Strojno učenje

8. Linearni diskriminativni modeli

prof. dr. sc. Bojana Dalbelo Bašić
doc. dr. sc. Jan Šnajder

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Ak. god. 2012/13.

 v1.0

Linearni diskriminativni modeli?

Diskriminativni: Za razliku od generativnih modela

$$P(c_j|x) \propto P(x|c_j) \cdot P(c_j)$$

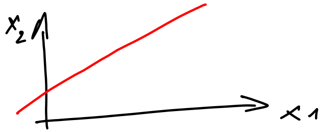
modeliramo

1) aposteriornu vjerojatnost klase $P(c_j|x)$

ili

2) izravno diskriminacijsku funkciju $h(x)$

Linearni: granica (u prostoru značajki) je linearna (hiperravnina)



Danas. . .

1. Poopćeni linearni model
2. Klasifikacija regresijom
3. Gradijentni spust
4. Perceptron

Danas. . .

1. Poopćeni linearni model
2. Klasifikacija regresijom
3. Gradijentni spust
4. Perceptron

Linearan model

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

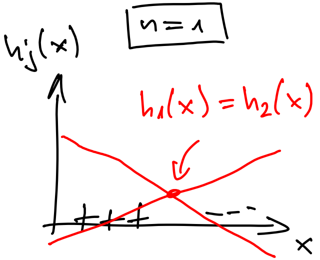
$\tilde{w} = (w_0, w_1 \ldots w_n)$

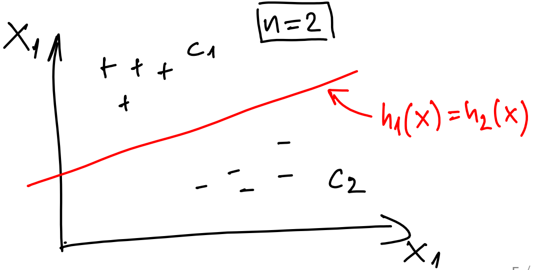
$\tilde{x} = (1, x_1 \ldots x_n)$

Model je linearan u $\mathbf{x}$ $\Rightarrow$ linearna granica u ulaznom prostoru

Granica između dviju klasa $\mathcal{C}_1$ i $\mathcal{C}_2$ zove se diskriminativna funkcija i definirana je jednadžbom:

$$h_1(\mathbf{x}) = h_2(\mathbf{x}) \quad \Rightarrow \quad h_1(\mathbf{x}) - h_2(\mathbf{x}) = h_{12}(\mathbf{x}) = 0$$

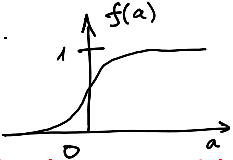




Poopćeni linearni model

Aktivacijska funkcija: nelinearna funkcija $f : \mathbb{R} \to [0,1]$ ili $f : \mathbb{R} \to [-1,1]$

Npr.



(sigmoidalna funkcija)

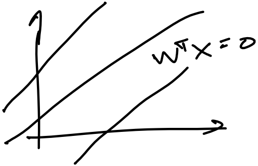
(Bez aktivacijske funkcije: $f(a) = a$)

Poopćeni linearan model (engl. *generalized linear model*, GLM):

$$\boxed{h(\mathbf{x}) = f\left(\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}\right)}$$

$\Rightarrow$ Linearna granica u ulaznom prostoru (premda je f nelinearna)

$\Rightarrow$ Model je nelinearan u parametrima (jer je f nelinearna)



f ne može utjecati na linearnost granice

Poopćeni linearni model

Kao i kod regresije, možemo koristiti preslikavanje $\phi : \mathbb{R}^n \to \mathbb{R}^m$ iz ulaznog prostora u prostor značajki:

$$h(\mathbf{x}) = f\big(\tilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})\big)$$

$\Rightarrow$ Linearna granica u prostoru značajki

$\Rightarrow$ Nelinearna granica u ulaznom prostoru

$\Rightarrow$ Model je nelinearan u parametrima (jer je f nelinearna)

Geometrija linearnog modela

Za točke $\mathbf{x}_1$ i $\mathbf{x}_2$ na hiperravnini:

Pretpostavka BSO: $\phi(x) = \tilde{x}$, $f(a) = a$

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

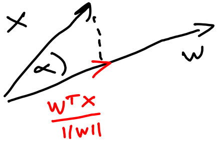
$\Rightarrow$ $\mathbf{w}$ je normala hiperravnine

Projekcija vektora x na vektor w

Za točku $\mathbf{x}$ na hiperravnini:

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0 \; \Big/ \cdot \frac{1}{\|w\|} \quad \Rightarrow \quad \frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}$$

