

2. Osnovni koncepti

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

Jan Šnajder, natuknice s predavanja, v1.6

1 Primjena algoritma strojnog učenja

1. Priprema i analiza podataka
2. Opcionalno: Označavanje podataka za učenje i ispitivanje
3. Ekstrakcija značajki
4. Opcionalno: Redukcija dimenzionalnosti
5. **Odabir modela**
6. **Učenje modela**
7. **Vrednovanje modela**
8. Dijagnostika i ispravljanje
9. Instalacija

2 Primjeri, hipoteza, model

- Primjer je **vektor značajki**: $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$
- \mathcal{X} je **ulazni prostor (prostor primjera)**; \mathcal{Y} je skup oznaka
- Skup označenih primjera: $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N \subseteq \mathcal{X} \times \mathcal{Y}$

	x_1	x_2	\dots	x_n	y
$\mathbf{x}^{(1)} =$	$x_1^{(1)}$	$x_2^{(1)}$	\dots	$x_n^{(1)}$	$y^{(1)}$
$\mathbf{x}^{(2)} =$	$x_1^{(2)}$	$x_2^{(2)}$	\dots	$x_n^{(2)}$	$y^{(2)}$
\vdots					
$\mathbf{x}^{(N)} =$	$x_1^{(N)}$	$x_2^{(N)}$	\dots	$x_n^{(N)}$	$y^{(N)}$

- **Hipoteza** – funkcija koja primjerima dodijeljuje oznake: $h : \mathcal{X} \rightarrow \mathcal{Y}$
- **Binarna klasifikacija**: $h : \mathcal{X} \rightarrow \{0, 1\}$

- Hipoteza je definirana do na parametre θ : pišemo $h(\mathbf{x}; \theta)$
 - Regresija u $\mathcal{X} = \mathbb{R}$: $h(x; \theta_0, \theta_1) = \theta_1 x + \theta_0$
 - Klasifikacija pravcem u $\mathcal{X} = \mathbb{R}^2$: $h(x_1, x_2; \theta_0, \theta_1, \theta_2) = \mathbf{1}\{\theta_1 x_1 + \theta_2 x_2 + \theta_0 \geq 0\}$
gdje $\mathbf{1}\{P\} = \begin{cases} 1 & \text{ako } P \equiv \top \\ 0 & \text{inače} \end{cases}$
- **Model** – skup hipoteza parametriziranih s θ : $\mathcal{H} = \{h(\mathbf{x}; \theta)\}_{\theta}$
- **Učenje (treniranje) modela** – pretraživanje skupa \mathcal{H} za najboljom hipotezom

3 Empirijska pogreška i funkcija gubitka

- **Empirijska pogreška** $E(h|\mathcal{D})$ – iskazuje netočnost hipoteze h na skupu podataka \mathcal{D}
 - Pogreška klasifikacije: $E(h|\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}\{h(\mathbf{x})^{(i)} \neq y^{(i)}\}$
- **Funkcija gubitka** (*loss function*) $L(y, h(\mathbf{x}))$ – mjeri pogrešku na jednom primjeru
 - **Gubitak nula-jedan** (*zero-one loss*): $L(y, h(\mathbf{x})) = \mathbf{1}\{h(\mathbf{x})^{(i)} \neq y^{(i)}\}$
- Empirijska pogreška je **očekivana vrijednost** funkcije gubitka na skupu \mathcal{D}

4 Tri komponente algoritma strojnog učenja

1. **Model:** $\mathcal{H} = \{h(\mathbf{x}; \theta)\}_{\theta}$
2. **Funkcija pogreške:** $E(h|\mathcal{D})$ odnosno $E(\theta|\mathcal{D})$
3. **Optimizacijski postupak** koji minimizira empirijsku pogrešku:

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} E(h|\mathcal{D})$$

odnosno:

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

5 Složenost modela

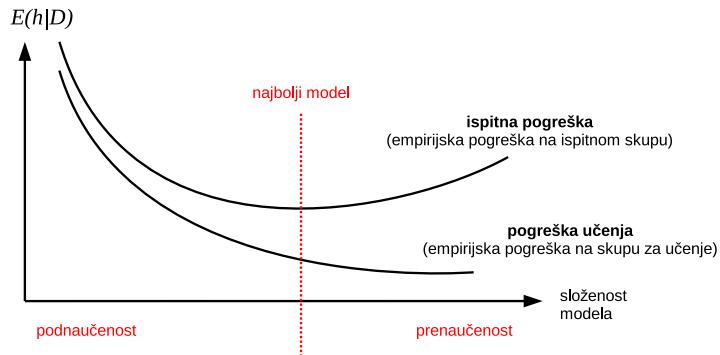
- U idealnom slučaju, $E(h|\mathcal{D}) = 0$
- Ako $\forall h \in \mathcal{H}. E(h|\mathcal{D}) > 0$, onda model nije dovoljne **složenosti (kapaciteta)**
- **Šum** – neželjena anomalija u podatcima
- Uzroci: nepreciznost, pogreške u označavanju, nedostajuće značajke, subjektivnost
- Posljedica šuma: granica između klasa je nepotrebno složena
- Presložen model previše se prilagođava šumu (uči šum)

