klasa. Klasifikator h_{ij} odjeljuje primjere klase $\mathcal{C}_i$ od primjera klase $\mathcal{C}_j$. U slučaju da neki

[1] Također, premda pogrešno: *jedan-naspram-svi* (engl. *one-vs-all*).



Slika 4.1: Geometrija dvodimenzijskog linearnog modela. Odmak pravca od ishodišta određen je pomakom w_0. Predznačena udaljenost točke $\mathbf{x}$ od pravca jednaka je $h(\mathbf{x})/\|\mathbf{w}\|$.

primjer više klasifikatora klasificira pozitivno, klasa dotičnog primjera može se odrediti glasanjem:

$$h(\mathbf{x}) = \underset{\mathcal{C}_i}{\operatorname{argmax}} \sum_{i<j} h_{ij}(\mathbf{x}) \,. \tag{4.3}$$

No i ovdje mogu nastupiti višeznačnosti, kao što prikazuje slika 4.2b. Prednost ovog pristupa jest što može bolje modelirati granicu kod linearno neodvojivih problema, budući da je izglednije da su parovi klasa linearno odvojivi, nego da je svaka klasa linearno odvojiva od svih drugih klasa.

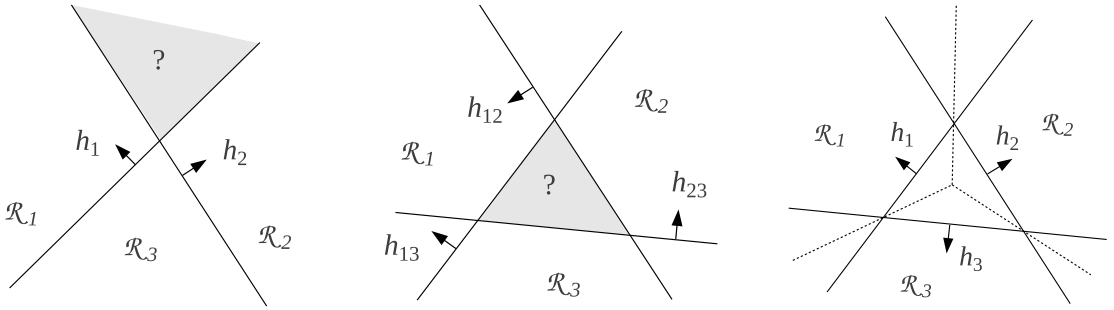
3. Problem višeznačnih regija može se riješiti uporabom K binarnih klasifikatora jedan-naspram-ostali, i zatim klasifikacijom u klasu čija je pouzdanost najveća (slika 4.2c):

$$h(\mathbf{x}) = \underset{\mathcal{C}_j}{\operatorname{argmax}}\, h_j(\mathbf{x}) \,. \tag{4.4}$$

Budući da se svaki višeklasni problem može na ovaj način svesti na binarni skup binarnih klasifikacijskih problema, u nastavku ćemo podrazumijevati da koristimo ovaj pristup i nećemo posebno komentirati višeklasni slučaj. Za slučaj $K = 2$ možemo koristiti ovaj pristup s dva klasifikatora, ili samo jedan klasifikator (rezultat će biti isti).

4.2 Klasifikacija linearnom regresijom

Linearna regresija temeljena na metodi najmanjih kvadrata pokazala se odgovarajućom za mnoge regresijske probleme. Nameće se pitanje može li se isti postupak koristiti za klasifikaciju. Osnovna je ideja naučiti model $h_j(\mathbf{x})$ koji bi za primjere koji pripadaju klasi $\mathcal{C}_j$ davao $h(\mathbf{x}) = 1$, a za sve druge primjere $h(\mathbf{x}) = 0$.



Slika 4.2: Višeklasna klasifikacija za slučaj $K = 3$ ostvarena pomoću binarnih klasifikatora: (a) $K - 1$ klasifikatora jedan-naspram-ostali, (b) $\binom{K}{2}$ klasifikatora jedan-naspram-jedan, (c) K klasifikatora jedan-naspram-ostali.

Razmotrimo najopćenitji slučaj za $n > 2$. Svaka klasa $\mathcal{C}_j$ ima svoj linearan model

$$h_j(\mathbf{x}) = \mathbf{w}_j^{\mathrm{T}}\mathbf{x} + w_{j0} = \tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{x}}$$

gdje $\tilde{\mathbf{w}} = (w_0, \mathbf{w}^{\mathrm{T}})^{\mathrm{T}}$ i $\tilde{\mathbf{x}} = (1, \mathbf{x}^{\mathrm{T}})^{\mathrm{T}}$. Vektor $\tilde{\mathbf{x}}$ je vektor primjera proširen sa značajkom x_0 čija je vrijednost fiksirana na jedinicu. Korištenje proširenih vektora $\tilde{\mathbf{w}}$ i $\tilde{\mathbf{x}}$ pojednostavljuje matematički zapis. Raspolažemo skupom primjera za učenje $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$. Neka je $\tilde{\mathbf{X}}$ matrica primjera dimenzija $N \times (n+1)$ čiji su retci $\tilde{\mathbf{x}}^{(i)}$:

$$\tilde{\mathbf{X}} = \begin{pmatrix} — \; \tilde{\mathbf{x}}^{(1)\mathrm{T}} \; — \\ — \; \tilde{\mathbf{x}}^{(2)\mathrm{T}} \; — \\ \vdots \\ — \; \tilde{\mathbf{x}}^{(N)\mathrm{T}} \; — \end{pmatrix}_{N \times (n+1)}$$

Oznake klasa prikazene su K-dimenzijskim binarnim vektorom $\mathbf{y}^{(i)}$, koji ima jedinicu na mjestu koje odgovara oznaci klase:

$$\mathbf{y}^{(i)} = (y_1^{(i)}, \ldots, y_j^{(i)}, \ldots, y_K^{(i)})^{\mathrm{T}} .$$

Npr. $\mathbf{y}^{(2)} = (0, 0, 1, 0)^{\mathrm{T}}$ znači da primjer $\mathbf{x}^{(2)}$ pripada klasi $\mathcal{C}_3$. Označimo sa $\mathbf{y}_j$ vektor dimenzije N koji sadrži oznake klase $\mathcal{C}_j$ za svaki od N primjera:

$$\mathbf{y}_j = (y_j^{(1)}, \ldots, y_j^{(N)})^{\mathrm{T}} .$$

Npr. $\mathbf{y}_1 = (1, 1, 0, 0, 0)^{\mathrm{T}}$ znači da su prva dva primjera u klasi $\mathcal{C}_1$, a ostali nisu. Prisjetimo se da je kod regresije funkcija gubitka definirana kao kvadratno odstupanje dobivene vrijednosti od ciljne vrijednosti:

$$E(\tilde{\mathbf{w}}_j|\mathcal{D}) = \frac{1}{2}\sum_{i=1}^{N}\left(\tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{x}}^{(i)} - y_j^{(i)}\right)^2 .$$

Isto možemo napisati u matričnom obliku:

$$\begin{aligned} E(\tilde{\mathbf{w}}_j|\mathcal{D}) =& \frac{1}{2}(\tilde{\mathbf{X}}\tilde{\mathbf{w}}_j - \mathbf{y}_j)^{\mathrm{T}}(\tilde{\mathbf{X}}\tilde{\mathbf{w}}_j - \mathbf{y}_j) \\ =& \frac{1}{2}(\tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}}_j - \tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}_j - \mathbf{y}_j^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}} + \mathbf{y}_j^{\mathrm{T}}\mathbf{y}_j) \\ =& \frac{1}{2}(\tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}}_j - 2\mathbf{y}_j^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}} + \mathbf{y}_j^{\mathrm{T}}\mathbf{y}_j) \end{aligned} \tag{4.5}$$

gdje smo iskoristili jednakost

$$\tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}_j = (\tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}_j)^{\mathrm{T}} = \mathbf{y}_j^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}}$$

koja vrijedi jer je rezultat skalarna vrijednost. Deriviranjem po $\tilde{\mathbf{w}}$ i izjednačavanjem s nulom dobivamo:

$$\frac{\mathrm{d}E}{\mathrm{d}\tilde{\mathbf{w}}_j} = \frac{1}{2}\Big(\tilde{\mathbf{w}}_j^{\mathrm{T}}\big(\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}} + (\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}})^{\mathrm{T}}\big) - 2\mathbf{y}_j^{\mathrm{T}}\tilde{\mathbf{X}}\Big) = \tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}}_j - \tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}_j = 0 \tag{4.6}$$

gdje smo iskoristili pravila za deriviranje matrica $\frac{\mathrm{d}}{\mathrm{d}x}x^{\mathrm{T}}Ax = x^{\mathrm{T}}(A + A^{\mathrm{T}})$ i $\frac{\mathrm{d}}{\mathrm{d}x}Ax = A$. Iz (4.6) za vektor težina dobivamo:

$$\tilde{\mathbf{w}}_j = (\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}_j = \tilde{\mathbf{X}}^{+}\mathbf{y}_j\,. \tag{4.7}$$

Matrica je $\tilde{\mathbf{X}}^{+}$ je Moore-Penroseov **pseudoinverz** (poopćeni inverz) matrice $\tilde{\mathbf{X}}$, koja inače nije kvadratna, pa nema inverz. Jedna od čestih uporaba pseudoinverza matrice jest upravo rješavanje sustava linearnih jednadžbi metodom najmanjih kvadrata. U našem slučaju, sustav linearnih jednadžbi je

$$\tilde{\mathbf{X}}\tilde{\mathbf{w}} = \mathbf{y}$$

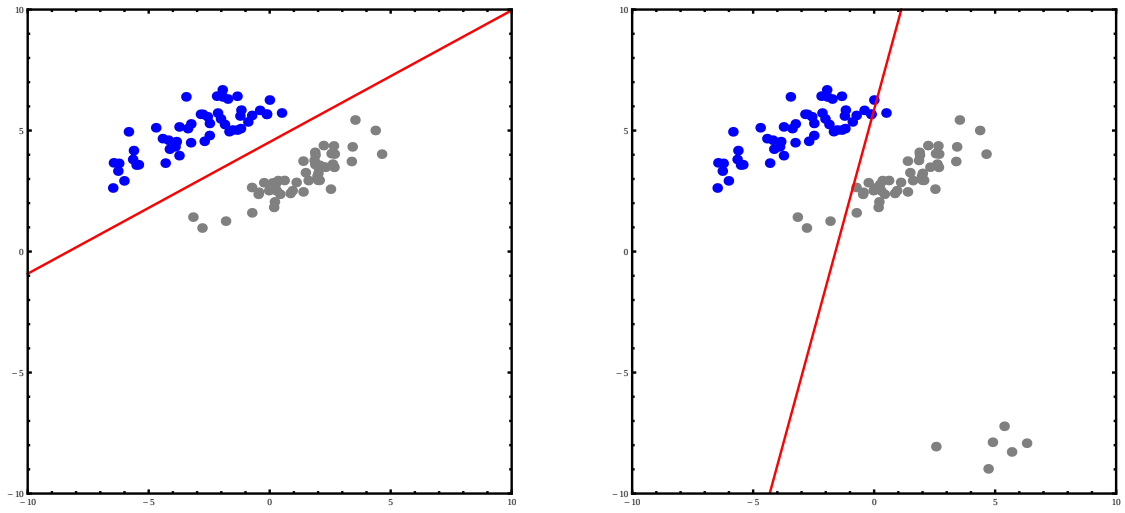
pa je njegovo rješenje u smislu najmanjih kvadrata

$$\tilde{\mathbf{w}} = \tilde{\mathbf{X}}^{+}\mathbf{y}$$

Ovime smo dobili model jedan-naspram-ostali, tj. za svaku klasu imamo zasebnu hipotezu. U slučaju $K = 2$, ne moramo koristiti po jednu hipotezu za svaku klasu, već je dovoljno koristiti samo jednu hipotezu. U tom slučaju granica između dviju klasa bit će hiperravnina $h(\mathbf{x}) = 0.5$. Alternativno, ako za ciljne oznake umjesto vrijednosti $\{0, 1\}$ koristimo vrijednosti $\{-1, 1\}$, granica između dviju klasa bit će hiperravnina $h(\mathbf{x}) = 0$.

Na slici 4.3a prikazana je granica između dviju klasa u prostoru dimenzije $n = 2$ dobivena metodom najmanjih kvadrata. Klase su linearno odvojive i kompaktne, pa je dobivena hipoteza konzistentna s primjerima za učenje.

Klasifikacija temeljena na metodi najmanjih kvadrata ima nekoliko nedostatka. Prvi nedostatak je to što izlazi modela nemaju vjerojatnosnu interpretaciju, budući da vrijednosti hipoteza $h(\mathbf{x}^{(i)})$ nisu ograničene na interval $\{0, 1\}$. Drugi i ozbiljniji nedostatak jest osjetljivost na vrijednosti koje odskaču. To je prikazano na slici 4.3b. Vidimo da već i malen broj primjera koji odskaču ima velik utjecaj na položaj granice, i to unatoč tome što bi ti primjeri ionako bili ispravno klasificirani. Problem je u tome što pogreška definirana izrazom (4.5) kažnjava primjere koji su "suviše točni", odnosno one koji se nalaze na ispravnoj strani granice, ali daleko od nje.



(a) (b)

Slika 4.3: Klasifikacija metodom najmanjih kvadrata za slučaj $n = 2$ dimenzije i $K = 2$ klase: (a) ispravno razdvojene klase, (b) vrijednosti koje odskaču imaju prevelik utjecaj na granicu između klasa.

4.3 Logistička regresija

Logistička regresija je probabilistički diskriminativni model. Unatoč nazivu, nije riječ o regresiji nego o klasifikaciji. Model je diskriminativan, ali – za razliku od diskriminativnih modela koje smo do sada razmotrili – daje izlaz koji ima vjerojatnosno tumačenje. Logistička regresija izravno modelira aposteriornu vjerojatnost $P(\mathcal{C}_j|\mathbf{x})$, dok generativni modeli tu vjerojatnost modeliraju posredno preko zajedničke gustoće $p(\mathbf{x}, \mathcal{C}_j)$.