$\Rightarrow$ udaljenost ravnine od ishodišta je $-w_0/\|w\|$

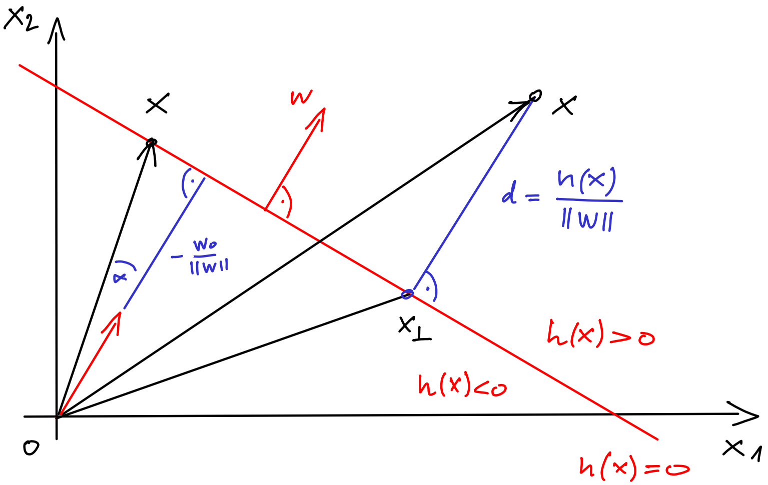


$$w^T x = \langle w, x \rangle = \|w\| \cdot \|x\| \cdot \cos\alpha \; \Big/ \cdot \frac{1}{\|w\|}$$

$$\frac{w^T x}{\|w\|} = \|x\| \cdot \cos\alpha$$

Geometrija linearnog modela

$\mathcal{X} = \mathbb{R}^2$



Geometrija linearnog modela

Za točku $\mathbf{x}$ izvan hiperravnine:

$$\mathbf{x} = \mathbf{x}_\perp + d\frac{\mathbf{w}}{\|\mathbf{w}\|} \quad / \cdot \mathbf{w}^T \quad / + w_0$$

$$\underbrace{\mathbf{w}^\mathrm{T}\mathbf{x} + w_0}_{h(\mathbf{x})} = \underbrace{\mathbf{w}^\mathrm{T}\mathbf{x}_\perp + w_0}_{h(\mathbf{x}_\perp)=0} + d\frac{\mathbf{w}^\mathrm{T}\mathbf{w}}{\cancel{\|\mathbf{w}\|}} \quad = \|\mathbf{w}\|^{\cancel{2}}$$

$$h(\mathbf{x}) = d\|\mathbf{w}\|$$

$\Rightarrow$ udaljenost točke $\mathbf{x}$ od ravnine je $d = h(\mathbf{x})/\|\mathbf{w}\|$ $\Rightarrow$ Predznačena udaljenost

$h(x) > 0$ x je na strani hiperravnine u smjeru normale w

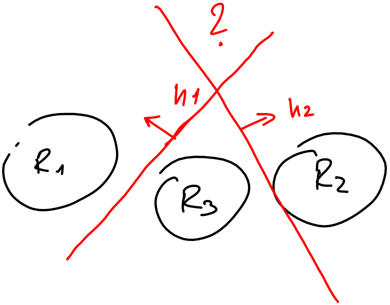
$h(x) < 0$ x je na suprotnoj strani hiperravnine

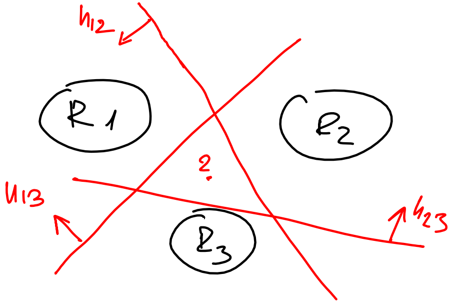
$h(x) = 0$ x je na hiperravnini

Višeklasna klasifikacija ($K > 2$)

(1) jedan-naspram-ostali: $K - 1$ binarnih klasifikatora
$\mathbf{x}$ klasificiramo u $\mathcal{C}_j$ ako $h_j(\mathbf{x}) \geqslant 0$

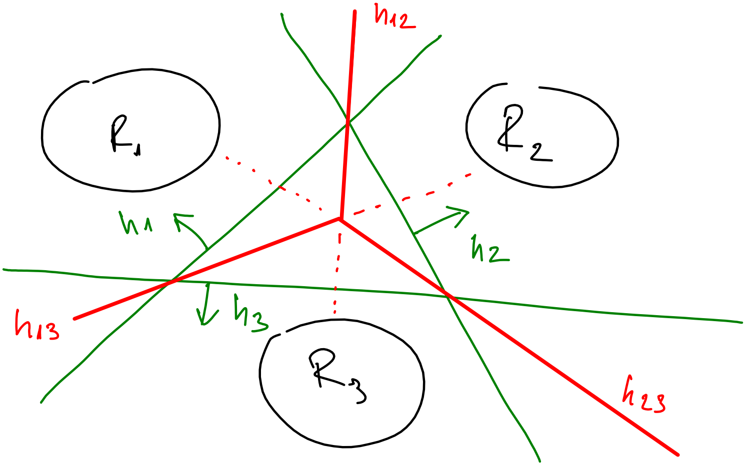
(2) jedan-naspram-jedan: $\binom{K}{2}$ binarnih klasifikatora
$h(\mathbf{x}) = \operatorname{argmax} \sum_{i<j} h_{ij}(\mathbf{x})$





(3) jedan-naspram-ostali: K binarnih klasifikatora s pouzdanošću

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



Danas. . .