6 Odabir modela

- Odabir modela iz **familije modela** $\{\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_k\}$
- Složenost modela određena je **hiperparametrima** (npr. stupanj nelinearnosti)
- **Odabir modela = optimizacija hiperparametara**
- Preferiramo jednostavnije modele jer bolje **generaliziraju**, lakše se uče i tumače
- **Podnaučenost** – \mathcal{H} je prejednostavan u odnosu na stvarnu funkciju
- **Prenaučenost** – \mathcal{H} je presložen u odnosu na stvarnu funkciju
- Prenaučena hipoteza nije točna na neviđenim primjerima \Rightarrow loša generalizacija

7 Unakrsna provjera

- Ideja: dio primjera iz označenog skupa koristiti kao “neviđene” primjere
- Disjunktna podjela skupa na **skup za učenje** i **skup za ispitivanje**: $\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$
- **Pogreška učenja** (*train error*): $E(h|\mathcal{D}_{\text{train}})$
- **Ispitna pogreška** (*test error*): $E(h|\mathcal{D}_{\text{test}})$
- $E(h|\mathcal{D}_{\text{train}})$ pada sa složenošću modela, $E(h|\mathcal{D}_{\text{test}})$ tipično prvo opada a zatim raste
- Skica: pogreška učenja i ispitna pogreška kao funkcije složenosti modela



- Optimalan model je onaj koji minimizira $E(h|\mathcal{D}_{\text{test}})$

3. Regresija

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

Jan Šnajder, natuknice s predavanja, v1.5

1 Jednostavna regresija

- Označen skup podataka: $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}, \mathbf{x} \in \mathbb{R}^n, y \in \mathbb{R}$
- Hipoteza $h : \mathbb{R}^n \rightarrow \mathbb{R}$
- \mathbf{x} – ulazne/nezavisne/prediktorske varijable; y – izlazna/zavisna/kriterijska varijabla
- **Linearna regresija:**

$$h(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

- **Jednostavna regresija** ($n = 1$):

$$h(x; w_0, w_1) = w_0 + w_1 x$$

- Funkcija gubitka je **kvadratni gubitak**: $L(y, h(x)) = (y - h(x))^2$
- Funkcija pogreške je zbroj kvadratnih gubitaka (**reziduala**):

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

- Optimizacijski postupak:

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} E(h|\mathcal{D}) = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^N (y^{(i)} - h(x^{(i)}))^2$$

- Za jednostavnu regresiju:

$$\nabla_{w_0, w_1} E(h|\mathcal{D}) = 0$$

$$\frac{\partial}{\partial w_0} \left[\frac{1}{2} \sum_i^N (y^{(i)} - (w_1 x^{(i)} + w_0))^2 \right] = 0$$

$$\frac{\partial}{\partial w_1} \left[\frac{1}{2} \sum_i^N (y^{(i)} - (w_1 x^{(i)} + w_0))^2 \right] = 0$$

⋮

$$w_0 = \bar{y} - w_1 \bar{x}$$

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

2 Vrste regresije

- Ulagne varijable: **jednostavna** ($n = 1$) ili **višestruka** ($n > 1$)
- Izlagne varijable: **univarijatna** ($f(\mathbf{x}) = y$) ili **multivarijatna** ($f(\mathbf{x}) = \mathbf{y}$)

	Jedan izlaz	Više izlaza
Jedan ulaz	(Univarijatna) jednostavna	Multivarijatna jednostavna
Više ulaza	(Univarijatna) višestruka	Multivarijatna višestruka

- Mi radimo samo univarijatnu regresiju

3 Tri komponente linearne regresije

- (1) Model:

$$h(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = \sum_{i=1}^n w_i x_i + w_0 = h(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \mathbf{x}$$

- (2) Funkcija gubitka i funkcija pogreške:

$$\begin{aligned} L(y^{(i)}, h(\mathbf{x}^{(i)})) &= (y^{(i)} - h(\mathbf{x}^{(i)}))^2 \\ E(h|\mathcal{D}) &= \frac{1}{2} \sum_{i=1}^N (y^{(i)} - h(\mathbf{x}^{(i)}))^2 \end{aligned}$$

- (3) Optimizacijski postupak:

$$\underset{\mathbf{w}}{\operatorname{argmin}} E(\mathbf{w}|\mathcal{D})$$

⇒ **metoda najmanjih kvadrata** (*ordinary least squares, OLS*)

- Postoji rješenje u **zatvorenoj formi**

4 Postupak najmanjih kvadrata

- Označeni primjeri daju N jednadžbi s $(n+1)$ nepoznanica:

$$\forall (\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}. \mathbf{w}^T \mathbf{x} = y^{(i)}$$

- Matrično:

$$\underbrace{\begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \vdots & & & & \\ 1 & x_1^{(N)} & x_2^{(N)} & \dots & x_n^{(N)} \end{pmatrix}}_{\mathbf{X}} \cdot \underbrace{\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix}}_{\mathbf{w}} = \underbrace{\begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{pmatrix}}_{\mathbf{y}}$$

- Matrica \mathbf{X} je **matrica dizajna**
- Egzaktno rješenje je $\mathbf{w} = \mathbf{X}^{-1}\mathbf{y}$, ali ono ne postoji ako:
 - \mathbf{X} nije kvadratna \Rightarrow pre/pododređenost sustava
 - \mathbf{X} je kvadratna, ali je sustav nekonzistentan
- Umjesto egzaktnog, tražimo približno rješenje (najmanja kvadratna odstupanja)
- Funkcija pogreške u matričnom obliku:

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

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

- Minimizacija:

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

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

- Matrica $\mathbf{X}^+ = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ je Moore-Penroseov **pseudoinverz** matrice dizajna \mathbf{X}
- Pseudoinverz minimizira normu $\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2$
- Ako je \mathbf{X} kvadratna i punog ranga, onda $\mathbf{X}^+ = \mathbf{X}^{-1}$
- $\mathbf{X}^T \mathbf{X}$ je **Gramova matrica**; $\text{rang}(\mathbf{X}^T \mathbf{X}) = \text{rang}(\mathbf{X})$
- Ako je $\text{rang}(\mathbf{X}) = n + 1$, onda $\mathbf{X}^+ = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$
- Dimenzija Gramove matrice je $(n + 1) \times (n + 1) \Rightarrow$ izračun inverza je moguće skup
- Ako $\text{rang}(\mathbf{X}) < n + 1$ (plitka matrica), onda \mathbf{X}^+ računamo pomoću SVD-a

5 Probabilistička interpretacija regresije

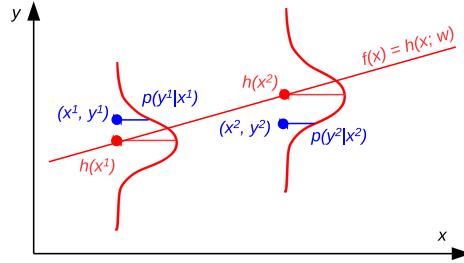
- Opažena oznaka je zbroj vrijednosti funkcije i šuma: $y^{(i)} = f(\mathbf{x}^{(i)}) + \varepsilon_i$
- Šum modeliramo kao normalno distribuiranu **slučajnu varijablu**: $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- **Normalna razdioba**:

$$p(Y = y | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - \mu)^2}{2\sigma^2}\right)$$

- Vjerojatnost oznake za zadani primjer: $p(y|\mathbf{x}) = \mathcal{N}(f(\mathbf{x}), \sigma^2)$
- Vjerojatnost da je cijeli skup primjera \mathbf{X} označen oznakama \mathbf{y} (uz pretpostavku **iid**):

$$p(\mathbf{y}|\mathbf{X}) = \prod_{i=1}^N p(y^{(i)}|\mathbf{x}^{(i)})$$

\Rightarrow **izglednost** (*likelihood*) (vjerojatnost oznaka pod modelom)



- Radi matematičke jednostavnosti, radimo s logaritmom izglednosti \Rightarrow **log-izglednost**
- Tražimo \mathbf{w} koji označe čini najvjerojatnijim \Leftrightarrow maksimizacija log-izglednosti
- Vrijedi $h(\mathbf{x}; \mathbf{w}) = f(\mathbf{x})$ (hipoteza treba aproksimirati funkciju $f(\mathbf{x})$)
- Log-izglednost težina \mathbf{w} :

$$\begin{aligned} \ln p(\mathbf{y}|\mathbf{X}, \mathbf{w}) &= \ln \prod_{i=1}^N p(y^{(i)}|\mathbf{x}^{(i)}) = \ln \prod_{i=1}^N \mathcal{N}(f(\mathbf{x}^{(i)}), \sigma^2) \\ &= \ln \prod_{i=1}^N \mathcal{N}(h(\mathbf{x}^{(i)}; \mathbf{w}), \sigma^2) \\ &= \ln \prod_{i=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - h(\mathbf{x}^{(i)}; \mathbf{w}))^2}{2\sigma^2}\right) \\ &= \underbrace{-N \ln(\sqrt{2\pi}\sigma)}_{=\text{konst.}} - \frac{1}{2\sigma^2} \sum_{i=1}^N (y^{(i)} - h(\mathbf{x}^{(i)}; \mathbf{w}))^2 \\ &\propto -\frac{1}{2} \sum_{i=1}^N (y^{(i)} - h(\mathbf{x}^{(i)}; \mathbf{w}))^2 \end{aligned}$$

\Rightarrow maksimizacija izglednosti \Leftrightarrow minimizacija pogreške kvadratnog gubitka

4. Regresija II

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

Jan Šnajder, natuknice s predavanja, v1.5

1 Nelinearna regresija

- Veza između nezavisnih varijabli i zavisne varijable često je **nelinearna**
- Neki nelinearni regresijski modeli:
 - Linearna višestruka regresija:

$$h(\mathbf{x}; \mathbf{w}) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

- Jednostruka polinomijalna regresija ($n = 1$):

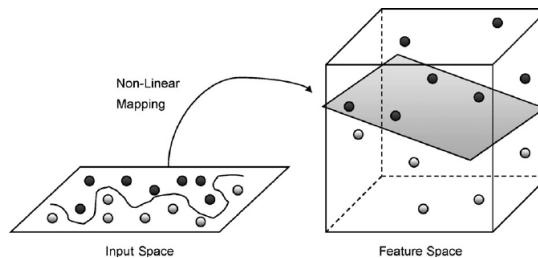
$$h(x; \mathbf{w}) = w_0 + w_1x + w_2x^2 + \cdots + w_dx^d$$

- Višestruka polinomijalna regresija ($n = 2, d = 2$):

$$h(\mathbf{x}; \mathbf{w}) = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + w_5x_2^2$$

gdje je x_1x_2 **interakcijska značajka** (*cross-term*)

- Umjesto da mijenjamo model, mijenjamo podatke \Rightarrow **preslikavanje u prostor značajki**



- **Bazne funkcije** (nelinearne funkcije ulaznih varijabli):

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

- **Funkcija preslikavanja** u prostor značajki:

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

- Model s ugrađenom funkcijom preslikavanja:

$$h(\mathbf{x}; \mathbf{w}) = \sum_{j=0}^m w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

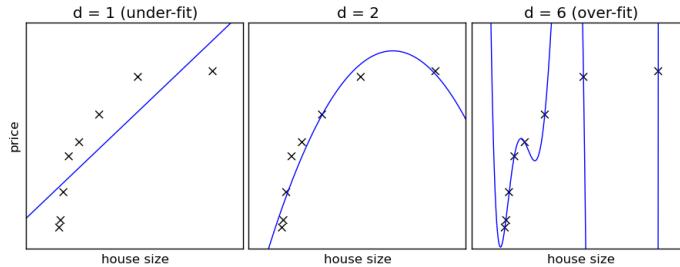
- Ova je **linearan model regresije** (linearan u parametrima) \neq linearna regresija
- Uobičajene funkcije preslikavanja:
 - Linearna višestruka regresija: $\boldsymbol{\phi}(\mathbf{x}) = (1, x_1, x_2, \dots, x_n)$
 - Jednostruka polinomijalna regresija: $\boldsymbol{\phi}(x) = (1, x, x^2, \dots, x^m)$
 - Višestruka polinomijalna regresija drugog stupnja: $\boldsymbol{\phi}(\mathbf{x}) = (1, x_1, x_2, x_1 x_2, x_1^2, x_2^2)$
- Matrica dizajna s preslikavanjem:

$$\boldsymbol{\Phi} = \begin{pmatrix} 1 & \phi_1(\mathbf{x}^{(1)}) & \dots & \phi_m(\mathbf{x}^{(1)}) \\ 1 & \phi_1(\mathbf{x}^{(2)}) & \dots & \phi_m(\mathbf{x}^{(2)}) \\ \vdots & & & \\ 1 & \phi_1(\mathbf{x}^{(N)}) & \dots & \phi_m(\mathbf{x}^{(N)}) \end{pmatrix}_{N \times (m+1)} = \begin{pmatrix} \boldsymbol{\phi}(\mathbf{x}^{(1)})^T \\ \boldsymbol{\phi}(\mathbf{x}^{(2)})^T \\ \vdots \\ \boldsymbol{\phi}(\mathbf{x}^{(N)})^T \end{pmatrix}_{N \times (m+1)}$$

- Rješenje najmanjih kvadrata: $\mathbf{w} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{y} = \boldsymbol{\Phi}^+ \mathbf{y}$

2 Prenaučenost

- Odabir preslikavanja $\boldsymbol{\phi}$ je **hiperparametar** modela
- Nelinearan model je složeniji od linearног \Rightarrow sklonost **prenaučenosti**



- Rješenje: učiti na više primjera, odabir modela, regularizacija, bayesovska regresija

3 Regularizacija

- Složeniji model \Leftrightarrow veće magnitude parametara (težina) \mathbf{w}
- Ograničavanje rasta parametara pri učenju \Rightarrow **regularizacija**
- **Rijetki modeli** (*sparse models*) – modeli s težinama pritegnutima na nulu

- Regularizirana funkcija pogreške:

$$E_R(\mathbf{w}|\mathcal{D}) = E(\mathbf{w}|\mathcal{D}) + \underbrace{\lambda \Omega(\mathbf{w})}_{\text{reg. izraz}}$$

gdje je λ **regularizacijski faktor** \Rightarrow kompromis između jednostavnosti i složenosti

- Regularizacijski izraz je p -norma vektora težina:

$$\Omega(\mathbf{w}) = \|\mathbf{w}\|_p = \left(\sum_{j=1}^m |w_j|^p \right)^{\frac{1}{p}}$$

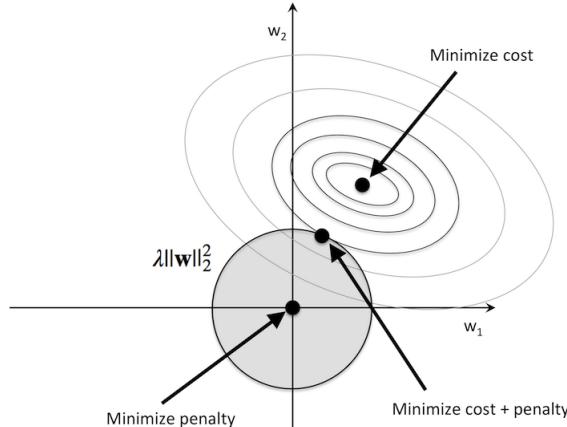
- L2-norma ($p = 2$): $\|\mathbf{w}\|_2 = \sqrt{\sum_{j=1}^m w_j^2} = \sqrt{\mathbf{w}^T \mathbf{w}}$
- L1-norma ($p = 1$): $\|\mathbf{w}\|_1 = \sum_{j=1}^m |w_j|$
- L0-norma ($p = 0$): $\|\mathbf{w}\|_0 = \sum_{j=1}^m \mathbf{1}\{w_j \neq 0\}$

- **L2-regularizacija** (Tikhonovljeva regularizacija):

$$E_R(\mathbf{w}|\mathcal{D}) = \frac{1}{2} \sum_{i=1}^N (\mathbf{w}^T \phi(\mathbf{x}^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

\Rightarrow **hrbatna regresija** (*ridge regression*)