U nastavku ćemo izvesti model logističke regresije. Pokažimo najprije kako se aposteriorna vjerojatnost $P(\mathcal{C}_j|\mathbf{x})$ može modelirati linearnim modelom. Razmotrimo slučaj klasifikacije u dvije klase, $\mathcal{C}_1$ i $\mathcal{C}_2$. Vrijedi $P(\mathcal{C}_1|\mathbf{x}) = 1 - P(\mathcal{C}_2|\mathbf{x})$. Aposteriornu vjerojatnost klase $\mathcal{C}_1$ možemo napisati kao

$$P(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2)} = \frac{1}{1 + \frac{p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2)}{p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1)}} = \frac{1}{1 + \exp(-\alpha)} = \sigma(\alpha) \quad (4.8)$$

gdje smo uveli

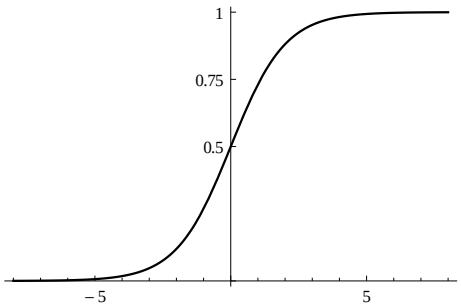
$$\alpha = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2)}\,.$$

Funkcija $\sigma(\alpha)$ je **logistička ili sigmoidalna funkcija**,[2] definirana kao

$$\sigma(\alpha) = \frac{1}{1 + \exp(-\alpha)}\,. \quad (4.9)$$

Logistička funkcija prikazana je na slici 4.4. Funkcija preslikava (možemo reći *spljošćuje*) sve realne brojeve na konačan interval $(0, 1)$. Ova funkcija ima ulogu aktivacijske funkcije

[2] Također: *squashing function*.



Slika 4.4: Logistička ili sigmoidalna funkcija.

f iz poopćenog linearnog modela definiranog s (4.1). Pored toga, njezina je derivacija u analitički vrlo pogodnom obliku:

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\alpha} = \sigma(1-\sigma)\,. \tag{4.10}$$

(Uvjerite se u ovu jednakost.)

Da bismo (4.8) tretirali kao poopćeni linearan model, α trebamo izraziti kao linearnu kombinaciju težina. Vrijednost α je

$$\alpha = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2)} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\,. \tag{4.11}$$

Kao što smo vidjeli u izvodu izraza (3.29), linearnu kombinaciju težina dobit ćemo ako izglednosti klasa $p(\mathbf{x}|\mathcal{C}_1)$ i $p(\mathbf{x}|\mathcal{C}_2)$ modeliramo Gaussovom razdiobom s dijeljenom kovarijacijskom matricom $\boldsymbol{\Sigma}$. U tom slučaju kvadratni se članovi poništavaju i dobivamo linearnu funkciju:

$$\begin{aligned}
\alpha = & \ln \frac{p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2)} = \ln p(x|\mathcal{C}_1)P(\mathcal{C}_1) - \ln p(\mathbf{x}|C_2)P(\mathcal{C}_2) \\
= & + \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \ln P(\mathcal{C}_1) \\
& - \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 - \ln P(\mathcal{C}_2) \\
= & \, \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - \frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln \frac{P(\mathcal{C}_1)}{P(\mathcal{C}_2)}\,.
\end{aligned}$$

Usporedimo li ovaj izraz s (4.11), vidimo da su težine

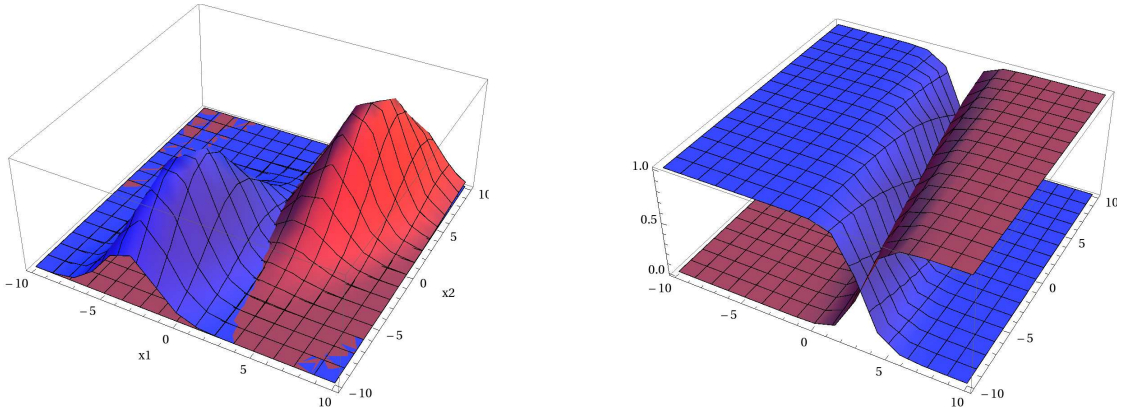
$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{4.12}$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln \frac{P(\mathcal{C}_1)}{P(\mathcal{C}_2)}\,. \tag{4.13}$$

Uvrštavanjem (4.12) i (4.13) u (4.8) konačno dobivamo poopćeni linearan model:

$$h(\mathbf{x}) = P(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0) = \sigma(\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}})\,. \tag{4.14}$$

Iz (4.13) je vidljivo da je pomak granične hiperravnine određen omjerom aposteriornih vjerojatnosti dviju klasa; granica će biti bliža klasi čija je aposteriorna vjerojatnost manja.



(a) (b)

Slika 4.5: Model za dvije klase: (a) zajednička gustoća modelirana generativnim modelom, (b) odgovarajuća asposterirona vjerojatnost modelirana logističkim funkcijama.

Za slučaj K klasa i shemu jedan-naspram-ostali imat ćemo K ovakvih hipoteza. Na slici 4.5 prikazan je model za dvije klase.

Logistička regresija je probabilistički model jer se njegov izlaz može tumačiti kao posteriorna vjerojatnost klase. Treba međutim imati na umu da se ta vjerojatnosna interpretacija temelji na pretpostavci da su klase normalno distribuirane i da imaju dijeljenu varijancu. Naravno, ako to nije slučaj, odnosno ako je naša pretpostavka pogrešna i podatci su distribuirani nekako drugačije, vjerojatnosne procjene neće biti dobre, ali nam barem govore koliko su pojedini primjeri udaljeni od granice. Primijetite da, za razliku od linearnog modela temeljenog na metodi najmanjih kvadrata, ovaj model neće kažnjavati ispravno klasificirane primjere koji se nalaze daleko od granice; za sve takve primjere aposteriorna vjerojatnost bit će blizu jedinice.

Diskriminativni model definiran s (4.14) odgovara dakle generativnom modelu s dvije normalno distribuirane klase i dijeljenom varijancom. Izrazi (4.12) i (4.13) opisuju vezu između parametara diskriminativnog modela i parametara odgovarajućeg generativnog modela. Diskriminativni model ima $n + 1$ parametara (težine i pomak), što je $\mathcal{O}(n)$, dok generativni model ima $n(n + 1)/2 + 2n + 1$ parametara (kovarijacijska matrica, vektori srednjih vrijednosti i apriorna vjerojatnost prve klase), što je $\mathcal{O}(n^2)$. To zorno ilustrira prednost diskriminativnih modela u odnosu na generativne: ukoliko je naš cilj klasifikacija, a ne i modeliranje izglednosti pojedinih klasa niti generiranje primjera, onda nam diskriminativni modeli omogućavaju da to ostvarimo s mnogo manje parametara.

4.3.1 Pogreška unakrsne entropije

Učenje modela logističke regresije svodi se na određivanje parametara $\tilde{\mathbf{w}}$ iz (4.14). Kao i kod drugih algoritama nadziranog učenja, optimizacija parametara svodi se na minimizaciju funkcije pogreške na skupu za učenje. Trebamo dakle najprije definirati funkciju pogreške, a onda optimirati težine $\tilde{\mathbf{w}}$ tako da ta pogreška bude najmanja. Kao i kod generativnih modela, funkciju pogreške možemo definirati kao negativnu log-izglednost na skupu za učenje:

$$E(h|\mathcal{D}) = E(\tilde{\mathbf{w}}|\mathcal{D}) = -\ln \mathcal{L}(\tilde{\mathbf{w}}|\mathcal{D})\,.$$

U tom slučaju minimizacija funkcije pogreške istovjetna je maksimizaciji log-izglednosti.

Pretpostavimo da raspolažemo skupom primjera za učenje $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$. Ako $y^{(i)} = 1$, onda to znači da je primjer $\mathbf{x}^{(i)}$ u klasi $\mathcal{C}_1$. Primijetimo da je, za zadani primjer $\mathbf{x}$, oznaka y zapravo Bernoullijeva slučajna varijabla, te vrijedi $P(y = 1|\mathbf{x}) = P(\mathcal{C}_1|\mathbf{x}) = h(\mathbf{x})$. Razdioba te varijable je

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

Kod generativnih modela modelirali smo izglednost $P(\mathbf{x}|y)$ i vjerojatnost $P(y)$. Kod diskriminativnog modela izravno modeliramo aposteriornu vjerojatnost $P(y|\mathbf{x})$, tj. vjerojatnost oznake za dani primjer. Zato ćemo izglednost izraziti u odnosu na oznake y, a ne u odnosu na primjere $\mathbf{x}$, kao što smo radili kod generativnih modela. Funkcija log-izglednosti parametra $\tilde{\mathbf{w}}$ dakle je

$$\ln \mathcal{L}(\tilde{\mathbf{w}}|\mathcal{D}) = \ln P(\mathcal{D}|\tilde{\mathbf{w}}) = \ln \prod_{i=1}^{N} P(y^{(i)}|\mathbf{x}^{(i)}) = \ln \prod_{i=1}^{N} h(\mathbf{x}^{(i)})^{y^{(i)}} \left(1 - h(\mathbf{x}^{(i)})\right)^{1-y^{(i)}} . \quad (4.15)$$

Izglednost parametara $\tilde{\mathbf{w}}$ bit će to veća što naša hipoteza $h(\mathbf{x})$ ispravnije klasificira primjere, tj. ako daje $h(\mathbf{x}) = 1$ za primjere iz klase $\mathcal{C}_1$ i ako daje $h(\mathbf{x}) = 0$ za primjere koji nisu iz klase $\mathcal{C}_1$. Ili, rečeno drugačije, izglednost $P(\mathcal{D}|\tilde{\mathbf{w}})$ nam kazuje koliko je vjerojatno da naš model primjere u skupu $\mathcal{D}$ klasificira baš onako kako su označeni, ako težine modela postavimo na $\tilde{\mathbf{w}}$.

Sada funkciju pogreške definiramo kao negativnu log-izglednost:

$$E(\tilde{\mathbf{w}}|\mathcal{D}) = -\ln \mathcal{L}(\tilde{\mathbf{w}}|\mathcal{D}) = -\sum_{i=1}^{N} \left\{ y^{(i)} \ln h(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \ln(1 - h(\mathbf{x}^{(i)})) \right\} . \quad (4.16)$$

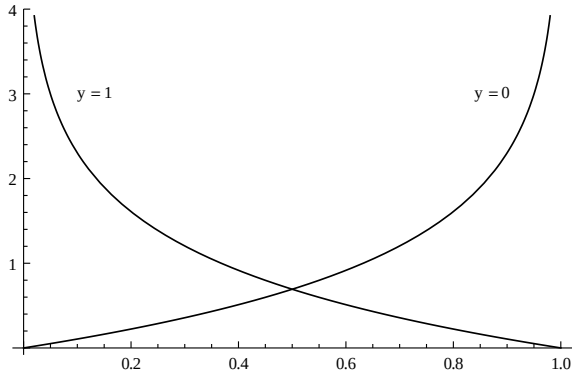
Ovako definiranu pogrešku nazivamo **pogreška unakrsne entropije** (engl. *cross-entropy error*).[3] U idealnom slučaju, kada je svaki primjer ispravno klasificiran, vrijedi $h(\mathbf{x}) = y^{(i)}$ i pogreška je nula. U najgorem slučaju neki je primjer potpuno pogrešno klasificiran uz $h(\mathbf{x}) = 0$ ili $h(\mathbf{x}) = 1$ i tada je pogreška jednaka $+\infty$. Iz (4.16) je vidljivo da je funkcija gubitka u ovom slučaju

$$L(h(\mathbf{x}), y) = -y \ln h(\mathbf{x}) - (1 - y) \ln(1 - h(\mathbf{x})) \quad (4.17)$$

koja je prikazana na slici 4.6. Da sažmemo: maksimizacija log-izglednosti (4.15) odgovara minimizaciji pogreške (4.16), što odgovara minimizaciji empirijske pogreške. Naime, empirijska je pogreška definirana kao očekivanje funkcije gubitka L, pa ako izraz (4.16) podijelimo s brojem primjera N, dobivamo empirijsku pogrešku.

Deriviranje pogreške (4.16) po $\mathbf{w}$ neće nam nažalost dati rješenje u zatvorenoj formi, kao što je to dosad bio slučaj. Moramo dakle koristiti iterativne optimizacijske metode. Najćešće se koristi **gradijentni spust** (engl. *gradient descent*), koji ćemo razmotriti u nastavku. Može se upotrijebiti i bilo koji drugi napredniji postupak iterativne optimizacije, npr. postupci drugog reda kao što su postupak konjugiranih gradijenata ili Newton-Raphsonov postupak, ili pak heurističke metode kao što su genetički algoritmi, optimizacija rojem čestica ili simulirano kaljenje.

[3] Unakrsna entropija definirana je kao $H(p, q) = -\sum_x p(x) \ln q(x)$, uz $0 \ln 0 = 0$.



Slika 4.6: Funkcija gubitka $L(h(\mathbf{x}), y)$ korištena u funkciji unakrsne pogreške.

4.3.2 Gradijentni spust

Gradijentni spust zasniva se na ideji da za funkciju $f(\mathbf{x})$ u točci ekstrema vrijedi $\nabla f(\mathbf{x}) = 0$, dok u ostalim točkama vrijednost gradijenta $\nabla f(\mathbf{x})$ odgovara smjeru porasta funkcije. Krenuvši od neke početno odabrane točke $\mathbf{x}$, minimum funkcije, ako takav postoji, možemo pronaći postepenim ažuriranjem vrijednosti $\mathbf{x}$ u smjeru koji je suprotan gradijentnom vektoru ∇f, sve dok se taj ne izjednači s nulom. Ako je funkcija konveksna, pronađeni minimum ujedno je i globalni minimum. Ako funkcija nije konveksna, postupak može pronaći lokalni optimum umjesto globalnog optimuma. Funkcija pogreške (4.16) jest konveksna, dakle minimum koji pronađemo gradijentnim spustom bit će globalni minimum.