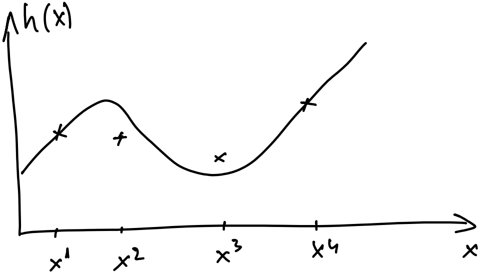
1. Poopćeni linearni model
2. Klasifikacija regresijom
3. Gradijentni spust
4. Perceptron

Regresija – podsjetnik

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

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

$\boxed{n=1}$



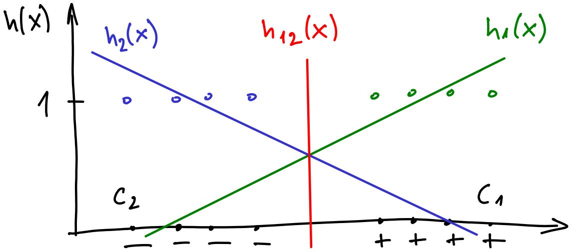
Q: Kako iskoristiti regresiju za klasifikaciju?

Klasifikacija regresijom – primjer

Ideja: regresijska funkcija $h_j(\mathbf{x})$ koji za primjere iz $\mathcal{C}_j$ daje $h_j(\mathbf{x}) = 1$, a za sve druge primjere $h_j(\mathbf{x}) = 0$ (shema jedan-naspram-ostali).

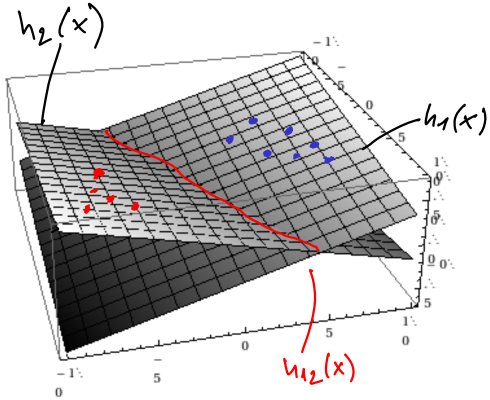
Primjer klasificiriamo u klasu $\mathcal{C}_j$ za koju je $h_j(\mathbf{x})$ najveći. Granica između $\mathcal{C}_1$ i $\mathcal{C}_2$ je tamo gdje $h_1(\mathbf{x}) = h_2(\mathbf{x})$.

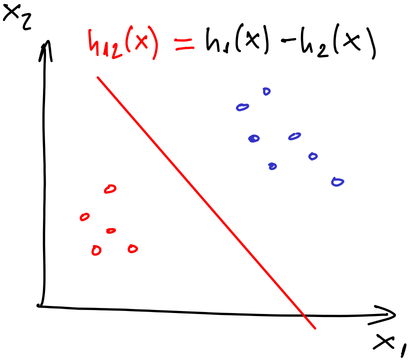
Npr. $n = 1$: $h_j(x) = w_{j0} + w_{j1}x$



(Za $K = 2$ dovoljan nam je jedan model $h(\mathbf{x})$ s granicom $h(\mathbf{x}) = 0.5$.)

Npr. $n = 2$: $h_j(\mathbf{x}) = w_{j0} + w_{j1}x_1 + w_{j2}x_2$





Klasifikacija regresijom

Model za svaku klasu:

c_j $$h_j(\mathbf{x}) = \tilde{\mathbf{w}}_j^{\mathrm{T}} \phi(\mathbf{x})$$

klasa primjera x^i

Primjeri za učenje $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^N$:

$$y^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Dizajn matrica:

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi(\mathbf{x}^{(1)})^{\mathrm{T}} \\ \phi(\mathbf{x}^{(2)})^{\mathrm{T}} \\ \vdots \\ \phi(\mathbf{x}^{(N)})^{\mathrm{T}} \end{pmatrix}_{N \times m}$$

primjeri klase j

$$\mathbf{y}_j = \begin{pmatrix} y_j^{(1)} \\ y_j^{(2)} \\ \vdots \\ y_j^{(N)} \end{pmatrix} \qquad y_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

