Fakultet elektrotehnike i računarstva Sveučilišta u Zagrebu
Strojno učenje, ak. god. 2011/12. – **bilješka za predavanja** (v1.0)

Bilješka 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.

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:

$$h(\mathbf{x}) = f(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)\,. \tag{1}$$

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 (1) nazivamo **poopćeni linearni modeli** (engl. *generalized linear models*).

1.1 Geometrija linearnog modela

Razmotrimo opet najjednostavniji slučaj:

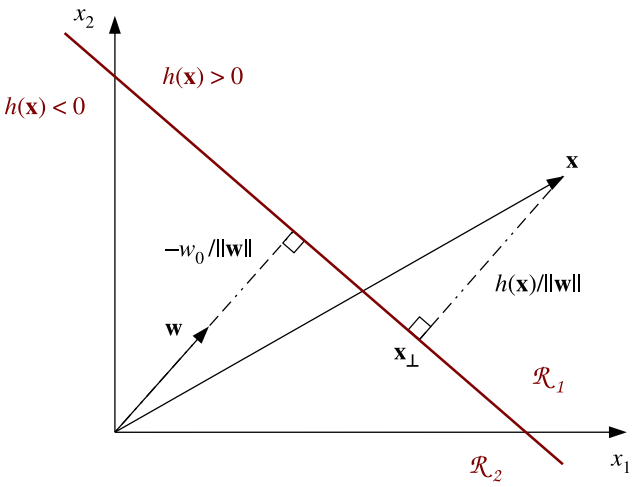
$$h(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\,.$$

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:

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



Slika 1: Geometrija dvodimenzijskog linearnog modela. Odmak pravca od ishodišta određen je pomakom w_0. Predznačena udaljenost točke $\mathbf{x}$ od pravca jednaka je $h(\mathbf{x})/||\mathbf{w}||$.

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

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

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

$$d = \frac{h(\mathbf{x})}{||\mathbf{w}||}\,. \tag{2}$$

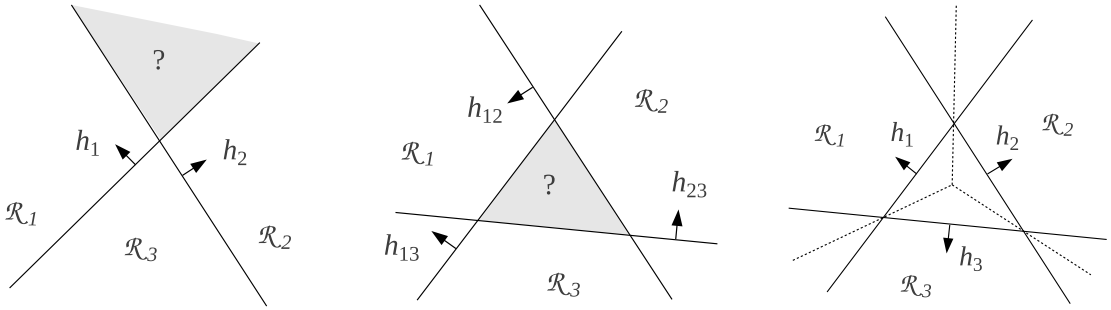
Ovi su odnosi prikazani na slici 1 za slučaj $n = 2$.

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

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



Slika 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.

kada više binarnih klasifikatora primjer klasificira pozitivno, jer tada ne možemo jednoznačno odrediti klasu. Ovo je prikazano slikom 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 primjer više klasifikatora klasificira pozitivno, klasa dotičnog primjera može se odrediti glasanjem:

$$h(\mathbf{x}) = \operatorname*{argmax}_{\mathcal{C}_i} \sum_{i<j} h_{ij}(\mathbf{x})\,. \tag{3}$$

No i ovdje mogu nastupiti višeznačnosti, kao što prikazuje slika 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 2c):

$$h(\mathbf{x}) = \operatorname*{argmax}_{\mathcal{C}_j} h_j(\mathbf{x})\,. \tag{4}$$

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 jednan klasifikator (rezultat će biti isti).