- Skica: izokonture L2-regularizirane funkcije pogreške

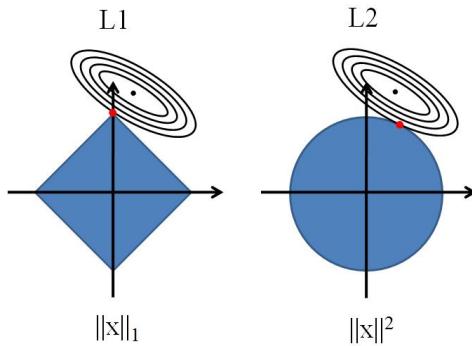


- **L1-regularizacija (LASSO):**

$$E_R(\mathbf{w}|\mathcal{D}) = \frac{1}{2} \sum_{i=1}^N (\mathbf{w}^T \phi(\mathbf{x}^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|_1$$

\Rightarrow daje rijetke modele

- Skica: usporedba izokontura L1- i L2-regulariziranih funkcija pogreške



- **L0-regularizacija:**

$$E_R(\mathbf{w}|\mathcal{D}) = \frac{1}{2} \sum_{i=1}^N (\mathbf{w}^T \phi(\mathbf{x}^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2} \sum_{j=1}^m \mathbf{1}\{w_j \neq 0\}$$

⇒ efektivno provodi **odabir značajki**

- L0-regularizacija je NP-potpuna, L1-regularizacija nema rješenje u zatvorenoj formi
- Rješenje najmanjih kvadrata s L2-regularizacijom:

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

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

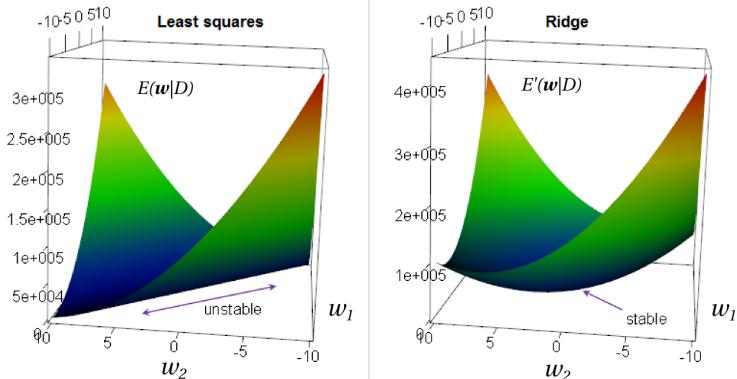
4 Regularizacija i kondicija matrice

- Rješenje najmanjih kvadrata: $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$
- $(\Phi^T \Phi)^{-1}$ definiran $\Leftrightarrow \text{rang}(\Phi^T \Phi) = \text{rang}(\Phi) = m + 1 \Leftrightarrow$ linearno nezavisni stupci
- Linearno zavisni stupci \Leftrightarrow redundantne značajke \Leftrightarrow **savršena multikolinearnost**
- **Multikolinearnost** – dvije varijable ili više njih su visoko korelirane
- Multikolinearnost daje **numerički nestabilno rješenje** \Rightarrow **prenaučenost**
- Nestabilnost rješenja iskazuje se **kondicijskim brojem** matrice
- $m \gg N \Leftrightarrow$ “široka i plitka” matrica dizajna $\Rightarrow \text{rang}(\Phi) < m + 1 \Rightarrow$ multikolinearnost
- Regularizacija smanjuje multikolinearnost:

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

⇒ dodavanje dijagonale smanjuje linearnu zavisnost \Rightarrow **rekondicioniranje matrice**

- Regularizacijom funkcija pogreške postaje konveksnija (nestaje hrbat)



5 Napomene

- Magnituda parametra w_i odgovara **važnosti značajke**, osim ako je model prenaučen
- Regularizacija **sprječava prenaučenost** prigušujući vrijednosti značajki
- Ako je model nelinearan, regularizacijom smanjujemo nelinearnost
- Težinu w_0 treba izuzeti iz regularizacijskog izraza ili treba centrirati podatke
- Odabir hiperparametra λ najčešće se provodi **unakrsnom provjerom**

5. Linearni diskriminativni modeli

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

Jan Šnajder, natuknice s predavanja, v1.9

1 Linearni diskriminativni modeli

- Linearni diskriminativni modeli – granica je linearna \Rightarrow **hiperravnina**:

$$h(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \mathbf{x}$$

- Granica između klasa je na $h(\mathbf{x}) = 0$ (ponekad: $h(\mathbf{x}) = 0.5$)
- **Diskriminativan model** – izravno modelira granicu između klasa
- Generativni vs. diskriminativni modeli

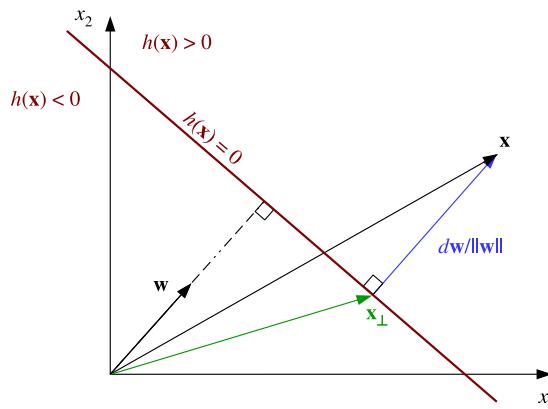
2 Geometrija linearog modela

- BSO, razmatramo sljedeći model:

$$h(\mathbf{x}; \mathbf{w}) = w_1 x_1 + w_2 x_2 + w_0$$

- Granica je pravac:

$$w_1 x_1 + w_2 x_2 + w_0 = 0$$



- \mathbf{w} je **normala** hiperravnine:

$$\begin{aligned} h(\mathbf{x}_1) &= h(\mathbf{x}_2) \\ \mathbf{w}^T \mathbf{x}_1 + w_0 &= \mathbf{w}^T \mathbf{x}_2 + w_0 \\ \mathbf{w}^T (\mathbf{x}_1 - \mathbf{x}_2) + w_0 - w_0 &= 0 \\ \mathbf{w}^T (\mathbf{x}_1 - \mathbf{x}_2) &= 0 \end{aligned}$$