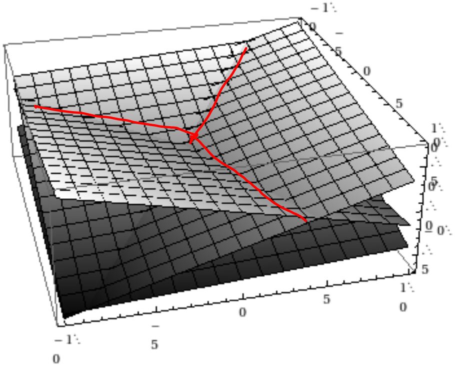
Rješenje (u smislu najmanjih kvadrata):

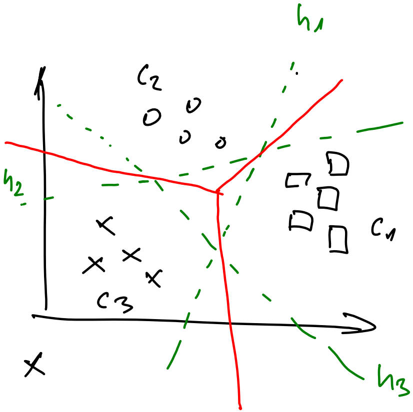
$$\tilde{\mathbf{w}}_j = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{y}_j = \boldsymbol{\Phi}^{+}\mathbf{y}_j$$

Klasifikacija: $h(x) = \arg\max_j \tilde{w}_j \phi(x)$

Klasifikacija regresijom

Primjer ($K = 3$):





Klasifikacija regresijom – prednosti i nedostatci

Prednosti:

(1) Postoji rješenje u zatvorenoj formi

(2) Jednostavan postupak

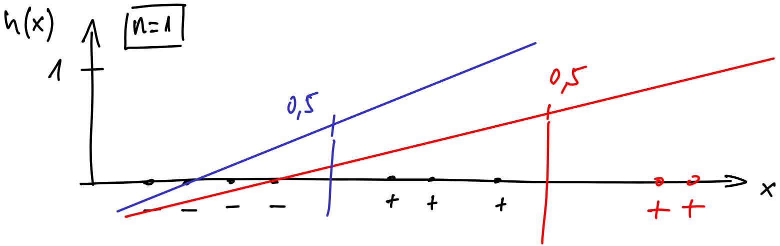
Nedostatci:

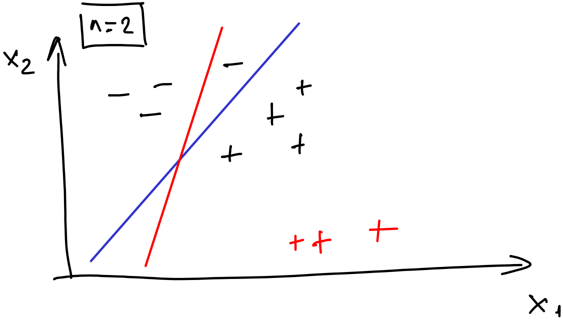
(1) Izlazi modela nemaju vjerojatnosnu intepretaciju ($h(\mathbf{x}^{(i)})$ nije ograničena na interval $[0, 1]$)

(2) Nerobusnost: osjetljivost na vrijednosti koje odskaču (kažnjavanje "pretočno" klasificiranih primjera)

(3) U nekim slučajevima: pogrešna klasifikacija unatoč tome što su primjeri linearno odvojivi
(povezano s (2))

Klasifikacija regresijom – nedostatci





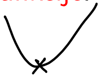
Danas. . .

1. Poopćeni linearni model
2. Klasifikacija regresijom
3. Gradijentni spust
4. Perceptron

Konveksna optimizacija

Jedna od osnovnih komponenti svakog algoritma strojnog učenja jest minimizacija funkcije pogreške.

Funkcije pogreške uglavnom su konveksne funkcije. Takve funkcije imaju globalni minimum.



Optimizacijom konveksnih funkcija bavi se konveksna optimizacija:

$$\begin{aligned} \text{minimiziraj} \quad & f_0(\mathbf{x}) && \longrightarrow \text{funkcija cilja} \\ \text{tako da} \quad & f_i(\mathbf{x}) \leqslant b_i, \quad i = 1, \dots, m && \rightarrow \text{ograničenja} \end{aligned}$$

U nekim slučajevima, iako je f_0 konveksna, minimizacija nema rješenje u zatvorenoj formi. Tada koristimo iterativne optimizacijske postupke.

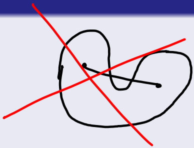
Najjednostavniji iterativan postupak (za optimizaciju bez ograničenja) jest gradijentni spust.