Gradijentni vektor funkcije pogreške je

$$\nabla_{\tilde{\mathbf{w}}} E = \Big(\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_n} \Big)^{\mathrm{T}} .$$

U svakom koraku vektor težina $\tilde{\mathbf{w}}$ ažuriramo u smjeru suprotnom od vektora gradijenta:

$$\tilde{\mathbf{w}} \leftarrow \tilde{\mathbf{w}} - \eta \nabla E(\tilde{\mathbf{w}})$$

gdje je η **faktor/stopa učenja** (engl. *learning rate*) koja određuje iznos pomaka u smjeru suprotnom od vektora gradijenta. Odabir faktora η vrlo je bitan: ako je premalen, konvergencija će biti prespora, a ako je prevelik, postupak može oscilirati ili čak divergirati. Prema tome, premda je funkcija $E(\tilde{\mathbf{w}})$ konveksna, nije zajamčeno da će tijekom optimizacijskog postupka pogreška monotono padati. Jedna mogućnost, koja se u praksi pokazuje dobrom jer ubrava konvergenciju, jest koristiti adaptivni faktor η. Najjednostavnija mogućnost jest krenuti s većim iznosom faktora η, a zatim ga tijekom iteracija postepeno smanjivati.

Izračunajmo sada gradijentni vektor za pogrešku unakrsne entropije definirane s (4.16):

$$\begin{aligned} \nabla E(\tilde{\mathbf{w}}) &= -\sum_{i=1}^{N} \Big(\frac{y^{(i)}}{h(\mathbf{x}^{(i)})} - \frac{1 - y^{(i)}}{1 - h(\mathbf{x}^{(i)})} \Big) h(\mathbf{x}^{(i)}) \big(1 - h(\mathbf{x}^{(i)})\big) \tilde{\mathbf{x}}^{(i)} \\ &= \sum_{i=1}^{N} \underbrace{\Big(h(\mathbf{x}^{(i)}) - y^{(i)} \Big) \tilde{\mathbf{x}}^{(i)}}_{\nabla E_i(\tilde{\mathbf{w}})} \end{aligned} \tag{4.18}$$

pri čemu smo iskoristili (4.10). Derivacija logističke funkcije se poništila, pa smo za gradijent log-izglednosti dobili vrlo jednostavan oblik. Vidimo da svaki primjer gradijentu doprinosi iznosom $\nabla E_i(\tilde{\mathbf{w}})$, koji je proporcionalan razlici između ciljne i dobivene vrijednosti za dotični primjer. Sada možemo napisati i algoritam za učenje modela logističke regresije pomoću gradijentnog spusta:

Algoritam 1. Logistička regresija (gradijentni spust)

$\tilde{\mathbf{w}} \leftarrow (0, 0, \ldots, 0)$
ponavljaj do konvergencije:
  $\Delta\tilde{\mathbf{w}} \leftarrow (0, 0, \ldots, 0)$
  za $i = 1, \ldots, N$
    $h \leftarrow \sigma(\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}^{(i)})$
    $\Delta\tilde{\mathbf{w}} \leftarrow \Delta\tilde{\mathbf{w}} + (h - y^{(i)})\,\tilde{\mathbf{x}}^{(i)}$
  $\tilde{\mathbf{w}} \leftarrow \tilde{\mathbf{w}} - \eta\Delta\tilde{\mathbf{w}}$

Uobičajeni kriteriji za zaustavljanje algoritma su dosezanje unaprijed određenog broja iteracija ili stagnacija u promjeni vrijednosti pogreške ($\|\nabla E(\tilde{\mathbf{w}})\| < \epsilon$). Kriterij zaustavljanja može biti i stagnacija u broju pogrešno klasificiranih primjera, ali tada treba biti oprezan da se algoritam ne zaustavi prerano, posebice ako je faktor η malen. U svakom slučaju, treba osigurati da se se algoritam ne zaustavi prerano, tj. da dobiveni rezultat bude što bliže pravom minimumu. U praksi je uvijek dobro pratiti promjenu iznosa pogreške kroz iteracije algoritma, pa na temelju toga odrediti prikladan kriterij zaustavljanja.

4.3.3 Stohastički gradijentni spust

Kod gore opisanog algoritma gradijentni vektor $\nabla E(\tilde{\mathbf{w}})$ izračunava se skupno za sve primjere iz skupa za učenje. Zbog toga se taj algoritam ponekad naziva i **grupni gradijentni spust** (engl. *batch gradient descent*). Alternativa je **stohastički gradijentni spust** (engl. *stochastic gradient decent*),[4] kod kojeg se ugađanje težina obavlja na temelju svakog primjera pojedinačno, a to se onda ponavlja za svaki primjer iz skupa za učenje. Na taj zapravo dobivamo aproksimaciju stvarnog vektora gradijenta. Za razliku od grupnog gradijentnog spusta, stohastički gradijentni spust u praksi je manje računalno zahtjevan i (kod funkcija koje nisu konveksne) manje podložan zaglavljivanju u lokalnom optimumu.

U nastavku je dan algoritam za učenje logističke regresije pomoću stohastičkog gradijentnog spusta.

4.3.4 Regularizacija

Kod diskriminativnih modela također može doći do prenaučenosti. Prenaučenosti su osobito skloni nelinarni modeli, kod kojih se nelinearnost može suviše prilagoditi skupu za učenje. Linearni modeli su jednostavniji i zbog toga je kod njih opasnost od prenaučenosti manja. Ipak, prenaučenost je moguća i kod linearnih modela, osobito kada je prostor primjera visoke dimenzionalnosti, a skup za učenje malen. U tom slučaju model se može prenaučiti na način da daje preveliku težinu nepotrebnim dimenzijama (značajkama). Dodatno, kod logističke regresije javlja se problem prenaučenosti kod linearno odvojivih

[4] Također: *sequential gradient descent*, *on-line gradient descent*

Algoritam 2. Logistička regresija (stoh. gradijentni spust)

$\tilde{\mathbf{w}} \leftarrow (0, 0, \ldots, 0)$
ponavljaj do konvergencije:
  slučajno permutiraj primjere u $\mathcal{D}$
  za $i = 1, \ldots, N$
    $h \leftarrow \sigma(\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}^{(i)})$
    $\tilde{\mathbf{w}} \leftarrow \tilde{\mathbf{w}} - \eta(h - y^{(i)})\tilde{\mathbf{x}}^{(i)}$

problema. Naime, ako su primjeri za učenje linearno odvojivi, gradijent pogreške nikada neće biti jednak nuli (tj. funkcija pogreške nema minimuma). Posljedično, gradijentni spust neće konvergirati i težine će rasti prema beskonačnosti. Sigmoida će time postajati sve strmija, njezini će izlazi biti sve bliži vrijednostima 0 i 1, pa se tako gubi blagi prijelaz između klasa.

Kao i uvijek, problem prenaučenosti može se riješiti postupkom odabira modela, npr. metodom unakrsne provjere. Međutim, postoji i alternativa, koja se sastoji u tome da se u funkciju pogreške eksplicitno ugradi mjera složenosti modela. Na taj se način zapravo spriječava se da model postane suviše složen, jer će sa složenošću modela rasti njegova ukupna pogreška. Takav postupak, kod kojeg je složenost modela ugrađena u funkciju pogreške, nazivamo **regularizacija**. Regularizacija kombinira minimizaciju empirijskog rizika (empirijske pogreške) i strukturnog rizika (složenosti modela) te omogućava učenje složenih modela na manjim skupovima podataka bez velike opasnosti od prenaučenosti.

Kod logističke regresije, regulariziranu funkciju pogreške možemo definirati kao

$$E(\tilde{\mathbf{w}}|\mathcal{D}) = -\ln \mathcal{L}(\tilde{\mathbf{w}}|\mathcal{D}) + \frac{\lambda}{2}\sum_{j=1}^{n} w_j^2 \tag{4.19}$$

$$-\sum_{i=1}^{N}\left\{y^{(i)}\ln h(\mathbf{x}^{(i)}) + (1-y^{(i)})\ln(1-h(\mathbf{x}^{(i)}))\right\} + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\,. \tag{4.20}$$

Izraz $\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$ je **regularizacijski izraz** (engl. *regularization term*), a faktor λ je **regularizacijski faktor** (engl. *regularization factor*). Složeniji modeli imat će veće apsolutne iznose težina w_j, pa će za takve modele ukupna pogreška biti veća. Što je λ veći, to se više kažnjavaju složeni modeli. Očito, složenost modela koji dobijemo optimizacijom ovisit će o izboru faktora λ. Za $\lambda = 0$ funkcija pogreške degenerira u neregulariziranu pogrešku i model koji dobivamo optimizacijom imat će najmanju empirijsku pogrešku, ali će ujedno biti najsloženiji.

Regularizaciju korištenu u (4.19) nazivamo **L2-regularizacija**. Općenit oblik regularizacijskog izraza je

$$\frac{\lambda}{2}\sum_{j=1}^{n}|w_j|^q\,.$$

Sa $q = 1$ dobivamo L1-regularizaciju, čija je prednost da dovodi do **rijetkih modela** (engl. *sparse models*), odnosno modela kod kojih je većina težina jednaka nuli, što znači da je većina dimenzija zanemarena. Sa $q = 2$ dobivamo L2-regularizaciju, koja ne dovodi do rijetkih modela, ali je analitički pogodna. Kod logističke regresije tipično se koristi L2-regularizacija.

Primijetite da se u regularizacijskome izrazu ne uzima u obzir težina w_0. Ta težina određuje pomak hiperravnine u prostoru, pa nju ne želimo regularizirati, jer bi to značilo da preferiramo da hiperravnina prolazi kroz ishodište (to možemo preferirati samo ako prethodno napravimo centriranje podataka). Zbog toga ćemo u nastavku posebno tretirati pomak w_0 od vektora težina $\mathbf{w}$.

Gradijentni vektor za regulariziranu funkciju pogreške je

$$\begin{aligned}\nabla E(w_0) &= \sum_{i=1}^{N}\Big(h(\mathbf{x}^{(i)}) - y^{(i)}\Big)\\ \nabla E(\mathbf{w}) &= \sum_{i=1}^{N}\Big(h(\mathbf{x}^{(i)}) - y^{(i)}\Big)\mathbf{x}^{(i)} + \lambda\mathbf{w}\,.\end{aligned} \tag{4.21}$$

U svakom koraku gradijentnog spusta vektor težina ugađat ćemo na sljedeći način:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\Big(\sum_{i=1}^{N}\big(h(\mathbf{x}^{(i)}) - y^{(i)}\big)\mathbf{x}^{(i)} + \lambda\mathbf{w}\Big)$$

što možemo napisati kao

$$\mathbf{w} \leftarrow \mathbf{w}(1-\eta\lambda) - \eta\sum_{i=1}^{N}\big(h(\mathbf{x}^{(i)}) - y^{(i)}\big)\mathbf{x}^{(i)}\,.$$

Primijetite da, ako bi drugi pribrojnik bio konstantan, težine bi se u svakom koraku smanivale proporcionalno s $(1-\eta\lambda)$. Takav efekt nazivamo **propadanje težina** (engl. *weight decay*). Također primijetite da promjena težina ovisi ne samo o faktoru η, već i o broju primjera N: što je N veći, to je veća promjena težina. Zbog toga stopu učenja η treba korigirati u ovisnosti o broju primjera. Za stopu učenja može se koristiti i vrijednost η/N, što bismo ionako dobili da smo funkciju pogreške $E(\tilde{\mathbf{w}})$ definirali kao očekivanje funkcije gubitka (4.17).

U nastavku su prikazani algoritmi za učenje regulariziranog modela logističke regresije (postupkom gradijentnog spusta i postupkom stohastičkog gradijentnog spusta).

Algoritam 3. Regularizirana logistička regresija (gradijentni spust)

$\tilde{\mathbf{w}} \leftarrow (0, 0, \ldots, 0)$
ponavljaj do konvergencije:
  $\Delta w_0 \leftarrow 0$
  $\Delta\mathbf{w} \leftarrow (0, 0, \ldots, 0)$
  za $i = 1, \ldots, N$
    $h \leftarrow \sigma(\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}^{(i)})$
    $\Delta w_0 \leftarrow \Delta w_0 + h - y^{(i)}$
    $\Delta\mathbf{w} \leftarrow \Delta\mathbf{w} + (h - y^{(i)})\mathbf{x}^{(i)}$
  $w_0 \leftarrow w_0 - \eta\Delta w_0$
  $\mathbf{w} \leftarrow \mathbf{w}(1-\eta\lambda) - \eta\Delta\mathbf{w}$

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

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

Poglavlje 5

Grupiranje

Do sada smo se bavili problemima nadziranog učenja, odnosno problemima klasifikacije i regresije. U oba ta slučaja na raspolaganju nam je bio skup označenih primjera za učenje, $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}$, kod kojega je svakome primjeru $\mathbf{x}^{(i)}$ pridružena diskretna ili kontinuirana ciljna vrijednosti $y^{(i)}$. U mnogim slučajevima, međutim, skup primjera nije označen, dakle na raspolaganju imamo samo neoznačene primjere, $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$. To može biti zato što je označavanje preskupo, ili pak zato što oznake (odnosno klase) nisu unaprijed poznate i potrebno je najprije otkriti kakve se zakonitosti kriju u podatcima. U takvim slučajevima moramo se osloniti na **nenadzirano učenje** (engl. *unsupervised learning*). Tri tipična zadatka nenadziranog učenja su **grupiranje** (engl. *clustering*) podataka, otkrivanje novih vrijednosti ili vrijednosti koje odskaču (engl. *novelty/outlier detection*) i **smanjenje dimenzionalnosti** (engl. *dimensionality reduction*).

U nastavku se usredotočujemo na grupiranje podataka. Grupiranje podataka je postupak razdjeljivanja primjera u grupe (klastere) primjera, tako da su slični primjeri (slični po nekom svojstvu) svrstani u istu grupu, a različiti primjeri u različite grupe. Svrha grupiranja jest nalaženje "prirodnih" (intrinzičnih) grupa u skupu neoznačenih podataka.

5.1 Vrste grupiranja

Postoje dvije osnovne vrste grupiranja. Kod **particijskog grupiranja** (engl. *partitional clustering*), skup primjera particionira se u grupe sličnih primjera. **Hijerarhijsko grupiranje** (engl. *hierarchical clustering*) skup primjera razdjeljuje u ugniježđene grupe koje sačinjavaju hijerarhiju grupa.