- Udaljenost primjera \mathbf{x} od hiperravnine:

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

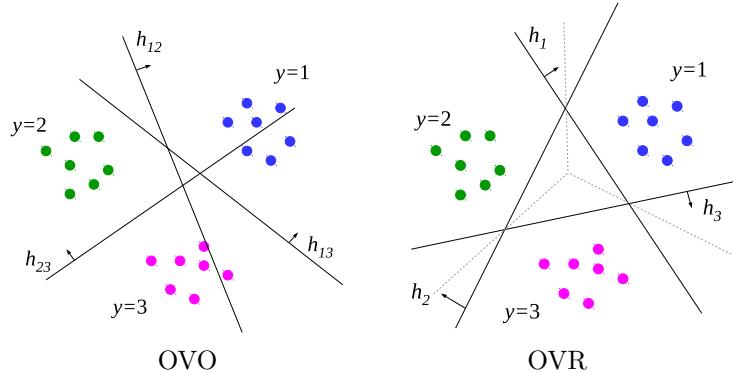
3 Višeklasna klasifikacija

- Shema **jedan-naspram-jedan** (**one-vs-one**, **OVO**) – $\binom{K}{2}$ binarnih modela:

$$h(\mathbf{x}) = \operatorname{argmax}_i \sum_{i \neq j} \operatorname{sgn}(h_{ij}(\mathbf{x})), \quad h_{ji}(\mathbf{x}) = -h_{ij}(\mathbf{x})$$

- Shema **jedan-naspram-ostali** (**one-vs-rest**, **OVR**) – K binarnih modela:

$$h(\mathbf{x}) = \operatorname{argmax}_j h_j(\mathbf{x})$$



- OVR ima manje modela od OVO, ali potencira neuravnotežnost klase

4 Klasifikacija regresijom

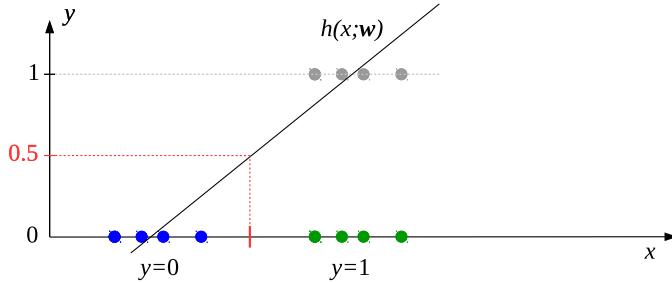
- Funkcija pogreške:

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

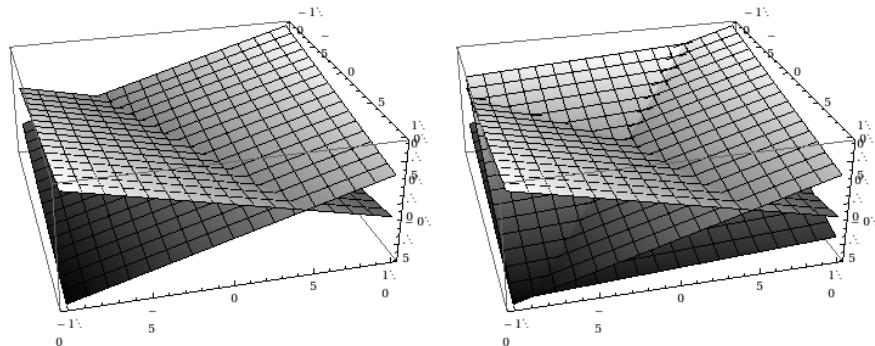
- Minimizator:

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

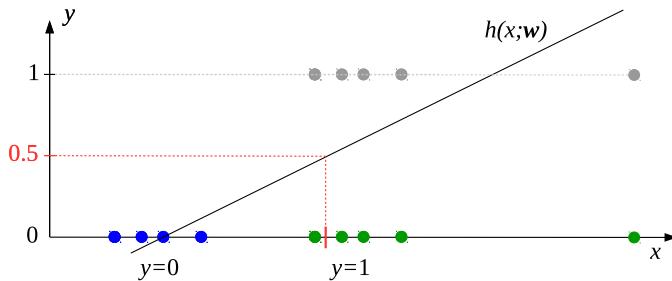
- Ideja: hipoteza koja predviđa $y = 1$ i $y = 0$ za primjere prve odnosno druge klase
- Model: $h(\mathbf{x}; \mathbf{w}) = \mathbf{1}\{\mathbf{w}^\top \phi(\mathbf{x}) \geq 0.5\}$
- Skica za $n = 1$:



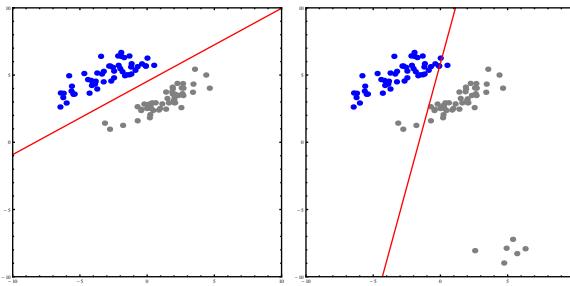
- Proširivo na $K > 2$ klase shemom OVR ili OVO



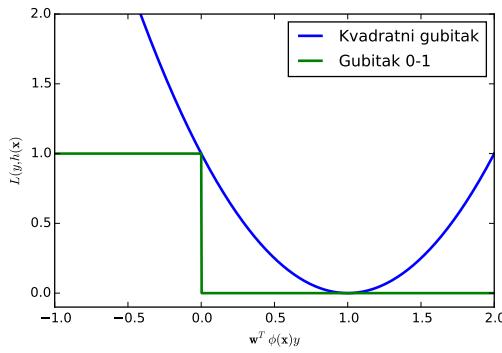
- Nedostatci: izlazi nisu vjerojatnosti, nerobusnost na vrijednosti koje odskaču
- Skica: nerobusnost na vrijednosti koje odskaču ($n = 1$):



- Nerobusnost na vrijednosti koje odskaču ($n = 2$):



- Uzrok: funkcija gubitka L kažnjava i dobro klasificirane primjere
- Za $y \in \{-1, +1\}$: ispravna klasifikacija $\Leftrightarrow \mathbf{w}^T \phi(\mathbf{x})y > 0$
- Skica: L kao funkcija od $\mathbf{w}^T \phi(\mathbf{x})y$



- Idealan gubitak je gubitak 0-1, ali nije konveksan, pa nije pogodan za optimizaciju

5 Perceptron

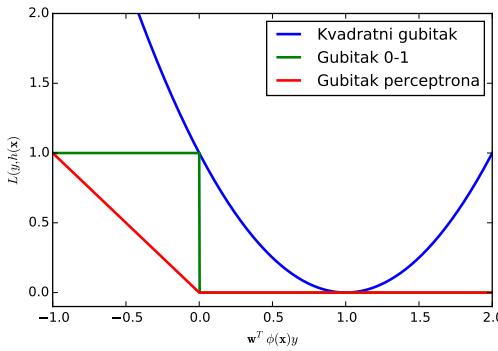
- Model:
- **Funkcija praga** kao aktivacijska funkcija:

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

- Perceptron – umjetni neuron (McCulloch & Pitts, 1943.)

- Funkcija gubitka:

$$L(y, h(\mathbf{x})) = \max(0, -\mathbf{w}^T \phi(\mathbf{x})y)$$

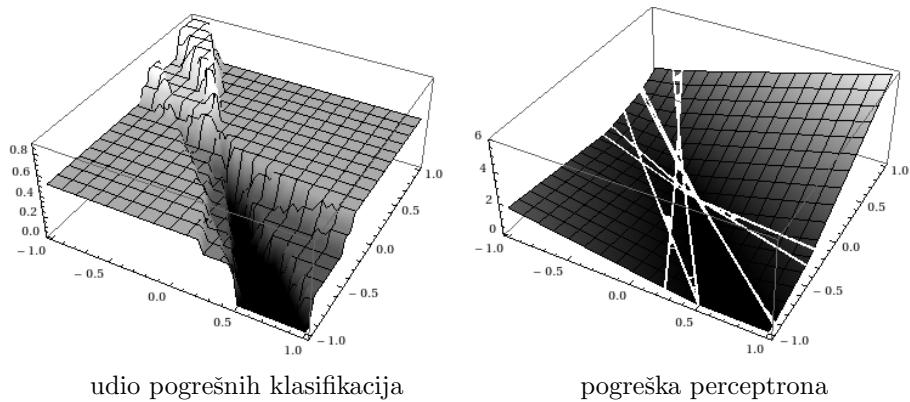


- Funkcija pogreške:

$$E(\mathbf{w}|\mathcal{D}) = - \sum_{i : f(\mathbf{w}^T \phi(\mathbf{x}^{(i)})) \neq y^{(i)}} \mathbf{w}^T \phi(\mathbf{x}^{(i)}) y^{(i)} = \sum_{i=1}^N \max(0, -\mathbf{w}^T \phi(\mathbf{x}^{(i)}) y^{(i)})$$

⇒ kažnjava samo netočno klasificirane primjere (za razliku od regresije)

- Površina pogreške u prostoru parametara:



- Ne postoji minimizator u zatvorenoj formi ⇒ primjenjujemo **gradijentni spust**
- Gradijentni spust: težine ažuriramo u smjeru suprotnome od gradijenta
- Gradijent funkcije gubitka za netočno klasificirane primjere:

$$\nabla_{\mathbf{w}} L = \nabla_{\mathbf{w}} (-\mathbf{w}^T \phi(\mathbf{x}) y) = -\phi(\mathbf{x}) y$$

- Ažuriranja težina – **Widrow-Hoffovo pravilo**:

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

tj.

$$\mathbf{w} \leftarrow \mathbf{w} + \eta \phi(\mathbf{x}) y$$

gdje je η stopa učenja

Algoritam perceptron-a

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

- Nedostatci:
 - Izlazi modela nisu vjerojatnosti
 - Konvergira samo ako su primjeri linearno odvojivi (Rosenblatt, 1962)
 - Rezultat ovisi o početnim težinama