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 $h(\mathbf{x}) = 0$.

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:

$$\begin{aligned} E(\tilde{\mathbf{w}}_j|\mathcal{D}) =& \frac{1}{2}(\tilde{\mathbf{X}}\tilde{\mathbf{w}}_j - \mathbf{y}_j)^{\mathrm{T}}(\tilde{\mathbf{X}}\tilde{\mathbf{w}}_j - \mathbf{y}_j) \\ =& \frac{1}{2}(\tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}}_j - \tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}_j - \mathbf{y}_j^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}} + \mathbf{y}_j^{\mathrm{T}}\mathbf{y}_j) \\ =& \frac{1}{2}(\tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}}_j - 2\mathbf{y}_j^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}} + \mathbf{y}_j^{\mathrm{T}}\mathbf{y}_j) \end{aligned} \tag{5}$$

gdje smo iskoristili jednakost

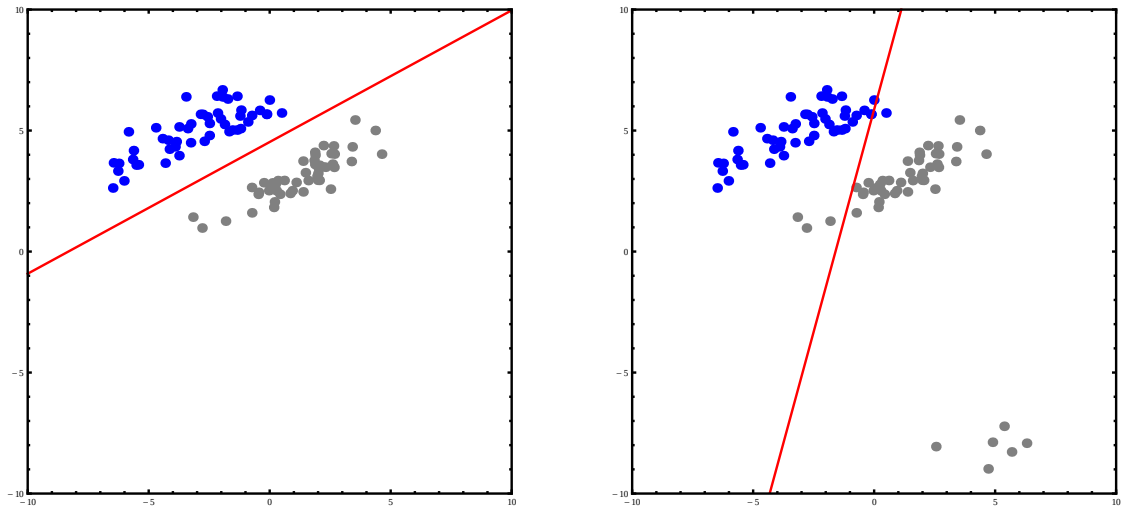
$$\tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}_j = (\tilde{\mathbf{w}}_j^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}_j)^{\mathrm{T}} = \mathbf{y}_j^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}}$$

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

$$\frac{\mathrm{d}E}{\mathrm{d}\tilde{\mathbf{w}}_j} = \frac{1}{2}\Big(\tilde{\mathbf{w}}_j^{\mathrm{T}}\big(\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}} + (\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}})^{\mathrm{T}}\big) - 2\mathbf{y}_j^{\mathrm{T}}\tilde{\mathbf{X}}\Big) = \tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\mathbf{w}}_j - \tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}_j = 0 \tag{6}$$

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 (6) za vektor težina dobivamo:

$$\tilde{\mathbf{w}}_j = (\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y} = \tilde{\mathbf{X}}^{+}\mathbf{y}\,. \tag{7}$$



(a) (b)

Slika 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.