Grupiranje također možemo podijeliti prema "čvrstoći" granica između grupa. Kod **čvrstog grupiranja** (engl. *hard clustering*) svaki primjer može pripadati isključivo jednoj grupi. Kod **mekog grupiranja** (engl. *soft clustering*) jedan primjer može pripadati u više grupa, i to eventualno s različitim stupnjem ili različitom vjerojatnošću pripadanja.

Možemo također govoriti o različitim pristupima s obzirom na kriterij koji se koristi za grupiranje. Kriterij može biti minimizacija kriterijske funkcije (npr. kod algoritma k-srednjih vrijednosti), maksimizacija izglednosti (kod EM-algoritma) ili se grupiranje može raditi prema nekoj funkciji udaljenosti ili mjeri sličnosti (kod hijerarhijskog grupiranja).

5.2 Primjene grupiranja

Primjene grupiranja mogu biti različite, od kojih neke tipične navodimo u nastavku.

- Grupiranje se često koristi za **istraživanje podataka** (engl. *data exploration*), s ciljem pronalaženja skrivene strukture u podatcima. Jednom kada se primjeri grupiraju, dobivene grupe mogu se ručno označiti, središta grupa mogu se tumačiti kao prototipni predstavnici grupa, a za svaku se grupu mogu utvrditi tipični rasponi vrijednosti značajki. To omogućava da se podatci opišu na jednostavniji način, da se uoče pravilnosti i sličnosti u podatcima te da se otkriju odnosi između grupa.

- Grupiranje se može koristiti za **kompresiju podataka**, uključivo za preslikavanje kontinuiranih vrijednosti u diskretne vrijednosti. Npr., boje 24-bitne digitalne slike mogu se grupirati u 256 grupa, a zatim se svaka boja može predstaviti centroidom grupe. Time se ostvaruje kompresija sa 24 na 8 bita po slikovnom elementu. Takav postupak nazivamo **kvantizacija vektora** (engl. *vector quantization*).

- Kod nadziranog učenja, grupiranje se može koristiti kao tehnika **predobrade** (engl. *preprocessing*) s ciljem **smanjenja dimenzionalnosti** (engl. *dimensionality reduction*) prostora primjera, odnosno smanjenja broja značajki. Smanjenje dimenzionalnosti dovodi do ušteda u smislu prostora i vremena izvođenja te smanjuje utjecaj šuma.

- U situaciji kada je u skupu primjera za učenje samo manji dio njih označen, grupiranje se može koristiti u kombinaciji s nadziranim učenjem kako bi se automatski označili svi primjeri za učenje. Osnovna ideja jest da se primjeri (i označeni i neoznačeni) najprije grupiraju, a zatim se neoznačeni primjeri unutar pojedine grupe označavaju prema označenim primjerima koji su se našli u istoj grupi. Pritom je najjednostavnije kao oznaku grupe odabrati onu koja se u grupi najčešće pojavljuje. Ova se tehnika naziva **grupiraj i označi** (engl. *cluster and label*) i tipičan je primjer **polunadziranog učenja** (engl. *semi-supervised learning*).

5.2.1 Smanjenje dimenzionalnosti grupiranjem

Grupiranje se može koristiti za smanjenje dimenzionalnosti prostora primjera. U izvornom prostoru primjera $\mathcal{X}$, primjere možemo prikazati matricom $N \times n$, čiji retci odgovaraju primjerima $\mathbf{x}^{(i)}$, a stupci značajkama $x_1, \ldots, x_n$. Grupiranjem u K grupa, $K < n$, dobivamo matricu smanjenih dimenzija $N' \times K$. Ovu matricu možemo dobiti na dva načina, ovisno o tome što grupiramo.

- Prva mogućnost jest grupiranje redaka matrice, odnosno grupiranje primjera. Grupiranjem N primjera u K grupa ostvarujemo preslikavanje u nov, K-dimenzijski prostor, u kojemu svakom primjeru odgovara vektor čije komponente indiciraju kojoj grupi dotični primjer pripada. Kod čvrstog grupiranja samo je jedna komponenta tog vektora jednaka jedinici, dok su ostale jednake nuli; kod mekog grupiranja više komponenti vektora može biti različito od nule. Posljedično, kod čvrstog grupiranja gubitak informacija bit će razmjerno velik jer će se mnogi primjeri preslikati u istu točku K-dimenzijskog prostora, pa će dobivena matrica imati manje od N redaka. Primijetite da se izvorne značajke koriste samo za grupiranje u n-dimenzijskog prostoru, a nakon toga te se značajke više ne koriste i zamjenjuju se značajkama u

K-dimenzijskom prostoru. Također primijetite da, ako je broj primjera veći od dimenzije izvornog prostora primjera, $N > n$, onda – ovisno o tome kako se provodi grupiranje – dimenzija K novog prostora također može biti veća od n, čime se zapravo ostvaruje povećanje dimenzionalnosti, što može biti korisno u nekim primjenama.

- Drugi način grupiranja jest da se zamijene uloge primjera i značajki: umjesto da grupiramo primjere (tj. retke matrice), grupiramo njihove značajke (tj. stupce matrice). Ovdje nam je cilj u istu grupu smjestiti značajke koje su međusobno slične. Dvije značajke smatramo to sličnijima što se više podudaraju njihove vrijednosti kroz pojedinačne primjere. Smanjenje dimenzionalnosti ostvaruje se zamjenom svih značajki koje su grupirane u zajedničku grupu jednom novom, reprezentativnom značajkom, npr. centroidom grupe. Ovaj pristup naziva se **grupiranje značajki** (engl. *feature clustering*).

5.3 Algoritam k-srednjih vrijednosti

Najjednostavniji i najpoznatiji algoritam grupiranja jest **algoritam k-srednjih vrijednosti** (engl. *k-means algorithm*). Algoritmom se primjeri iz neoznačenog skupa primjera $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ grupiraju u K čvrstih grupa, gdje se parametar K zadaje unaprijed. Svaka grupa predstavljena je svojim centroidom, $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$. Grupiranjem primjera u grupe predstavljene vektorom $\boldsymbol{\mu}_k$ nastaje određena pogreška. Pogrešku grupiranja izražava **kriterijska funkcija**:[1]

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} b_k^{(i)} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|^2 \tag{5.1}$$

gdje je $\|\cdot\|$ **euklidska norma**, tj.

$$\|\mathbf{x} - \boldsymbol{\mu}\|^2 = (\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu})\,.$$

Vrijednost $b_j^{(i)}$ u (5.1) je indikatorska varijabla koja indicira kojoj grupi pripada primjer $\mathbf{x}^{(i)}$: ako $b_k^{(i)} = 1$, onda primjer $\mathbf{x}^{(i)}$ pripada grupi k. Pogreška je jednaka zbroju kvadratnih pogreški, odnosno kvadratima euklidske udaljenosti primjera $\mathbf{x}^{(i)}$ od središta $\boldsymbol{\mu}_k$ grupe u koju su ti primjeri svrstani. Želimo li minimizirati pogrešku J, svaki primjer $\mathbf{x}^{(i)}$ trebamo svrstati u grupu čije je središte $\boldsymbol{\mu}_k$ tom primjeru najbliže, to jest

$$b_k^{(i)} = \begin{cases} 1 & \text{ako } k = \underset{j}{\operatorname{argmin}} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_j\| \\ 0 & \text{inače} \end{cases} \tag{5.2}$$

Očito je da bi bilo kakvo drugačije razvrstavanje primjera u grupe rezultiralo samo još većom pogreškom.

Sada se međutim postavlja pitanje kako odabrati srednje vrijednosti $\boldsymbol{\mu}_k$, a da one minimiziraju pogrešku J. Budući da vrijednosti $b_k^{(i)}$ također ovise o $\boldsymbol{\mu}_k$, optimalne vrijednosti za $\boldsymbol{\mu}_k$ nije moguće izraziti u zatvorenoj formi. Umjesto toga, algoritam k-srednjih vrijednosti optimizaciju provodi iterativno. Algoritam započinje sa slučajno odabranim srednjim vrijednostima $\boldsymbol{\mu}_k$. Zatim se u svakoj iteraciji temeljem (5.2) za svaki primjer $\mathbf{x}^{(i)}$

[1] Također: *ciljna funkcija* (engl. *objective function*), *mjera distorzije* (engl. *distortion measure*) ili (u kontekstu kvantizacije vektora) *pogreška rekonstrukcije*.

izračunava vrijednost $b_k^{(i)}$, odnosno svaki se primjer pridjeljuje grupi čijem je centroidu najbliži. Nakon toga – budući da sad imamo fiksirane vrijednosti $b_k^{(i)}$ – možemo izravno minimizirati pogrešku (5.1). Postavljanjem $\nabla_{\boldsymbol{\mu}_k} J = \mathbf{0}$ i rješavanjem po $\boldsymbol{\mu}_k$ dobivamo:

$$2\sum_{i=1}^{N} b_k^{(i)}(\mathbf{x}^{(i)} - \boldsymbol{\mu}_k) = \mathbf{0}$$

iz čega slijedi

$$\boldsymbol{\mu}_k = \frac{\sum_i b_k^{(i)}\mathbf{x}^{(i)}}{\sum_i b_k^{(i)}}\,. \tag{5.3}$$

Vektor $\boldsymbol{\mu}_k$ jednak je dakle srednjoj vrijednosti vektora svih primjera koji su svrstani u grupu k. Budući da je ovime ostvarena promjena vektora $\boldsymbol{\mu}_k$ u odnosu na njegovu prethodnu vrijednost, sada treba opet primijeniti izraz (5.2) i ponovno izračunati koji primjeri pripadaju grupi k. Ova dva koraka ponavljaju se sve dok se ne dosegne stacionarno stanje, odnosno stanje u kojemu nema daljnjih promjena vrijednosti $\boldsymbol{\mu}_k$.

Algoritam 1. Algoritam k-srednjih vrijednosti

inicijaliziraj centroide $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$ (npr. na slučajno odabrane $\mathbf{x}^{(i)}$)
ponavljaj
  za svaki $\mathbf{x}^{(i)} \in \mathcal{D}$
    $b_k^{(i)} \leftarrow \begin{cases} 1 & \text{ako } k = \operatorname{argmin}_j \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_j\| \\ 0 & \text{inače} \end{cases}$
  za svaki $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$
    $\boldsymbol{\mu}_k \leftarrow \sum_{i=1}^{N} b_k^{(i)}\mathbf{x}^{(i)} / \sum_{i=1}^{N} b_k^{(i)}$
dok $\boldsymbol{\mu}_k$ ne konvergiraju

Razmotrimo računalnu složenost algoritma. Složenost izračuna euklidske udaljenosti je $\mathcal{O}(n)$, gdje je n broj značajki. U prvom koraku (pridjeljivanje primjera grupama) izračunavamo KN udaljenosti, pa je složenost prvog koraka $\mathcal{O}(nNK)$. U drugom koraku (izračun centroida), svaki se primjer pridodaje jednome centroidu (za sve ostale centroide k vrijedi $b_k^{(i)} = 0$), pa je složenost $\mathcal{O}(nN)$. Ukupna vremenska složenost algoritma je linearna po svim parametrima, $\mathcal{O}(TnNK)$, gdje je T broj iteracija. U praksi je broj iteracija T redovito mnogo manji od broja primjera N.

Uočite da algoritam, osim odabira početnih središta, ima dodatan izvor nedeterminističnosti, a to je razrješavanje izjednačenja udaljenosti dvaju primjera od centroida. Pri implementaciji treba voditi računa da se razrješavanje provodi na proizvoljan, ali konzistentan način (u suprotnom se može dogoditi da algoritam zaglavi u beskonačnoj petlji).

Algoritam k-srednjih vrijednosti naziva se također i **Lloydov algoritam**. U kontekstu kvantizacije vektora, algoritam je poznat pod nazivom **algoritam Linde-Buzo-Gray** (LBG).

5.3.1 Odabir početnih središta

Algoritam k-srednjih vrijednosti u stvari pretražuje prostor stanja kojih ima onoliko koliko ima različitih particija od N primjera u K skupova. Nameću su sljedeća pitanja: doseže li algoritam uvijek stacionarno stanje te je li grupiranje koje algoritam nalazi optimalno u

smislu pogreške (5.1)? Lako je dokazati da će algoritam k-srednjih vrijednosti zajamčeno doseći stacionarno stanje. Broj mogućih particija iznosi K^N i konačan je, pa je konačan i broj konfiguracija u kojima se svaki $\boldsymbol{\mu}_k$ nalazi u središtu svoje grupe. Budući da se u svakoj iteraciji pogreška smanjuje, to u svakoj iteraciji algoritam nalazi novu konfiguraciju (nikada ne posjećuje dva puta istu konfiguraciju), a takvih je konfiguracija konačno mnogo, pa se algoritam nužno zaustavlja u konačnom broju koraka.

S druge strane, lako je pokazati da algoritam ne nalazi uvijek optimalno grupiranje. Algoritam k-srednjih vrijednosti je pohlepan i pronalazi lokalno optimalno rješenje. Hoće li to rješenje biti i globalno optimalno, ovisi o izboru početnih srednjih vrijednosti $\boldsymbol{\mu}_k$. Postoje razni načini kako odabrati početne srednje vrijednosti:

- Nasumično odabrati K primjera kao početne vrijednosti $\boldsymbol{\mu}_k$. Ovime se doduše izbjegava postavljanje centroida na mjesta u prostoru primjera u kojemu uopće nema primjera (a koje se lako može dogoditi ako se središta izabiru posve nasumično), ali se ne rješava problem zaglavljivanja u lokalnom optimumu. Problem također predstavljaju primjeri koji odskaču (engl. *outliers*), koji lako mogu završiti u zasebnim grupama. Na prvi pogled možda se čini da je dobro da takvi primjeri završe u zasebnim grupama, ali to nije tako jer je broj grupa K ograničen i one se trebaju poklapati s "prirodnim" (većinskim) grupama koje postoje u podatcima;

- Izračunati srednju vrijednost (centroid) sviju primjera, $\boldsymbol{\mu}$, a zatim srednjoj vrijednosti $\boldsymbol{\mu}$ dodavati manje slučajne vektore i tako dobiti K vektora $\boldsymbol{\mu}_k$. Ovo rješava problem izoliranih primjera, ali ne rješava problem zaglavljivanja u lokalnome optimumu;

- Izračunati prvu glavnu komponentu skupa primjera metodom PCA, razdijeliti raspon na K jedakih intervala, čime se primjeri razdjeljuju u K grupa, i zatim uzeti srednje vrijednosti tih grupa kao početne vrijednosti $\boldsymbol{\mu}_k$;

- Slučajno odabrati jedno početno središte $\boldsymbol{\mu}_k$, a zatim svako iduće središte odabrati tako da je što dalje od ostalih središta. Primjer ovakvog pristupa je algoritam **k-means++**, kod kojega je vjerojatnost da primjer $\mathbf{x}^{(i)}$ bude odabran kao novo središte $\boldsymbol{\mu}_i$ proporcionalna kvadratu udaljenosti tog primjera od njemu najbližeg, već odabranog središta $\boldsymbol{\mu}_k$.

$$P(\boldsymbol{\mu}_i = \mathbf{x}^{(i)}|\mathcal{D}) = \frac{\|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|^2}{\sum_j \|\mathbf{x}^{(j)} - \boldsymbol{\mu}_k\|^2}\,. \tag{5.4}$$

  Premda na ovaj način vrijednosti koje odskaču imaju veću vjerojatnost da budu odabrane za središte, njih je u pravilu manje, pa je ipak vjerojatniji odabir nekog od prosječnih primjera, koji su brojniji. Pokazano je da ovaj način odabira početnih središta znatno smanjuje pogrešku grupiranja, a također ubrzava konvergenciju algoritma.