Podsjetnik: konveksna funkcija

Konveksna funkcija

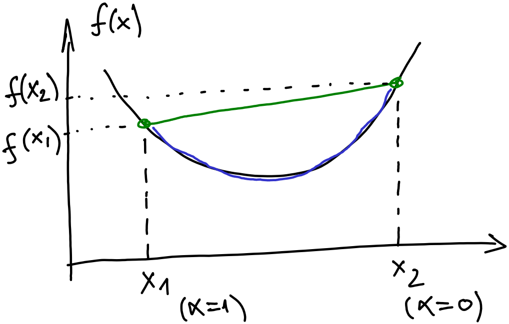
Funkcija $\mathbb{R}^n \to \mathbb{R}$ je konveksna akko

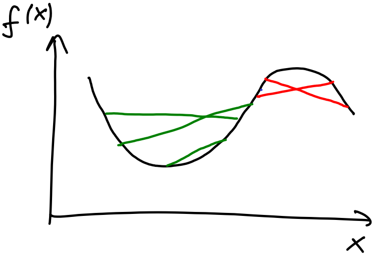
(1) Njezina domena $\mathrm{dom}(f)$ je konveksni skup



(2) Za svaki $\mathbf{x}_1, \mathbf{x}_2 \in \mathrm{dom}(f)$ i svaki $\alpha \in [0,1]$ vrijedi:

$$f(\mathbf{x}) = f\big(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2\big) \leqslant \alpha f(\mathbf{x}_1) + (1-\alpha) f(\mathbf{x}_2)$$





Podsjetnik: konveksna funkcija

Neke konveksne funkcije:

- afina funkcija: $f(x) = ax + b$, za $a, b \in \mathbb{R}$
- eksponencijalna funkcija: $f(x) = e^{ax}$, za $a \in \mathbb{R}$
- potenciranje (na $\mathbb{R}^+$): $f(x) = x^a$, za $a \leqslant 0$ ili $a \geqslant 1$
- potencije apsolutne vrijednosti: $|x|^p$, za $p \geqslant 1$
- norme: $\|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{1/p}$, za $p \geqslant 1$

Neke konkavne funkcije:

- afina funkcija: $f(x) = ax + b$, za $a, b \in \mathbb{R}$
- potenciranje (na $\mathbb{R}^+$): $f(x) = x^a$, za $0 \leqslant a \leqslant 1$
- logaritamska funkcija (na $\mathbb{R}^+$): $f(x) = \ln x$

$f(x)$ je konkavna $\iff$ $-f(x)$ je konveksna

Podsjetnik: konveksna funkcija

Operacije koje čuvaju konveksnost:

(1) Ako je f konveksna, onda je αf konveksna, $\alpha \geqslant 0$

(2) Ako su f i g konveksne, onda je $f(x) + g(x)$ konveksna

(3) Kompozicija s afinom funkcijom:
Ako je f konveksna, onda je $f(Ax + b)$ konveksna

(4) Ako su $f_1, \ldots, f_m$ konveksne, onda je $\max(f_1, \ldots, f_m)$ konveksna

(5) Kompozicija sa skalarnom funkcijom:
ako je $h : \mathbb{R} \to \mathbb{R}$ konveksna i neopadajuća (nerastuća), a g konveksna, onda je $f(x) = h(g(x))$ koveksna (konkavna)

$\vdots$

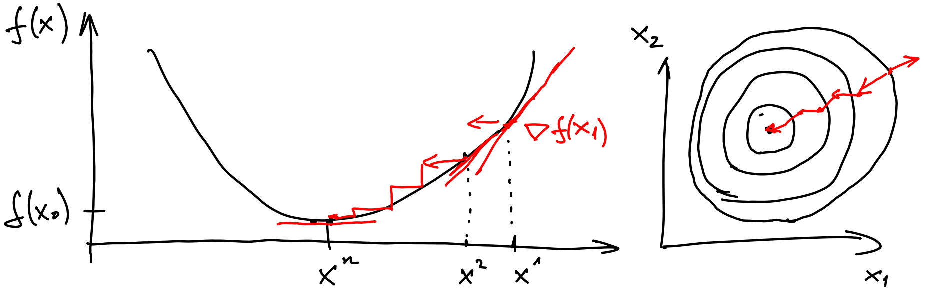
Podsjetnik: konveksna funkcija

Q: Jesu li sljedeće funkcije konveksne?

- $\mathbf{w}^{\mathrm{T}}\mathbf{x}$
- $\sum_i \mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)}$
- $-\ln\exp\left(-x^2\right)$
- $-\ln\prod_i \exp(-\|\mathbf{x}^{(i)}\|^2)$

DZ

Gradijentni spust



Gradijentni vektor $\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_0}, \frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right)$

$\nabla f(\mathbf{x})$ odgovara smjeru porasta funkcije u točki $\mathbf{x}$.
Ako je $\mathbf{x}$ minimum, onda $\nabla f(\mathbf{x}) = 0$.