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

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

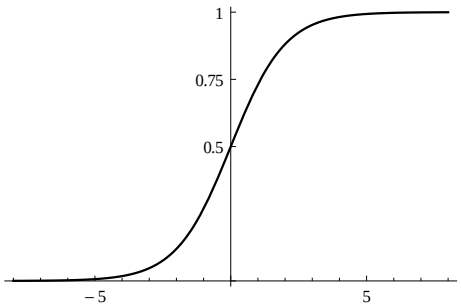
pa je njegovo rješenje u smislu najmanjih kvadrata

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

Ovime smo dobili model jedan-naspram-ostali, tj. za svaku klasu imamo zasebnu hipotezu. U slučaju $K = 2$, granica između klasa definirana je jednadžbom $h_1(\mathbf{x}) = h_2(\mathbf{x})$. Alternativno, možemo koristiti samo jednu hipotezu, no tada za ciljne oznake umjesto vrijednosti $\{0, 1\}$ treba koristiti vrijednosti $\{-1, 1\}$.

Na slici 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 $h(x^{(i)})$ nisu ograničene na interval $\{0, 1\}$. Drugi i ozbiljniji nedostatak jest osjetljivost na vrijednosti koje odskaču. To je prikazano na slici 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 (5) kažnjava primjere koji su "suviše točni", odnosno one koji se nalaze na ispravnoj strani granice, ali daleko od nje.



Slika 4: Logistička ili sigmoidalna funkcija.

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

$$P(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2)} = \frac{1}{1 + \frac{p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2)}{p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1)}} = \frac{1}{1 + \exp(-\alpha)} = \sigma(\alpha) \quad (8)$$

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

$$\sigma(\alpha) = \frac{1}{1 + \exp(-\alpha)} \,. \quad (9)$$

Logistička funkcija prikazana je na slici 4. Funkcija preslikava (možemo reći *spljošćuje*) sve realne brojeve na konačan interval $(0, 1)$. Ova funkcija ima ulogu aktivacijske funkcije f iz poopćenog linearnog modela definiranog s (1). Pored toga, njezina je derivacija u analitički vrlo pogodnom obliku:

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\alpha} = \sigma(1 - \sigma) \,. \quad (10)$$

(Uvjerite se u ovu jednakost.)

Da bismo (8) tretirali kao poopćeni linearan model, α trebamo izraziti kao linearnu kombinaciju težina. Vrijednost α je

$$\alpha = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2)} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 \,. \quad (11)$$

[2] Također: *squashing function.*

Kao što smo vidjeli u bilješci 3, to je ostvarivo ako izglednosti klasa $p(\mathbf{x}|\mathcal{C}_1)$ i $p(\mathbf{x}|\mathcal{C}_2)$ modeliramo Gaussovom razdiobom s dijeljenom kovarijacijskom matricom Σ. U tom slučaju kvadratni se članovi poništavaju i dobivamo linearnu funkciju:

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

Usporedimo li ovaj izraz s (11), vidimo da su težine

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2) \tag{12}$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{P(\mathcal{C}_1)}{P(\mathcal{C}_2)}\,. \tag{13}$$

Uvrštavanjem (12) i (13) u (8) konačno dobivamo poopćeni linearan model:

$$h(\mathbf{x}) = P(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0) = \sigma(\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}})\,. \tag{14}$$

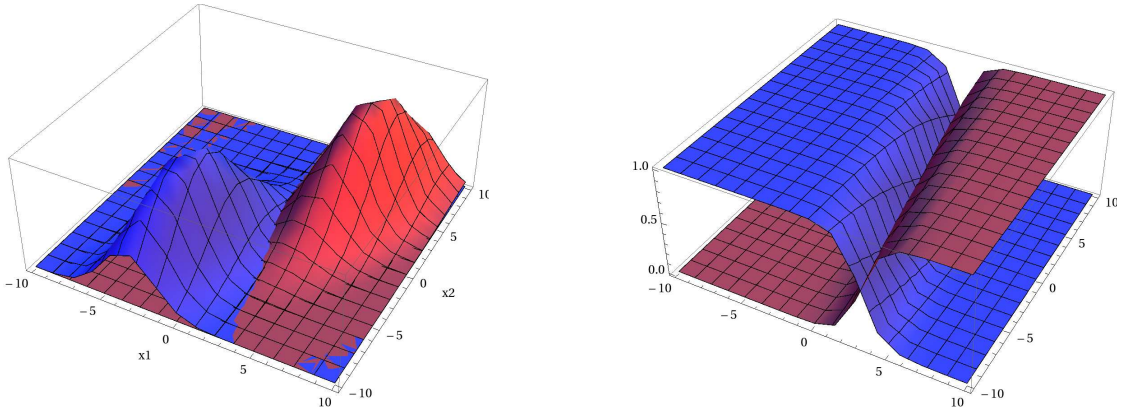
Iz (13) je vidljivo da je pomak granične hiperravnine određen omjerom aposteriornih vjerojatnosti dviju klasa; granica će biti bliža klasi čija je aposteriorna vjerojatnost manja. Za slučaj K klasa i shemu jedan-naspram-ostali imat ćemo K ovakvih hipoteza. Na slici 5 prikazan je model za dvije klase.