U slučajevima kada se početna središta određuju nedeterministički, preporuča se algoritam pokrenuti više puta kako bi se dobio rezultat sa što manjom pogreškom grupiranja.

Osim o izboru početnih središta grupa, ishod grupiranja ovisi očito i o odabranom broju grupa, odnosno vrijednosti parametra K. Problem odabira broja grupa zajednički je svim algoritmima grupiranja i razmotrit ćemo ga u odjeljku 5.7.

5.3.2 Algoritam k-medoida

Algoritam k-srednjih vrijednosti upotrebljava euklidsku udaljenost kao mjeru različitosti između primjera i prototipnih vektora. To algoritam čini vrlo osjetljivim na vrijednosti koje odskaču. Algoritam je također ograničen na slučajeve kada je primjere moguće prikazati u vektorskome prostoru, pa je između njih moguće izračunati euklidsku udaljenost i srednju vrijednost (centroid) grupe. Međutim, u nekim slučajevima raspolažemo samo informacijom o međusobnoj sličnosti parova primjera. To je slučaj ako, primjerice, želimo grupirati riječi na temelju sličnosti znakovnih nizova, tako da dobijemo grupe grafijski sličnih riječi, ili grupirati ljude na temelju jakosti njihova poznanstava, tako da dobijemo grupe ljudi koji se međusobno dobro poznaju. U ovakvim situacijama ne raspolažemo mjerom udaljenosti, već **mjerom sličnosti** (engl. *similarity measure*) ili njezinim komplementom, **mjerom različitosti** (engl. *dissimilarity measure*), izračunatom između svih parova primjera.

Algoritam k-medoida poopćenje je algoritma k-srednjih vrijednosti kod kojega je kriterijska funkcija definirana pomoću općenite mjere različitosti $\nu(\mathbf{x}, \mathbf{x}')$ između dvaju primjera:

$$\tilde{J} = \sum_{i=1}^{N} \sum_{k=1}^{K} b_k^{(i)} \nu(\mathbf{x}^{(i)}, \boldsymbol{\mu}_k)\,. \tag{5.5}$$

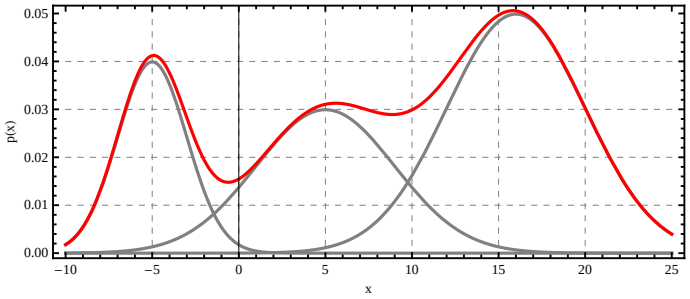
Mjera različitosti ν (odnosno mjera sličnosti) općenitija je od euklidske udaljenosti i od bilo koje druge mjere udaljenosti budući da ne mora ispunjavati uvjete metrike (v. odjeljak 5.5.1).

Kod algoritma k-medoida, prototipe grupa čine medoidi, odnosno reprezentativni primjeri iz skupa $\mathcal{D}$, a ne centroidi. Tipična izvedba je **algoritam PAM** (engl. *partitioning around medoids*).

Algoritam 2. Algoritam PAM

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

Algoritam se također izvršava s po dva koraka u svakoj iteraciji. U prvome koraku, primjeri se svrstavaju u grupu za koju je vrijednost mjere različitosti najmanja. To iziskuje $\mathcal{O}(K(N-K))$ izračuna mjere različitosti ν. U drugom koraku prototipi grupa odabiru se tako da minimiziraju $\tilde{J}$. Za svaku od K grupa, svaki od $N-K$ primjera koji trenutno nisu odabrani kao medoidi zamjenjuje se s trenutno odabranim medoidom, izračunava se zbroj mjere ν između $N-K$ primjera i predloženog medoida te se odabire onaj medoid za koji je ta vrijednost najmanja. To iziskuje $\mathcal{O}(K(N-K)^2)$ izračuna mjere različitosti ν. Posljedično, ukupno algoritam iziskuje $\mathcal{O}(TK(N-K)^2)$ izračuna mjere ν, gdje je T broj iteracija.



Slika 5.1: Mješavina Gaussovih gustoća sačinjena od komponenti $\mathcal{N}(-5, 2)$, $\mathcal{N}(5, 4)$ i $\mathcal{N}(16, 5)$ s koeficijentima $\pi_1 = 0.2$, $\pi_2 = 0.3$ i $\pi_3 = 0.5$.

Visoka vremenska složenost glavni je nedostatak PAM-algoritma, pa su predložena različita poboljšanja, npr. algoritmi **CLARA** (engl. *clustering in large appications*) ili **CLARANS** (engl. *CLARA based upon radnomized search*).

5.4 Model miješane gustoće

Sada ćemo razmotriti probabilistički pristup grupiranju. Za razliku od algoritma k-srednjih vrijednosti, kod kojega su granice između grupa čvrste, kod probabilističkog pristupa primjeri grupama pripadaju s određenom vjerojatnošću. Drugim riječima, jedan te isti primjer može pripadati u dvije ili više grupa, što dovodi do mekih granica između grupa.

Probabilistički pristup grupiranju je generativni i parametarski: pretpostavljamo da se primjeri iz svake grupe pokoravaju nekoj teorijskoj razdiobi. Kod klasifikacije nam je unaprijed bilo poznato koji primjeri pripadaju kojoj klasi, pa smo svaku klasu mogli modelirati zasebnom gustoćom vjerojatnosti (npr. Gaussovom). Kod grupiranja nemamo informaciju o tome koji primjer pripada kojoj klasi i naš je pristup zato drugačiji. Umjesto da izravno modeliramo izglednosti za pojedinačne grupe, rješavat ćemo općenitiji i teži problem modeliranja gustoće vjerojatnosti $p(\mathbf{x})$. Funkcija gustoće $p(\mathbf{x})$ općenito je složenog oblika, pa ćemo koristiti **miješani model** (engl. *mixture model*) sačinjen od linearne kombinacije K osnovnih razdioba. Svaka od tih razdioba odgovara jednoj od K grupa, pa će naš cilj zapravo biti utvrditi parametre tih razdioba i tako ustanoviti s kojom vjerojatnošću primjeri pripadaju pojedinim grupama.

Miješana gustoća (engl. *mixture density*) je linearna kombinacija K funkcija gustoća vjerojatnosti:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \, p(\mathbf{x}|\boldsymbol{\theta}_k) \tag{5.6}$$

gdje su $p(\mathbf{x}|\boldsymbol{\theta}_k)$ **komponente mješavine** (engl. *mixture components*), svaka s parametrima $\boldsymbol{\theta}_k$. Ako se za komponente koriste Gaussove gustoće vjerojatnosti, $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, miješanu gustoću nazivamo **mješavina Gaussovih gustoća** (engl. *mixture of Gaussians*); primjer takve mješavine za jednodimenzijski slučaj dan je na slici 5.1.

Parametri π_k u (5.6) su **koeficijenti mješavine** (engl. *mixture coefficients*). Ako integriramo (marginaliziramo) obje strane izraza (5.6) po $\mathbf{x}$, onda, budući da su pojedinačne komponente normalizirane, mora vrijediti $\sum_{k=1}^{K} \pi_k = 1$. Također, budući da $p(\mathbf{x}|\boldsymbol{\theta}_k) \geqslant 0$

i $p(\mathbf{x}) \geqslant 0$, to mora vrijediti $0 \leqslant \pi_k \leqslant 1$. Vidimo dakle da se koeficijenti π_k mogu tumačiti kao vjerojatnosti. Sukladno pravilima o zbroju i umnošku vjerojatnosti, marginalnu gustoću $p(\mathbf{x})$ možemo izraziti kao

$$p(\mathbf{x}) = \sum_{k=1}^{K} P(\mathcal{G}_k) p(\mathbf{x}|\mathcal{G}_k) \tag{5.7}$$

gdje je $\pi_k = P(\mathcal{G}_k)$ apriorna vjerojatnost odabira komponente k, dok je $p(\mathbf{x}|\boldsymbol{\theta}_k) = p(\mathbf{x}|\mathcal{G}_k)$ gustoća od $\mathbf{x}$ uz odabranu komponentu k. Možemo primijeniti Bayesovo pravilo kako bismo izračunali aposteriorne vjerojatnosti $P(\mathcal{G}_k|\mathbf{x})$:

$$P(\mathcal{G}_k|\mathbf{x}) = \frac{P(\mathcal{G}_k)p(\mathbf{x}|\mathcal{G}_k)}{p(\mathbf{x})} = \frac{P(\mathcal{G}_k)p(\mathbf{x}|\mathcal{G}_k)}{\sum_j P(\mathcal{G}_j)p(\mathbf{x}|\mathcal{G}_j)} = \frac{\pi_k p(\mathbf{x}|\boldsymbol{\theta}_k)}{\sum_j \pi_j p(\mathbf{x}|\boldsymbol{\theta}_j)} \equiv h_k \,. \tag{5.8}$$

Ovu veličinu označavamo s h_k i nazivamo **odgovornost**. Odgovornost h_k iskazuje kolika je vjerojatnost da primjer $\mathbf{x}$ pripada komponenti $\mathcal{G}_k$.

Usporedimo li izraz (5.8) s izrazom za Bayesov klasifikator, možemo zaključiti da je generativni klasifikacijski model u biti miješani model kod kojega grupe $\mathcal{G}_k$ odgovaraju klasama $\mathcal{C}_j$, gustoće komponenti $p(\mathbf{x}|\mathcal{G}_k)$ odgovaraju izglednostima $p(\mathbf{x}|\mathcal{C}_j)$, vjerojatnosti komponenti $P(\mathcal{G}_k)$ odgovaraju apriornim vjerojatnostima klasa $P(\mathcal{C}_j)$, a odgovornost $P(\mathcal{G}_k|\mathbf{x})$ odgovara posteriornoj vjerojatnosti $P(\mathcal{C}_j|\mathbf{x})$. Razlika je međutim u tome što primjeri iz skupa za učenje $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ nisu označeni, pa ne znamo koji primjer pripada kojoj komponenti.

Kao i kod klasifikacije, naš je zadatak odrediti parametre modela

$$\boldsymbol{\theta} = \{P(\mathcal{G}_k), \boldsymbol{\theta}_k\}_{k=1}^K \,.$$

Na primjer, kod mješavine univarijatnih Gaussovih gustoća, parametri koje moramo odrediti su

$$\boldsymbol{\theta} = \{P(\mathcal{G}_k), \mu_k, \sigma^2\}_{k=1}^K$$

odnosno ukupno $3K$ parametara. U općenitijem slučaju viševarijatne Gaussove gustoće, parametri su

$$\boldsymbol{\theta} = \{P(\mathcal{G}_k), \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K \,.$$

a to je ukupno $(\frac{1}{2}n(n+1)+n+1)K$ odnosno $\mathcal{O}(n^2K)$ parametara. Kao i kod generativnih klasifikacijskih modela, i ovdje se možemo odlučiti za jednostavniji model (model s dijeljenom, dijagonalnom ili kovarijacijskom matricom) i tako smanjiti ukupan broj parametara.

Kao i kod klasifikacije, parametre možemo procijeniti metodom najveće izglednosti. No, budući da ne znamo koji primjer pripada kojoj komponenti, procjenu ne možemo raditi zasebno za svaku klasu, pa je procjena parametara sada složenija. Funkcija log-izglednosti za gustoću (5.6) jednaka je

$$\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \ln \prod_{i=1}^{N} p(\mathbf{x}^{(i)}) = \ln \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \, p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) = \sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k \, p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) \,. \tag{5.9}$$

Nažalost, maksimizacija izraza (5.9) nema rješenja u zatvorenoj formi. Problem predstavlja sumacija koja se nalazi između logaritamske funkcije i funkcije gustoće vjerojatnosti. Rješenje zbog toga moramo potražiti u iterativnim metodama. Jedna mogućnost je optimizacija gradijentnom metodom. Češće upotrebljavana je metoda maksimizacije očekivanja, koju opisujemo u nastavku.

5.4.1 Algoritam maksimizacije očekivanja

Algoritam maksimizacije očekivanja (engl. *expectation maximization algorithm*) ili **EM-algoritam** je iterativan optimizacijski postupak za rješavanje problema najveće izglednosti kod modela s **latentnim varijablama**. Latentna varijabla je slučajna varijabla čije realizacije ne opažamo izravno, već o njoj zaključujemo na temelju drugih, opaženih varijabli. Latentna varijabla može biti uvedena samo kao sredstvo apstrakcije, s ciljem pojednostavljenja modela. Također, latentna varijabla može modelirati nešto što je stvarno, no nedostupno, i takvu latentnu varijablu nazivamo **skrivena varijabla**.

Primjer 5.1 (Latentna varijabla) Imamo dva novčića, A i B. Vjerojatnost da bacanjem novčića A dobijemo glavu neka je μ_A, dok je kod novčića B to μ_B. Dakle, $P(A) = \mu_A$ i $P(B) = \mu_B$.