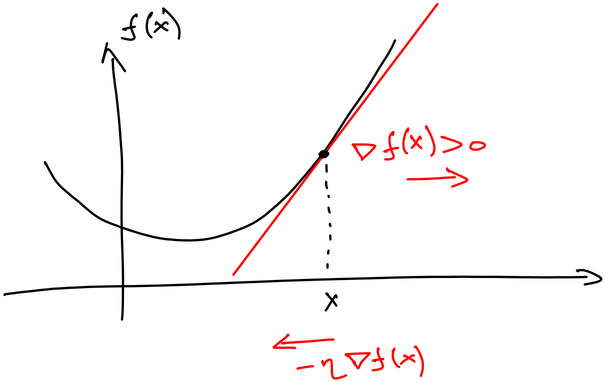
Minimumu se približavamo kretanjem u smjeru suprotnom od $\nabla f(\mathbf{x})$:

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

stopa učenja

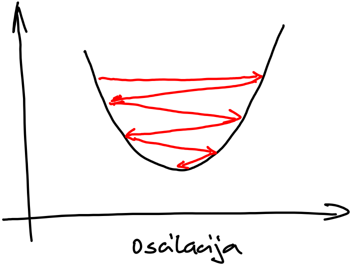
Gradijentni spust (engl. *gradient descent, steepest descent*)

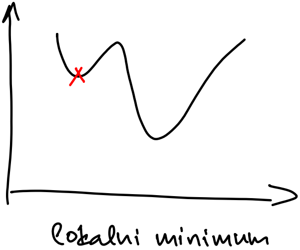
inicijaliziraj $\mathbf{x} \leftarrow (0, \ldots, 0)$
ponavljaj do konvergencije
$\quad \mathbf{x} \leftarrow \mathbf{x} - \eta \nabla f(\mathbf{x})$



Gradijentni spust

- Kriteriji konvergencije:
  (a) dosezanje unaprijed zadanog broja iteracija
  (b) stagnacija u promjeni vrijednosti funkcije: $\|\nabla f(\mathbf{x})\|_2 \leqslant \epsilon$
- Stopa učenja η ne smije biti ni prevelika niti premala, inače postupak može oscilirati ili čak divergirati.
- Gradijentni spust nalazi minimum ako je funkcija (striktno) konveksna, inače postoje lokalni mimimumi u kojima postupak može zaglaviti.





Minimizacija funkcije pogreške gradijentnim spustom

Funkcija pogreške je očekivanje funkcije gubitka:

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

Ako je funkcija gubitka L konveksna, onda je to i funkcija pogreške E. Funkcije gubitka uglavnom su konveksne.

$\nabla E = \nabla L(x^1) + \ldots \nabla L(x^N)$

Funkcija E je definirana kao suma. Imamo dvije mogućnosti:

(1) Računamo gradijent na temelju svih primjera i zatim korigiramo $\mathbf{w}$
$\Rightarrow$ standardni gradijentni spust (*batch*)

(2) Za svaki primjer $\mathbf{x}^{(i)}$ računamo gradijent i odmah korigiramo $\mathbf{w}$
$\Rightarrow$ stohastički gradijentni spust (*on-line*)

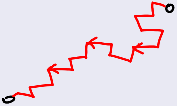
Minimizacija funkcije pogreške gradijentnim spustom

Gradijentni spust (*batch*)

inicijaliziraj $\mathbf{w} \leftarrow (0, \dots, 0)$
ponavljaj do konvergencije
  $\Delta\mathbf{w} = (0, \dots, 0)$
  za $i = 1, \dots, N$
    $\Delta\mathbf{w} \leftarrow \Delta\mathbf{w} + \nabla L\big(h(\mathbf{x}^{(i)}|\mathbf{w}), y^{(i)}\big)$
  $\mathbf{w} \leftarrow \mathbf{w} - \eta\Delta\mathbf{w}$

Stohastički gradijentni spust (*on-line*)

inicijaliziraj $\mathbf{w} \leftarrow (0, \dots, 0)$
ponavljaj do konvergencije
  (slučajno permutiraj primjere u $\mathcal{D}$)
  za $i = 1, \dots, N$
    $\mathbf{w} \leftarrow \mathbf{w} - \eta\nabla L\big(h(\mathbf{x}^{(i)}|\mathbf{w}), y^{(i)}\big)$



Gradijentni spust – napomene

- Stohastički gradijentni spust manje je računalno zahtjevan od standardnog gradijentnog spusta, pa je prikladniji za veće skupove podataka
- Parametar η treba prikladno odabrati. Najbolje je provjeriti ponašanje funkcije pogreške E kroz iteracije (trebala bi monotono padati)
- Parametar η ovisi o broju primjera: za veći N treba smanjiti η
- Može se koristiti adaptivni parametar η (krenuti s većom vrijednošću, pa je postepeno smanjivati)
- Moguće je u svakom koraku odabrati optimalnu veličinu koraka: gradijentni spust s linijskim pretraživanjem (engl. *line search*)
- Umjesto gradijentnog spusta, mogu se koristiti napredniji postupci: postupak konjugiranih gradijenata ili Newton-Raphsonov postupak. Ti su postupci računalno složeniji, no brže kovergiraju.