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 (14) odgovara dakle generativnom modelu s dvije normalno distribuirane klase i dijeljenom varijancom. Izrazi (12) i (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.

3.1 Pogreška unakrsne entropije

Učenje modela logističke regresije svodi se na određivanje parametara $\tilde{\mathbf{w}}$ iz (14). Kao i kod drugih algoritama nadziranog učenja, optimizacija parametara svodi se na minimizaciju



(a) (b)

Slika 5: Model za dvije klase: (a) zajednička gustoća modelirana generativnim modelom, (b) odgovarajuća asposterirona vjerojatnost modelirana logističkim funkcijama.

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:

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

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

Pretpostavimo da raspolažemo skupom primjera za učenje $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$. Označimo sa $\mathbf{y} = (y^{(1)}, \ldots, y^{(N)})$ vektor dimenzije N koji sadrži oznake klase za svaki od N primjera, pri čemu $y^{(i)} \in \{0, 1\}$. 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

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

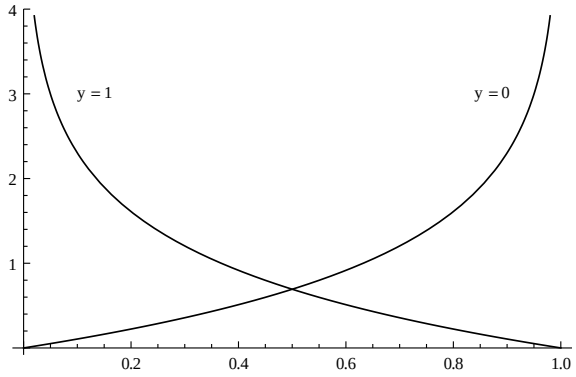
Kod generativnih modela modelirali smo izglednost $P(\mathbf{x}|y)$ i vjerojatnost $P(y)$. Kod diskriminativnog modela izravno modeliramo aposteriornu vjerojatnost $P(y|\mathbf{x})$, tj. vjerojatnost oznake za dani primjer. Zato ćemo izglednost izraziti u odnosu na oznake y, a ne u odnosu na primjere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

$$\mathcal{L}(\tilde{\mathbf{w}}|\mathcal{D}) = \ln P(\mathcal{D}|\tilde{\mathbf{w}}) = \ln \prod_{i=1}^{N} P(y^{(i)}|\mathbf{x}^{(i)}) = \ln \prod_{i=1}^{N} h(\mathbf{x}^{(i)})^{y^{(i)}} (1 - h(\mathbf{x}^{(i)})^{1-y^{(i)}}\,. \tag{15}$$

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:

$$E(\tilde{\mathbf{w}}|\mathcal{D}) = -\mathcal{L}(\tilde{\mathbf{w}}|\mathcal{D}) = -\sum_{i=1}^{N} \left\{ y^{(i)} \ln h(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \ln(1 - h(\mathbf{x}^{(i)})) \right\}. \tag{16}$$



Slika 6: Funkcija gubitka $L(h(\mathbf{x}), y)$ korištena u funkciji unakrsne pogreške.