Parametri μ_A i μ_B su nam nepoznati te ih želimo procijeniti na temelju uzorka. Pretpostavimo da smo uzorak dobili tako da smo slučajno odabrali jedan od dvaju novčića, bacali ga 10 puta, te sve to ponavljali ukupno 5 rundi (ukupno dakle imamo 50 bacanja). Neka je $X_i \in \{0, 10\}$ broj koliko smo puta dobili glavu u i-toj rundi. Neka slučajna varijabla $Z_i \in \{A, B\}$ određuje koji od dviju novčića je korišten u i-toj rundi. Naš uzorak $\{(x_i, z_i)\}_{i=1}^5$ je

$$\{(5, B), (9, A), (8, A), (4, B), (7, A)\}.$$

Gornji problem je problem s potpunim podatcima jer nam je poznato sve što nam treba da bismo mogli izračunati procjenu parametara: znamo koliko smo puta u svakoj rundi dobili glavu (x_i) i znamo koji je novčić korišten u kojoj rundi (z_i). Parametare lako možemo procijeniti metodom najveće izglednosti:

$$\hat{\mu}_A = 24/30 = 0.8 \qquad \hat{\mu}_B = 9/20 = 0.45$$

Problem postaje teži kada ne znamo koji je novčić korišten u kojoj rundi, odnosno kada na raspolaganju imamo samo uzorak $\mathbf{X} = \{x_i\}_{i=1}^5$. Varijabla Z_i sada je skrivena te je riječ o problemu s nepotpunim podatcima. U tom slučaju parametre ne možemo izračunati izravno metodom najveće izglednosti.

Veza ovog primjera s problemom grupiranja podataka je sljedeća: svaka runda odgovara jednom neoznačenom primjeru $\mathbf{x}^{(i)}$. Ukupno imamo pet primjera. Primjer može pripadati prvoj grupi (novčić A) ili drugoj grupi (novčić B). Ove su grupe modelirane Bernoullijevom razdiobom s parametrom μ_A odnosno μ_B. Varijabla Z_i određuje kojoj od dviju grupa pripada primjer $\mathbf{x}^{(i)}$, ali je nama ta informacija skrivena.

U nastavku ćemo najprije dati općenitu formulaciju EM-algoritma. Nakon toga pokazat ćemo primjenu algoritma na model miješane gustoće, a zatim konkretno na Gaussovu mješavinu. Treba napomenuti da EM-algoritam ima široku primjenu i da je grupiranje podataka samo jedna od mogućih primjena ovog algoritma.

5.4.2 Općenita formulacija algoritma

Cilj algoritma maksimizacije očekivanja jest nalaženje parametara $\boldsymbol{\theta}$ koji maksimiziraju log-izglednost $\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X})$, gdje su $\mathbf{X}$ podatci odnosno primjeri. Model $p(\mathbf{X}|\boldsymbol{\theta})$ proširujemo skupom latentnih varijabli $\mathbf{Z}$ i radimo sa zajedničkom gustoćom $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$. Marginalna

gustoća $p(\mathbf{X}|\boldsymbol{\theta})$ uvijek se može rekonstruirati marginalizacijom zajedničke gustoće po latentnim varijablama:

$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \,.$$

Skup $\{\mathbf{X}, \mathbf{Z}\}$ nazivamo potpunim, a $\mathbf{X}$ nepotpunim skupom podataka. Analogno, $\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z})$ zovemo **potpuna log-izglednost**, a $\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X})$ zovemo **nepotpuna log-izglednost**. Nepotpuna log-izglednost jednaka je

$$\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}) = \ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \tag{5.10}$$

dok je potpuna log-izglednost jednaka

$$\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z}) = \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \,. \tag{5.11}$$

Važna razlika između izraza (5.10) (optimizacija marginalne gustoće) i (5.11) (optimizacija zajedničke gustoće) jest u tome što u prvome slučaju logaritam djeluje na zbroj gustoća, dok u drugome djeluje izravno na gustoću vjerojatnosti. Ta je razlika ključna: u prvome slučaju analitičko rješenje nije moguće, dok u drugom jest.

Nažalost, budući da nemamo pristup potpunom skupu podataka, ne možemo izravno raditi s potpunom log-izglednošću. Vrijednosti latentnih varijabli $\mathbf{Z}$ su nam nepoznate. Umjesto da izravno radimo s potpunom log-izglednošću, radit ćemo s očekivanjem potpune log-izglednosti, $\mathbb{E}[\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z})]$. Osnovna ideja EM-algoritma jest iterativno ugađati parametre $\boldsymbol{\theta}$ kako bi se maksimiziralo to očekivanje. Može se pokazati da maksimizacija očekivanja potpune log-izglednosti ujedno dovodi do povećanja nepotpune log-izglednosti $\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X})$, što je zapravo naš cilj.

Maksimizacija očekivanja $\mathbb{E}[\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z})]$ ostvaruje se alterniranjem između dva koraka: E-koraka i M-koraka. U **E-koraku** (korak procjene) računamo očekivanje potpune log-izglednosti uz fiksirane trenutne vrijednosti parametara $\boldsymbol{\theta}^{(t)}$. To očekivanje označavamo sa $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ i računamo kao

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &\equiv \mathbb{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)}}[\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z})] \\
&= \mathbb{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)}}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})] \qquad (5.12) \\
&= \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \,. \qquad (5.13)
\end{aligned}$$

Očekivanje smo izrazili po slučajnoj varijabli $\mathbf{Z}$ uz fiksirane varijable $\mathbf{X}$ i parametre $\boldsymbol{\theta}^{(t)}$, tj. po varijabli $\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}$. Vjerojatnost $P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)})$ je aposteriorna vjerojatnost latentne varijable uz trenutne vrijednosti parametara. Nju možemo izračunati (primjenom Bayesovog teorema) jer su nam $\mathbf{X}$ i $\boldsymbol{\theta}^{(t)}$ poznati. Jedino što je u izrazu (5.12) slobodno jesu parametri $\boldsymbol{\theta}$, i to su parametri koje trebamo optimirati.

U **M-koraku** (korak maksimizacije) odabiremo nove parametre $\boldsymbol{\theta}^{(t+1)}$ koji maksimiziraju (5.12):

$$\boldsymbol{\theta}^{(t+1)} = \operatorname*{argmax}_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \,. \tag{5.14}$$

Ovo je manje težak optimizacijski problem od onog od kojeg smo krenuli i u pravilu (ovisno o odabiru parametarskog modela) može se riješiti analitički.

U nastavku je dan pseudokod općenitog EM-algoritma.

Algoritam 3. Općenit EM-algoritam

inicijaliziraj parametre $\boldsymbol{\theta}^{(0)}$
$t \leftarrow 0$
ponavljaj:
  E-korak: izračunaj $P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)})$
  M-korak: $\boldsymbol{\theta}^{(t+1)} \leftarrow \text{argmax}_{\boldsymbol{\theta}}\, \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$
    gdje $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$
  $t \leftarrow t + 1$
do konvergencije (nepotpune) log-izglednosti ili konvergencije parametara

Krenuvši od nekih početno odabranih parametara $\boldsymbol{\theta}^{(0)}$, E-korak i M-korak se izmjenjuju sve dok $\boldsymbol{\theta}^{(t)}$ ne konvergira. Konvergencija je zajamčena budući da algoritam u svakoj iteraciji povećava očekivanje izglednosti. S druge strane, algoritam ne nalazi nužno globalni optimum izglednosti. Zbog toga je važno odabrati dobre početne vrijednosti parametara. Kod grupiranja se taj problem tipično rješava tako da se za inicijalizaciju koristi algoritam k-srednjih vrijednosti: grupiranje u prvih nekoliko koraka provodi se algoritmom k-srednjih vrijednosti (ili algoritmom k-means++), a onda se tako dobivene srednje vrijednosti koriste kao početne vrijednosti EM-algoritma.

5.4.3 Primjena na model miješane gustoće

Primijenimo sada EM-algoritam na optimizaciju parametara modela miješane gustoće. U slučaju modela miješane gustoće, latentne varijable modeliraju izvore primjera, odnosno modeliraju koji primjer pripada kojoj komponenti. Očito, kada bismo unaprijed znali koji primjer pripada kojoj komponenti, kao što je to slučaj kod klasifikacije, ne bismo imali potrebe za takvim varijablama i točno bismo znali koja je to komponenta čije parametre ugađamo.

Neka je $\mathbf{z} = (z_1, \ldots, z_K)$ vektor indikatorskih varijabli, takvih da $z_k = 1$ ako je primjer generiran iz grupe $\mathcal{G}_k$, a inače $z_k = 0$. Ovdje pretpostavljamo da je primjer generiran od samo jedne grupe, ali će se postupak u konačnici svesti na izračun vjerojatnosti pripadanja primjera svakoj od grupa. Vektor $\mathbf{z}$ je u stvari slučajan vektor s k međusobno isključivih stanja. Svako stanje odnosno grupa ima određenu apriornu vjerojatnost

$$P(z_k = 1) = \pi_k \,.$$

Budući da vrijedi $\sum_k z_k = 1$ (sve indikatorske varijable osim jedne su jednake nuli), to za vjerojatnost $P(\mathbf{z})$ vrijedi

$$P(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k} \,. \tag{5.15}$$

Slično možemo izraziti izglednost $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$:

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \prod_{k=1}^{K} p(\mathbf{x}|\boldsymbol{\theta}_k)^{z_k} \,. \tag{5.16}$$

Izglednost $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ nam kazuje koja je vjerojatnost primjera $\mathbf{x}$, ako pretpostavimo da je primjer generiran komponentom koja je predstavljenom vektorom $\mathbf{z}$. Zajedničku gustoću

$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ sada možemo izraziti pomoću (5.15) i (5.16) kao

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = P(\mathbf{z})p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \prod_{k=1}^{K} \pi_k^{z_k} \prod_{k=1}^{K} p(\mathbf{x}|\boldsymbol{\theta}_k)^{z_k} = \prod_{k=1}^{K} \pi_k^{z_k} p(\mathbf{x}|\boldsymbol{\theta}_k)^{z_k} . \tag{5.17}$$

Posljednja jednakost vrijedi jer je $z_k = 0$ za svaki k, osim za jedan i to isti k.

Izrazimo sada potpunu log-izglednost, $\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{Z})$. Raspolažemo skupom neoznačenih primjera $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ i skupom latentnih varijabli $\mathcal{Z} = \{\mathbf{z}^{(i)}\}_{i=1}^N$. Svakom primjeru $\mathbf{x}^{(i)}$ odgovara po jedna latentna varijabla $\mathbf{z}^{(i)}$, budući da različiti primjeri mogu biti generirani od različitih komponenti. Za model (5.17), potpuna log-izglednost je

$$\begin{aligned} \ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{Z}) &= \ln \prod_{i=1}^{N} p(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}|\boldsymbol{\theta}) = \ln \prod_{i=1}^{N} \prod_{i=1}^{K} \pi_k^{z_k^{(i)}} p(\mathbf{x}|\boldsymbol{\theta}_k)^{z_k^{(i)}} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} z_k^{(i)} \Big(\ln \pi_k + \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) \Big) . \end{aligned} \tag{5.18}$$

Usporedimo li izraz (5.18) s izrazom (5.9) za nepotpunu log-izglednost, uočavamo da se izrazi razlikuju po tome što su redoslijed sumacije $\sum_k$ i logaritamska funkcija zamijenjeni.

E-korak

Za E-korak algoritma trebamo izraziti očekivanje $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$. Dobivamo

$$\begin{aligned} \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \mathbb{E}_{\mathcal{Z}|\mathcal{D},\boldsymbol{\theta}^{(t)}}[\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{Z})] \\ &= \mathbb{E}_{\mathcal{Z}|\mathcal{D},\boldsymbol{\theta}^{(t)}} \Big[\sum_{i=1}^{N} \sum_{k=1}^{K} z_k^{(i)} \big(\ln \pi_k + \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) \big) \Big] \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{E}[z_k^{(i)}|\mathcal{D}, \boldsymbol{\theta}^{(t)}] \big(\ln \pi_k + \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) \big) \end{aligned} \tag{5.19}$$

gdje smo iskoristili linearnost očekivanja i činjenicu da drugi član umnoška nije slučajna varijabla (jedino $z_k^{(i)}$ je slučajna varijabla). Sada treba uočiti da za slučajnu varijablu $z_k^{(i)}$ vrijedi

$$\mathbb{E}[z_k^{(i)}|\mathcal{D}, \boldsymbol{\theta}^{(t)}] = \mathbb{E}[z_k^{(i)}|\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(t)}] = P(z_k^{(i)} = 1|\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(t)}) . \tag{5.20}$$

Prva jednakost vrijedi zato što od svih primjera iz $\mathcal{D}$ samo primjer $\mathbf{x}^{(i)}$ uvjetuje varijablu $\mathbf{z}^{(i)}$, a time i varijablu $z_k^{(i)}$. Druga jednakost vrijedi zato što je $z_k^{(i)}$ Bernoullijeva varijabla, za koje općenito vrijedi $\mathbb{E}[z] = P(z = 1)$.

Očekivanje latentne varijable jednako je dakle njezinoj aposteriornoj vjerojatnosti. Nju možemo izračunati primjenom Bayesovog pravila:

$$P(z_k^{(i)} = 1|\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(t)}) = \frac{p(\mathbf{x}^{(i)}|z_k^{(i)} = 1, \boldsymbol{\theta}^{(t)}) P(z_k^{(i)} = 1)}{\sum_{j=1}^{K} p(\mathbf{x}^{(i)}|z_j^{(i)} = 1, \boldsymbol{\theta}^{(t)}) P(z_j^{(i)} = 1)} \tag{5.21}$$

$$= \frac{p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k^{(t)}) \pi_k^t}{\sum_{j=1}^{K} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k^{(t)}) \pi_j^t} \equiv h_k^{(i)} . \tag{5.22}$$

Ovu veličinu već smo bili izrazili u (5.8) i nazvali je **odgovornost**. Odgovornost $h_k^{(i)}$ je vjerojatnost da je primjer $x^{(i)}$ generirala komponenta k. Budući da je to vjerojatnost, dakle broj između 0 i 1, granica između grupa je meka, za razliku od čvrste granice kakvu nalazi algoritam k-srednjih vrijednosti.

Uvrštavanjem $h_k^{(i)}$ u (5.19), za očekivanje potpune log-izglednosti dobivamo

$$\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum_{i=1}^{N}\sum_{k=1}^{K} h_k^{(i)}\Big(\ln \pi_k + \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k)\Big)\,. \tag{5.23}$$

Radi lakšeg izračuna u nastavku, pribrojnike možemo razdvojiti:

$$\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum_{i=1}^{N}\sum_{k=1}^{K} h_k^{(i)} \ln \pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} h_k^{(i)} \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k)\,. \tag{5.24}$$

M-korak

U M-koraku izračunavamo (5.14), odnosno maksimiziramo (5.24) kako bismo dobili nove parametre $\boldsymbol{\theta}^{(t+1)}$. Optimume nalazimo analitički, rješavanjem jednadžbe $\nabla_{\boldsymbol{\theta}}\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = 0$. Primijetite da optimizaciju možemo raditi nezavisno za svaki π_k i za svaki $\boldsymbol{\theta}_k$, budući da se svaki od njih u (5.24) pojavljuje zasebno u linearnoj kombinaciji.