Danas. . .

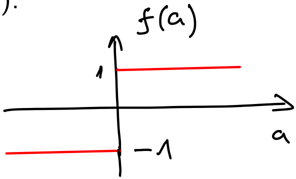
Model perceptrona

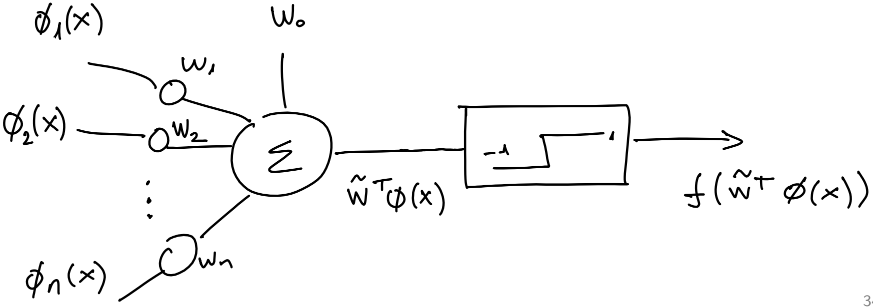
$$\boxed{h(\mathbf{x}) = f\big(\tilde{\mathbf{w}}^{\mathrm{T}}\phi(\mathbf{x})\big)}$$

$y^{(i)} = \{-1, +1\}$

Aktivacijska funkcija je funkcija praga (step-funkcija):

$$f(a) = \begin{cases} +1 & \text{ako } a \geqslant 0 \\ -1 & \text{inače} \end{cases}$$



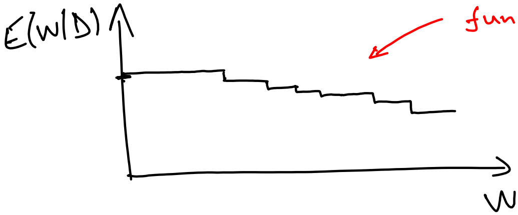


Idealna funkcija pogreške je udio pogrešno klasificiranih primjera (engl. *misclassification ratio*) (očekivanje gubitka 0-1):

$$\boxed{E(\mathbf{w}|\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{f\big(\tilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^{(i)})\big) \neq y^{(i)}\}}$$

Međutim, ova funkcija je po dijelovima konstantna, pa ne možemo primijeniti gradijentni spust.

funkcija je "većinom konstanta"



Kriterij perceptrona

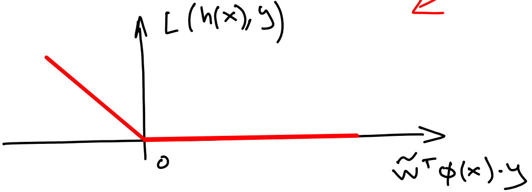
Za točno klasificirane primjere vrijedi $\tilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^{(i)})y^{(i)} > 0$, a za netočno klasificirane $\tilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^{(i)})y^{(i)} < 0$.

Kriterij perceptrona minimizira "količinu pogrešne klasifikacije":

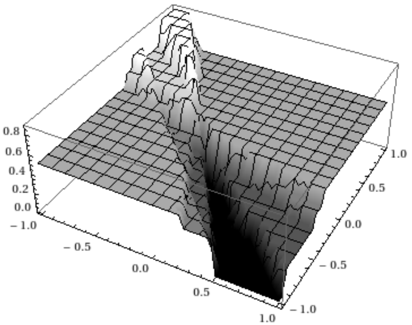
$$E(\mathbf{w}|\mathcal{D}) = -\sum_{i\,:\,f(\tilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^{(i)}))\neq y^{(i)}} \tilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^{(i)})y^{(i)} = \sum_{i=1}^{N} \underbrace{\max\left(0, -\tilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^{(i)})y^{(i)}\right)}_{\text{funkcija gubitka}}$$

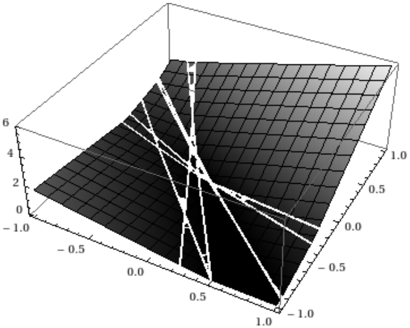
Minimizira vrijednost $-\tilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^{(i)})$ za pogrešno klasificirane primjere. Točno klasificirani primjeri ne doprinose pogrešci.

konveksna funkcija



Kriterij perceptrona





udio pogrešnih klasifikacija ⟶ kriterij perceptrona

(idealno) aproksimacija

Kriterij perceptrona je po dijelovima linearna funkcija, pa možemo primijeniti gradijentni spust. Nadalje, funkcija je konveksna, pa sigurno nalazimo globalni minimum.