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 (16) je vidljivo da je funkcija gubitka u ovom slučaju

$$L(h(\mathbf{x}), y) = -y \ln h(\mathbf{x}) - (1 - y) \ln(1 - h(\mathbf{x})) \tag{17}$$

koja je prikazana na slici 6. Da sažmemo: maksimizacija log-izglednosti (15) odgovara minimizaciji pogreške (16), što odgovara minimizaciji empirijske pogreške. Naime, empirijska je pogreška definirana kao očekivanje funkcije gubitka L, pa ako izraz (16) podijelimo s brojem primjera N, dobivamo empirijsku pogrešku.

Deriviranje pogreške (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.

3.2 Gradijentni spust

Gradijentni spust zasniva se na ideji da za funkciju $f(\mathbf{x})$ u točci ekstrema vrijedi $\nabla f(x) = 0$, 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 (16) jest konveksna, dakle minimum koji pronađemo gradijentnim spustom bit će globalni minimum.

Gradijentni vektor funkcije pogreške je

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

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

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 og 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 (16):

$$\begin{aligned}\nabla E(\tilde{\mathbf{w}}) &= -\sum_{i=1}^{N} \Big(\frac{y^{(i)}}{h(\mathbf{x}^{(i)})} - \frac{1-y^{(i)}}{1-h(\mathbf{x}^{(i)})} \Big) h(\mathbf{x}^{(i)}) \big(1 - h(\mathbf{x}^{(i)})\big) \tilde{\mathbf{x}}^{(i)} \\ &= \sum_{i=1}^{N} \underbrace{\Big(h(\mathbf{x}^{(i)}) - y^{(i)} \Big) \tilde{\mathbf{x}}^{(i)}}_{\nabla E_i(\tilde{\mathbf{w}})} \end{aligned} \tag{18}$$

pri čemu smo iskoristili (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:
  $\nabla E \leftarrow (0, 0, \ldots, 0)$
  za $i = 1, \ldots, N$
    $h \leftarrow \sigma(\tilde{\mathbf{w}}^{\mathrm{T}} \tilde{\mathbf{x}}^{(i)})$
    $\nabla E \leftarrow \nabla E + (h - y^{(i)})\, \tilde{\mathbf{x}}^{(i)}$
  $\tilde{\mathbf{w}} \leftarrow \tilde{\mathbf{w}} - \eta \nabla E$
do konvergencije

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.

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 logistička regresije pomoću stohastičkog gradijentnog spusta.

Algoritam 2. Logistička regresija (stoh. gradijentni spust)

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

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 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

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

$$E(\tilde{\mathbf{w}}|\mathcal{D}) = -\ln \mathcal{L}(\tilde{\mathbf{w}}|\mathcal{D}) + \frac{\lambda}{2}\sum_{j=1}^{n} w_j^2 \tag{19}$$

$$-\sum_{i=1}^{N}\Big\{y^{(i)}\ln h(\mathbf{x}^{(i)}) + (1-y^{(i)})\ln(1-h(\mathbf{x}^{(i)}))\Big\} + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\,. \tag{20}$$

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 (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

$$\nabla E(w_0) = \sum_{i=1}^{N}\Big(h(\mathbf{x}^{(i)}) - y^{(i)}\Big)$$

$$\nabla E(\mathbf{w}) = \sum_{i=1}^{N}\Big(h(\mathbf{x}^{(i)}) - y^{(i)}\Big)\mathbf{x}^{(i)} + \lambda\mathbf{w}\,. \tag{21}$$

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 (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:
  $\nabla E_0 \leftarrow 0$
  $\nabla E \leftarrow (0, 0, \ldots, 0)$
  za $i = 1, \ldots, N$
    $h \leftarrow \sigma(\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}^{(i)})$
    $\nabla E_0 \leftarrow \nabla E_0 + h - y^{(i)}$
    $\nabla E \leftarrow \nabla E + (h - y^{(i)})\mathbf{x}^{(i)}$
  $w_0 \leftarrow w_0 - \eta \nabla E_0$
  $\mathbf{w} \leftarrow \mathbf{w}(1 - \eta\lambda) - \eta \nabla E$
do konvergencije

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

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