Za određivanje koeficijenata mješavine, $\pi_k^{(t+1)}$, trebamo riješiti $\nabla_{\pi_k}\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = 0$. Drugi pribrojnik u (5.24) je neovisan o π_k, pa ga možemo zanemariti. U obzir moramo uzeti uvjet $\sum_k \pi_k = 1$, pa za nalaženje uvjetnog ekstrema koristimo metodu Lagrangeovih multiplikatora:

$$\nabla_{\pi_k}\Big(\sum_{i=1}^{N}\sum_{k=1}^{K} h_k^{(i)} \ln \pi_k + \lambda\big(\sum_k \pi_k - 1\big)\Big) = 0$$

gdje je λ Lagrangeov multiplikator. Deriviranjem dobivamo

$$\frac{1}{\pi_k}\sum_{i=1}^{N} h_k^{(i)} + \lambda = 0\,. \tag{5.25}$$

Ako obje strane jednadžbe pomnožimo s π_k i zatim lijevu stranu sumiramo po svim k i izjednačimo s nulom (što vrijedi budući da, ako su svi pojedinačni pribrojnici jednaki nuli, onda je i njihov zbroj jednak nuli), dobivamo

$$\sum_{k=1}^{K}\Big(\pi_k\frac{1}{\pi_k}\sum_{n=1}^{N} h_k^{(i)} + \pi_k\lambda\Big) = \sum_{i=1}^{N}\sum_{k=1}^{K} h_k^{(i)} + \sum_{k=1}^{N}\pi_k\lambda = N + \lambda = 0$$

pri čemu smo iskoristili $\sum_k h_k^{(i)} = \sum_k P(z_k^{(i)}|\mathbf{x}^{(i)},\boldsymbol{\theta}) = 1$ (marginalizacija) i $\sum_k \pi_k = 1$. Iz gornje jednadžbe za Lagrangeov multiplikator dobivamo $\lambda = -N$. Uvrštavanjem u (5.25) konačno dobivamo

$$\pi_k^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N} h_k^{(i)}\,. \tag{5.26}$$

Usporedimo li ovaj izraz s izrazom za procjenu apriorne vjerojatnosti klase $P(\mathcal{C}_j)$ kod generativnog klasifikatora, uočavamo da je jedina razlika u tome što se umjesto oznaka klase $y_j^{(i)}$, koje su nam ovdje nepoznate, koriste trenutne procjene odgovornosti $h_k^{(i)}$. Drugim

riječima, umjesto procjene relativnim frekvencijama, procjenu radimo temeljem težinskog zbroja.

Da bismo odredili parametre pojedinih komponenata, $\boldsymbol{\theta}_k^{(t+1)}$, trebamo riješiti jednadžbu $\nabla_{\boldsymbol{\theta}_k} \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = 0$. Prvi pribrojnik u (5.24) ne ovisi o $\boldsymbol{\theta}_k$ i možemo ga zanemariti. Tako dobivamo

$$\nabla_{\boldsymbol{\theta}_k} \sum_{i=1}^{N} \sum_{k=1}^{K} h_k^{(i)} \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) = \nabla_{\boldsymbol{\theta}_k} \sum_{i=1}^{N} h_k^{(i)} \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) = 0\,. \tag{5.27}$$

Rješenje dalje ovisi o tome kako su modelirane komponente mješavine, $p(\mathbf{x}|\boldsymbol{\theta}_k)$.

5.4.4 Primjena na Gaussovu mješavinu

Za mješavinu univarijatnih (jednodimenzijskih) Gaussovih gustoća, uvrštavanjem $p(x|\boldsymbol{\theta}_k) \sim \mathcal{N}(\mu_k, \sigma_k^2)$ u (5.27), dobivamo

$$\nabla_{\boldsymbol{\theta}_k} \sum_{i=1}^{N} h_k^{(i)} \ln \Big(\frac{1}{\sqrt{2\pi}\sigma_k} \exp \Big\{ - \frac{(x^{(i)} - \mu_k)^2}{2\sigma_k^2} \Big\} \Big) = 0 \tag{5.28}$$

gdje $\boldsymbol{\theta}_k = (\mu_k, \sigma_k^2)$. Deriviranjem i rješavanjem po μ_k odnosno σ_k^2, dobivamo:

$$\mu_k^{(t+1)} = \frac{\sum_i h_k^{(i)} x^{(i)}}{\sum_i h_k^{(i)}} \tag{5.29}$$

$$(\sigma^2)_k^{(t+1)} = \frac{\sum_i h_k^{(i)} \big(x^{(i)} - \mu_k^{(t+1)}\big)^2}{\sum_i h_k^{(i)}} \tag{5.30}$$

Slično, za multivarijatnu (višedimenzijsku) Gaussovu gustoću $p(\mathbf{x}|\boldsymbol{\theta}_k) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ dobivamo:

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_i h_k^{(i)} \mathbf{x}^{(i)}}{\sum_i h_k^{(i)}} \tag{5.31}$$

$$\boldsymbol{\Sigma}_k^{(t+1)} = \frac{\sum_i h_k^{(i)} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_k^{(t+1)})(\mathbf{x}^{(i)} - \boldsymbol{\mu}_k^{(t+1)})^{\mathrm{T}}}{\sum_i h_k^{(i)}} \tag{5.32}$$

Opet možemo uočiti da su ove jednadžbe analogne onima za ML-procjenitelje generativnog klasifikacijskog modela, s tom razlikom da se umjesto oznaka $y_j^{(i)}$ koriste trenutne procjene odgovornosti $h_k^{(i)}$.

Pseudokod algoritma dan je u nastavku.

Algoritam 4. EM-algoritam za Gaussovu mješavinu

1: **inicijaliziraj** parametre Gaussove mješavine, $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$

2: **ponavljaj:**

3: **E-korak**: Izračunaj odgovornosti temeljem trenutnih vrijednosti parametara. Za svaki primjer $\mathbf{x}^{(i)} \in \mathcal{D}$ i svaku komponentu $k = 1, \ldots, K$:

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

4: **M-korak**: Izračunaj procjene parametara temeljem trenutnih odgovornosti. Za svaku komponentu $k = 1, \ldots, K$:

$$\boldsymbol{\mu}_k \leftarrow \frac{\sum_i h_k^{(i)} \mathbf{x}^{(i)}}{\sum_i h_k^{(i)}}$$

$$\boldsymbol{\Sigma}_k \leftarrow \frac{\sum_i h_k^{(i)} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_k)(\mathbf{x}^{(i)} - \boldsymbol{\mu}_k)^{\mathrm{T}}}{\sum_i h_k^{(i)}}$$

$$\pi_k \leftarrow \frac{1}{N} \sum_{i=1}^N h_k^{(i)}$$

5: Izračunaj trenutnu vrijednost log-izglednosti

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

6: **do** konvergencije log-izglednosti ili parametara

Usporedimo li EM-algoritam s algoritmom k-srednjih vrijednosti, možemo zaključiti da su algoritmi zapravo vrlo slični. Algoritam k-srednjih vrijednosti provodi čvrsto grupiranje, dok EM-algoritam provodi meko grupiranje. Može se pokazati da je algoritam k-srednjih vrijednosti ustvari poseban slučaj EM-algoritma. Konkretno, ako komponente Gaussove mješavine imaju dijeljenu i izotropnu kovarijacijsku matricu $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$ te ako se vrijednosti $h_k^{(i)}$ zaokružuju na 0 ili 1, dobivamo algoritam k-srednjih vrijednosti.

5.5 Hijerarhijsko grupiranje

Hijerarhijsko grupiranje, za razliku od particijskog, rezultira hijerarhijom grupa. Hijerarhija grupa može se pregledno prikazati **dendrogramom**:[2] stablom kod kojega listovi odgovaraju primjerima, a vodoravne linije odgovaraju povezivanjima na određenoj udaljenosti. Dendrogram se može presjeći na bilo kojoj željenoj udaljenosti, čime se dobivaju grupe kakve bi se dobile particijskim grupiranjem na toj određenoj udaljenosti. Hijerarhijsko grupiranje može biti **aglomerativno** ili **divizivno**. Aglomerativno grupiranje kreće od grupa koje sadrže svaka po samo jedan primjer i zatim postepeno stapa grupe dok sve primjere ne stopi u jednu grupu. Suprotno, divizivno grupiranje kreće od jedne grupe koju postepeno razdjeljuje. Hijerarhijsko grupiranje, za razliku od algoritma k-srednjih vrijednosti i EM-algoritma, nema nekakvu teorijsku osnovu i zapravo je heuristički postupak.

[2] Od grč. *dendron* – stablo, *gramma* – crtež. Često se u literaturi koristi pogrešan naziv *dendogram*.

5.5.1 Funkcija udaljenosti

Hijerarhijsko grupiranje provodi se temeljem funkcije udaljenosti ili mjere sličnosti, s ciljem da se pronađu grupe primjera koji su najsličniji jedan drugome. **Funkcija udaljenosti** (engl. *distance function*) ili **metrika** je funkcija $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ za koju vrijede sljedeća svojstva:

(i) $d(\mathbf{x}^a, \mathbf{x}^b) \geqslant 0$ (nenegativnost),

(ii) $d(\mathbf{x}^a, \mathbf{x}^b) = 0$ ako i samo ako $\mathbf{x}^a = \mathbf{x}^b$ (strogost),

(iii) $d(\mathbf{x}^a, \mathbf{x}^b) = d(\mathbf{x}^b, \mathbf{x}^a)$ (simetričnost),

(iv) $d(\mathbf{x}^a, \mathbf{x}^b) + d(\mathbf{x}^b, \mathbf{x}^c) \geqslant d(\mathbf{x}^a, \mathbf{x}^c)$ (nejednakost trokuta).

Svojstva (i) i (ii) zajedno nazivaju se pozitivna definitnost. Najčešće korištena je **Minkowskijeva udaljenost**, odnosno Minkowskijev razred metrika:

$$d(\mathbf{x}^a, \mathbf{x}^b) = \Big(\sum_{j=1}^{n} (x_j^a - x_j^b)^p \Big)^{1/p}.$$

Za $p = 1$ dobivamo **L1-udaljenost**,[3] a za $p = 2$ euklidsku udaljenost.

Poopćenje euklidske udaljenosti za općenit slučaj koreliranih značajki s različitim varijancama jest **Mahalanobisova udaljenost**:

$$d^2(\mathbf{x}^a, \mathbf{x}^b) = (\mathbf{x}^a - \mathbf{x}^b)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}^a - \mathbf{x}^b)$$

gdje je $\boldsymbol{\Sigma}$ kovarijacijska matrica. Prednost Mahalonbisove udaljenosti nad euklidskom jest to što ne ovisi o razlikama u rasponima vrijednosti pojedinih dimenzija. Euklidska udaljenost sve dimenzije tretira jednako, što nije dobro ako su rasponi vrijednosti vrlo neujednačeni. U načelu, korištenje euklidske udaljenosti opravdano je samo ako je matrica kovarijacije izotropna, $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$, tj. ako su značajke nekorelirane i primjeri su jednoliko raspršeni po svim dimenzijama. Mahalanobisova udaljenost degradira na euklidsku za $\boldsymbol{\Sigma} = \mathbf{I}$, tj. za slučaj sferičnih Gaussovih gustoća s jediničnom varijancom.

5.5.2 *Mjera sličnosti

Općenitiji pojam od udaljenosti je **mjera sličnosti** (engl. *similarity measure*), odnosno njoj komplementarna **mjera različitosti** (engl. *dissimilarity measure*). Udaljenost se može tumačiti kao geometrijska interpretacija sličnosti, odnosno različitosti. Bitna razlika jest što mjera sličnosti (odnosno mjera različitosti) nije metrika. Mjera sličnosti je funkcija $s : \mathcal{X} \times \mathcal{X} \to [0, 1]$ za koju se obično podrazumijeva da zadovoljava sljedeće:

(i) $s(\mathbf{x}, \mathbf{x}) = 1$,

(ii) $0 \leqslant s(\mathbf{x}^a, \mathbf{x}^b) \leqslant 1$,

(iii) $s(\mathbf{x}^a, \mathbf{x}^b) = s(\mathbf{x}^b, \mathbf{x}^a)$.

[3] Također: *Manhattan distance*, *city block distance*, *taxicab distance*.

Iz navedenih je svojstava očigledno da mjera udaljenosti i mjera sličnosti nisu suprotne u smislu da je jedna komplement druge, već su suprotne u smislu da je jednu moguće preslikati u drugu nekom monotono padajućom funkcijom. Na primjer, za preslikavanje udaljenosti d u sličnost s često se koristi

$$s(\mathbf{x}^a, \mathbf{x}^b) = \frac{1}{1 + d(\mathbf{x}^a, \mathbf{x}^b)}\,.$$

Obrnuti smjer, pretvorba sličnosti u udaljenosti, nešto je teži zbog uvjeta nejednakosti trokuta. U većini slučajeva, udaljenosti i sličnosti (odnosno različitosti) mogu se koristiti jednako. U nekim slučajevima međutim sličnosti pružaju dodatnu fleksibilnost.

5.5.3 Hijerarhijsko aglomerativno grupiranje

Algoritam **hijerarhijskog aglomerativnog grupiranja** (engl. *hierarchical agglomerative clustering*, HAC) najčešće je upotrebljavan algoritam hijerarhijskog grupiranja. Algoritam započinje tako da se svaki primjer nalazi u svojoj zasebnoj grupi, a zatim u svakom koraku stapa dvije najbliže grupe, sve dok se ne dosegne unaprijed zadani broj grupa K.

Algoritam 5. Hijerarhijsko aglomerativno grupiranje

inicijaliziraj K, $k \leftarrow N$, $\mathcal{G}_i \leftarrow \{\mathbf{x}^{(i)}\}$ za $i = 1, \ldots, N$
ponavljaj
  $k \leftarrow k - 1$
  $(\mathcal{G}_i, \mathcal{G}_j) \leftarrow \operatorname{argmin}_{\mathcal{G}_a, \mathcal{G}_b} d(\mathcal{G}_a, \mathcal{G}_b)$
  $\mathcal{G}_i \leftarrow \mathcal{G}_i \cup \mathcal{G}_j$
dok je $k > K$

Za $K = 1$ algoritam će rezultirati potpunim dendrogramom koji onda možemo naknadno presijecati na željenim udaljenostima.