Algoritam perceptrona

$\nabla E(\mathbf{w}|\mathcal{D}) = -\sum_{\mathbf{x}^{(i)}\in\mathcal{D}} \boldsymbol{\phi}(\mathbf{x}^{(i)})y^{(i)}$ za pogrešno klasificirane $\mathbf{x}^{(i)}$

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

Algoritam perceptrona (*on-line*)

inicijaliziraj $\mathbf{w} \leftarrow (0,\dots,0)$
ponavljaj do konvergencije
  za $i = 1,\dots,N$
    ako $f\big(\tilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^{(i)})\big) \neq y^{(i)}$ **onda** $\tilde{\mathbf{w}} \leftarrow \tilde{\mathbf{w}} + \eta\boldsymbol{\phi}(\mathbf{x}^{(i)})y^{(i)}$

pogrešna klasifikacija

Interpretacija: dodavanje/oduzimanje vektora primjera na vektor težina
(FN) (FP)

Teorem o konvergenciji perceptrona: ako su primjeri linearno odvojivi, algoritam perceptrona nalazi rješenje u konačnom broju koraka (Rosenblatt, 1962)

Algoritam perceptrona – primjer

Algoritam perceptrona – prednosti i nedostatci

Prednosti:

(1) Robustniji od regresije (ispravno klasificirani primjeri ne utječu na granicu)

(2) Jednostavan postupak

Nedostatci:

(1) Izlazi modela nemaju vjerojatnosnu intepretaciju ($h(\mathbf{x}^{(i)})$ nije ograničena na interval $[0, 1]$)

(2) Rezultat (hipoteza) ovisi o početnim težinama i redoslijedu korekcije

(3) Ne konvergira ako primjeri nisu linearno odvojivi

Algoritam perceptrona – prednosti i nedostatci

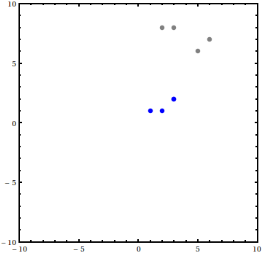
Q: Zašto rezultat ovisi o početnim težinama/redoslijedu korekcije?

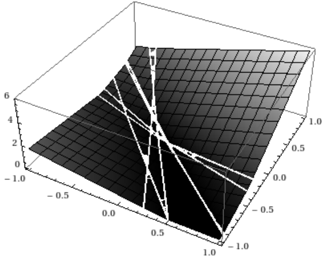
DZ

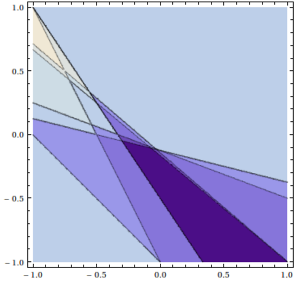
Q: Zašto postupak ne konvergira ako primjeri nisu linearno odvojivi?

DZ

Algoritam perceptrona – prednosti i nedostatci







Pogled unaprijed

Glavni problem klasifikacije regresijom jest nerobusnost: “kažnjavanje” pretočno klasificiranih primjera. Perceptron nema taj problem.

Možemo li nekako kombinirati ova dva postupka, i dobiti najbolje od oba? I k tome još probabilistički izlaz?

→ Logistička regresija!

Sažetak

- **Poopćeni linearni model** je linearni model s aktivacijskom funkcijom
- **Klasifikacija regresijom** je jednostavan postupak, ali nije robusan
- **Konveksna optimizacija** važna je za strojno učenje jer je funkcija gubitka (a time i funkcija pogreške) tipično konveksna
- Ako minimizacija nema rješenje u zatvorenoj formi, a možemo izračunati gradijent, onda možemo primijeniti **gradijentni spust**
- **Perceptron** je linearni klasifikacijski koji gradijentnim spustom mimimizira aproksimaciju broja pogrešnih klasifikacija
- Perceptron **ne konvergira** za linearno nedvojive probleme, dok za linearno odvojive rješenje ovisi o inicijalizaciji i redoslijedu primjera
- Niti regresija niti perceptron ne daju probabilistički izlaz

Sljedeća tema: Logistička regresija