Uočite da algoritam u svakom koraku pronalazi par najbližih grupa. Udaljenost između grupa tipično se definira na jedan od sljedeća dva načina:

$$d_{min}(\mathcal{G}_i, \mathcal{G}_j) = \min_{\mathbf{x} \in \mathcal{G}_i, \mathbf{x}' \in \mathcal{G}_j} d(\mathbf{x}, \mathbf{x}') \tag{5.33}$$

$$d_{max}(\mathcal{G}_i, \mathcal{G}_j) = \max_{\mathbf{x} \in \mathcal{G}_i, \mathbf{x}' \in \mathcal{G}_j} d(\mathbf{x}, \mathbf{x}') \tag{5.34}$$

Mjera d_{min} udaljenost između dviju grupa definira kao najmanju udaljenost između pojedinačnih primjera u tim grupama. Takvo grupiranje nazivamo grupiranje temeljem **jednostruke povezanosti** (engl. *single-link clustering*).[4] Tomu suprotna je mjera d_{max}, koja udaljenost između dviju grupa definira kao najveću udaljenost između pojedinačnih primjera u tim grupama. Takvo grupiranje nazivamo grupiranje **potpunom povezanošću** (engl. *complete-link clustering*).[5] Ako su grupe kompaktne i prirodno dobro odvojene, onda ove dvije udaljenosti ne daju značajno različite rezultate. Međutim, ako to nije slučaj, razlike mogu biti značajne. Tada jednostruko povezivanje rezultira dugim, ulančanim grupama, dok potpuno povezivanje rezultira manjim, zbijenijim grupama.

[4] Također: grupiranje metodom najbližeg susjeda (engl. *nearest-neighbor algorithm*).
[5] Također: grupiranje metodom najdaljeg susjeda (engl. *farthest-neighbor algorithm*).

Rezultati ove dvije vrste grupiranja imaju lijepo tumačenje u teoriji grafova. Stapanje dviju grupa, $\mathcal{G}_i$ i $\mathcal{G}_j$, odgovara uvođenju brida između odgovarajućih primjera u tim dvjema grupama. Kod jednostrukog povezivanja, to su dva najbliža primjera iz svake grupe. Budući da se bridovi uvijek uvode između primjera različitih grupa, a nikad između primjera iz iste grupe, rezultirajući graf je stablo (tj. nema ciklusa). Ako $K = 1$, algoritam HAC generira **minimalno razapinjuće stablo** (engl. *minimal spanning tree*) (stablo sa stazom između svaka dva brida kod kojega je ukupan težinski zbroj bridova minimalan). Suprotno, kod potpunog povezivanja, stapanje dviju grupa odgovara uvođenju bridova između svih parova primjera, pa algoritam HAC rezultira **potpuno povezanim grafom**.

Jednostruko i potpuno povezivanje su dva krajnja slučaja izračuna udaljenosti između grupa i dosta su osjetljivi na šum. Kompromisno rješenje je grupiranje temeljem **prosječne povezanosti** (engl. *average-linkage clustering*):

$$d_{avg}(\mathcal{G}_i, \mathcal{G}_j) = \frac{1}{N_i N_j} \sum_{\mathbf{x} \in \mathcal{G}_i} \sum_{\mathbf{x}' \in \mathcal{G}_j} d(\mathbf{x}, \mathbf{x}') \tag{5.35}$$

gdje je N_i odnosno N_j broj primjera u grupi $\mathcal{G}_i$ odnosno $\mathcal{G}_j$.

Alternativu predstavlja mjera koja udaljenost između grupa definira kao udaljenost između njihovih centroida:

$$d_{mean}(\mathcal{G}_i, \mathcal{G}_j) = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| . \tag{5.36}$$

To je računalno najjednostavnija mjera, ali je ograničena na udaljenosti definirane u vektorskome prostoru. Kada udaljenosti nije moguće izračunati u vektorskome prostoru, u prednosti je mjera d_{avg}, koja je primjenjiva na bilo kakvu mjeru sličnosti.

Razmotrimo sada složenost algoritma hijerarhijskog aglomerativnog grupiranja. Algoritam treba izračunati udaljenost između svih $\binom{N}{2}$ parova primjera. Taj se izračun može učiniti jednokratno i pohraniti u tzv. **matricu udaljenosti** (odnosno matricu sličnosti) dimenzija $N \times N$. Prostorna složenost zbog toga je $\mathcal{O}(N^2)$. Tijekom grupiranja, pri svakom stapanju grupa, ova se matrica ažurira i smanjuje (izbacuje se jedan redak i jedan stupac), dok naposlijetku ne dosegne dimenziju $K \times K$. Nakon svakog stapanja potrebno je (ovisno o vrsti povezivanja) izračunati udaljenosti između stopljene grupe i svih preostalih grupa, što iziskuje $\mathcal{O}(N)$ izračuna mjere udaljenosti (ako je riječ o vektorskom prostoru dimenzije n, složenost se može preciznije izraziti kao $\mathcal{O}(nN)$). Nalaženje minimalno ili maksimalno udaljenog para grupa iziskuje $\mathcal{O}(N^2)$ usporedbi u svakom od ukupno $N - K$ koraka algoritma, pa je ukupna vremenska složenost algoritma $\mathcal{O}((N - K)N^2)$. Tipično je $K \ll N$ (za izgradnju potpunog dendrograma vrijedi $K = 1$), pa je vremenska složenost zapravo $\mathcal{O}(N^3)$. Kubna vremenska složenost očit je nedostatak ovog algoritma (za razliku od npr. algoritma k-srednjih vrijednosti čija je složenost u svim parametrima linearna). Postoje implementacije koje umjesto matrice udaljenosti koriste neke druge strukture podataka (npr. prioritetni red), čime je vremensku složenost moguće smanjiti na $\mathcal{O}(N^2 \log N)$. Ipak, kod vrlo velikog broja primjera, kvadratna protorna složenost može već biti nepremostiv problem.

5.6 Predgrupiranje

Kod grupiranja velikog broja primjera, velik nedostatak hijerarhijskog grupiranja predstavlja kvadratna prostorna složenost algoritma. Kod vrlo velikih količina podataka, čak je i linearna složenost algoritama k-srednjih vrijednosti ili EM-algoritma problematična.

Moguće rješenje jest provođenje **predgrupiranja** (**predstrukturiranja**), odnosno razdjeljivanja prostora primjera na particije temeljem nekog računalno jednostavnije postupka. Nakon što se primjeri razdijele u particije, grupiranje se može provesti nekom od složenijih metoda unutar svake takve particije zasebno. Na primjer, ako se N primjera razdijeli u particije s najviše po M primjera, $M \ll N$, prostorna složenost aglomerativnog hijerarhijskog grupiranja reducira se na $\mathcal{O}(M^2)$, a vremenska na $\mathcal{O}((M-K)NM)$, odnosno $\mathcal{O}(NM^2)$ za izgradnju potpunoga dendrograma. Daljnja ubrzanja moguće je ostvariti paralelizacijom.

Primjer takvog postupka je **algoritam krošnji** (engl. *canopy algorithm*). Algoritam prostor primjera razdjeljuje na preklapajuće particije. Za particioniranje se koriste dva praga, T_1 i T_2, gdje $T_1 > T_2$. Algoritam kreće s listom svih primjera te slučajno (ili po nekom odabranome kriteriju) odabire jedan primjer $\mathbf{x}$ kao središte prve krošnje. Zatim za sve preostale primjere $\mathbf{x}'$ računa udaljenost $d(\mathbf{x}, \mathbf{x}')$. Taj izračun može biti aproksimativan; ideja je da ne bude računalno zahtjevan. Primjeri za koje $d(\mathbf{x}, \mathbf{x}') < T_1$ smještaju se pod istu krošnju kao i primjer $\mathbf{x}$, dok se primjeri za koje $d(\mathbf{x}, \mathbf{x}') < T_2$ dodatno uklanjaju iz liste i oni više ne mogu biti izabrani za središta novih krošnji. Postupak se zatim ponavlja za preostale primjere iz liste, sve dok se lista ne isprazni. Uočite da su primjeri koji se nađu u pojasu između T_1 i T_2 pridjeljeni nekoj krošnji, ali se također mogu koristiti kao središta novih krošnji. To rezultira preklapajućim tj. mekim particijama, što algoritam čini robusnijim na šum u podatcima i na pogreške zbog eventualnog korištenja aproksimacije funkcije udaljenosti.

5.7 Provjera grupa

Kod svih je metoda grupiranja broj grupa, odnosno parametar K, potrebno odrediti unaprijed. (Kod hijerarhijskog grupiranja može se odabrati $K = 1$ i izgraditi čitav dendrogram, no onda je naknadno potrebno odlučiti gdje napraviti njegovo presijecanje.) Odabir broja grupa jedan je od glavnih problema kod grupiranja. Idealno, broj grupa odgovarat će broju "prirodnih grupa" u skupu podataka, no on nam je najčešće nepoznat.

Parametar K je zapravo **hiperparametar**: parametar složenosti modela koji se ne ugađa učenjem. Taj parametar ne možemo optimirati na način da minimiziramo kriterijsku funkciju (odnosno, kod probabilistički modela, da maksimiziramo log-izglednost). Razlog je isti kao što, primjerice, parametar K kod algoritma k-NN ne možemo optimirati na temelju empirijske pogreške učenja. Kriterijska funkcija grupiranja monotono opada s porastom K te doseže svoj minimum za $K = N$, ali to očito dovodi do prenaučenosti modela. Odabrati optimalan K znači dakle odabrati optimalnu složenost modela, odnosno onu složenost kod koje je sposobnost generalizacije najveća. Problem je analogan problemu provjere modela kod nadziranog učenje, pa ga nazivamo **provjera (validacija) grupa** (engl. *cluster validation*). Poteškoću predstavlja činjenica da kod grupiranja u pravilu primjeri nisu označeni, pa nije moguće primijeniti metodu unakrsne provjere.

Postoje razni načini kako se može napraviti provjera grupa:

- Kod nekih je primjena, kao npr. kvantizacije boja, broj grupa K unaprijed poznat;
- Broj grupa može se odrediti tako da se, primijenom neke od tehnika redukcije dimenzionalnosti, podatci prikažu dvodimenzijskom prostoru, pa se temeljem toga pokuša odrediti prirodan broj grupa;

- Iterativan pristup: unaprijed definiramo maksimalan dozvoljeni iznos kriterijske funkcije (odnosno minimalni dozvoljeni iznos log-izglednosti) i K postepeno povećavamo sve dok kriterijska funkcija ne padne ispod te vrijednosti;

- Kod nekih je primjena rezultat grupiranja moguće provjeriti ručno te utvrditi ima li grupiranje sa zadanim K smisla;

- Možemo označiti manji podskup primjera, zatim grupirati zajedno označene i neoznačene primjere, a onda napraviti provjeru samo na označenim primjerima. Kao mjeru pogreške treba koristiti neku mjeru koja je temeljena na izračunu gubitaka, npr. **Randov indeks** (engl. *rand index*), mjera **normalizirane uzajamne informacije** (engl. *normalized mutual information*, NMI) ili mjera F_1. U ovakvom kontekstu, treba razlikovati vanjski i unutarnji kriterij grupiranja: mjera F_1 predstavlja **vanjski kriterij**, dok kriterij koji se koristi za samo grupiranje (kriterijska funkcija, log-izglednost ili udaljenost/sličnost) predstavlja **unutarnji kriterij**. Grupiranje se provodi prema unutarnjem kriteriju, za koji pretpostavljamo da je dobro usklađen s vanjskim kriterijem;

- Grafički prikažemo ovisnost kriterijske funkcije o parametru K i tražimo "koljeno" krivulje. S porastom vrijednosti K vrijednost kriterijske funkcije će padati. Kod dovoljno velikog K algoritam će početi razdjeljivati prirodne grupe, pa daljnje smanjenje kriterijske funkcije više neće biti tako značajno. (Ako je algoritam nedeterministički, npr. k-means sa slučajno odabranim početnim središtima, za svaki izbor vrijednosti K treba napraviti više mjerenja.) Slično, kod hijerarhijskog grupiranja možemo prikazati broj dobivenih grupa u ovisnosti o udaljenosti; na mjestima gdje taj broj stagnira (odnosno gdje su razlike između razina dendrograma velike) grupiranje dobro odražava prirodne grupe koje postoje u podatcima;

- Minimiziramo kriterij koji kombinira kriterijsku funkciju i složenost modela te na neki način kažnjavamo modele s prevelikim brojem grupa. Općenit oblik tog kriterija je
$$K^* = \operatorname*{argmin}_K \big(J(K) + \lambda K\big) \tag{5.37}$$
gdje je $J(K)$ vrijednost kriterijske funkcije za model s K grupa, a λ težinski faktor. Veće vrijednosti faktora λ favoriziraju rješenja s manjim brojem grupa. Za $\lambda = 0$ povećanje broja grupa se ne kažnjava i optimalan broj grupa tada je $K^* = N$.

  Odabir parametra λ može se temeljiti na našem iskustvu stečenom na grupiranju sličnih skupova podataka. Alternativa jest da koristimo neki teorijski utemeljen kriterij, poput Akaikeova informacijskog kriterija (AIC):
$$K^* = \operatorname*{argmin}_K \big(-2 \ln \mathcal{L}(K) + 2q(K)\big) \tag{5.38}$$
gdje je $-\ln \mathcal{L}(K)$ negativna log-izglednost podataka za K grupa, a $q(K)$ je broj parametara modela s K grupa. Konkretno, u slučaju grupiranja algoritmom k-srednjih vrijednosti, kriterij AIC svodi se na:
$$K^* = \operatorname*{argmin}_K \big(J(K) + 2nK\big) \tag{5.39}$$
gdje je $J(K)$ vrijednost kriterijske funkcije za model s K grupa, a n je dimenzija prostora primjera. Izraz (5.39) može se izvesti iz (5.38) uzevši u obzir činjenicu

da za model s K grupa vrijedi $q(K) = nK$ i $\ln \mathcal{L}(K) \propto -\frac{1}{2}J(K)$. Posljednje slijedi iz pretpostavke da su podatci generirani Gaussovom mješavinom s uniformnim koeficijentima mješavina te dijeljenom izotropnom kovarijacijskom matricom.

Indeks