6. Logistička regresija

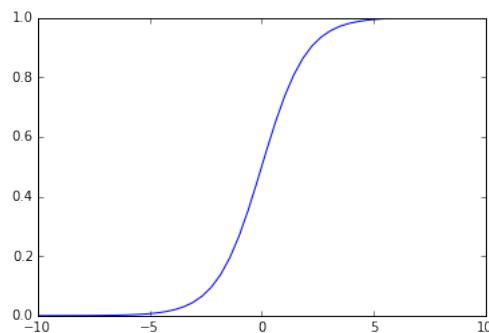
Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

Jan Šnajder, natuknice s predavanja, v1.5

1 Model logističke regresije

- **Logistička (sigmoidalna) funkcija:**

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



- Funkcija je derivabilna:

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

- Model logističke regresije:

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

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

- Ovo je primjer **poopćenog linearog modela** (*generalized linear model, GLM*)
- GLM – linearni modeli s (nelinarnom) **aktivacijskom funkcijom** f :

$$h(\mathbf{x}; \mathbf{w}) = f(\mathbf{w}^T \phi(\mathbf{x}))$$

gdje je $f : \mathbb{R} \rightarrow [0, 1]$ ili $f : \mathbb{R} \rightarrow (0, 1)$ ili ($f : \mathbb{R} \rightarrow [-1, +1]$ ili $f : \mathbb{R} \rightarrow (-1, +1)$)

2 Pogreška unakrsne entropije

- Izlaz modela je **Bernoullijeva varijabla**:

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

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

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

- Log-izglednost oznaka iz skupa označenih primjera:

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

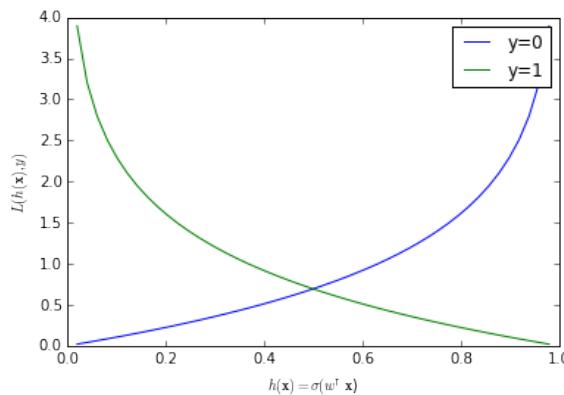
- Empirijska pogreška je negativna log-izglednost:

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

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

- Gubitak unakrsne entropije (cross-entropy loss):**

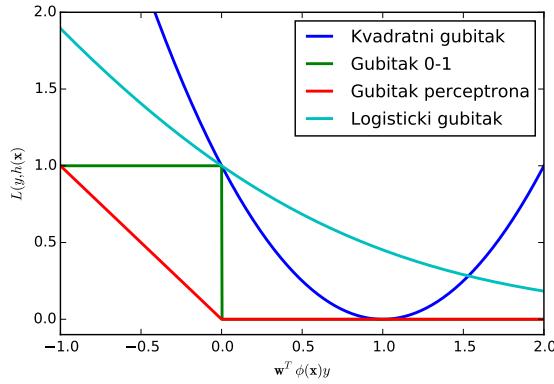
$$L(y, h(\mathbf{x})) = -y \ln h(\mathbf{x}) - (1 - y) \ln (1 - h(\mathbf{x}))$$



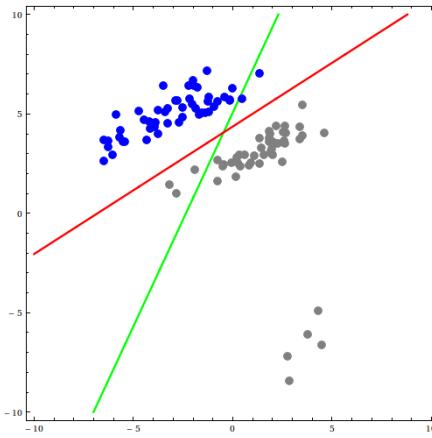
- Reformulacija $y \in \{0, 1\} \rightarrow y \in \{-1, +1\}$ i skaliranje sa $1/\ln 2$:

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

- Usporedba funkcija gubitaka:



- Logistička regresija robusnija je od modela linearne regresije:



- Minimizacija u zatvorenoj formi nije moguća \Rightarrow iterativna optimizacija

3 Gradijentni spust

- **Gradijentni spust** – minimum nalazimo krećući se u smjeru suprotnom od gradijenta:

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

- η je **stopa učenja**: prevelika $\eta \Rightarrow$ divergencija; premalena $\eta \Rightarrow$ spora konvergencija
- Želimo **globalnu konvergenciju** (konvergencija uvijek i svugdje)
- Ostvarivo **linijskim pretraživanjem** – η koji minimizira $f(\mathbf{x})$ u smjeru spusta $\Delta\mathbf{x}$:

$$g(\eta) = f(\mathbf{x} + \eta \Delta\mathbf{x})$$

- Pronađeni optimum bit će globalni optimum ako je $f(\mathbf{x})$ **konveksna**
- Funkcija $f : \mathbb{R}^n \rightarrow \mathbb{R}$ je **konveksna** akko

(1) Njezina domena $\text{dom}(f)$ je **konveksni skup**:

Za svaki $\mathbf{x}_1, \dots, \mathbf{x}_n \in \text{dom}(f)$ i za svaki $\alpha_1, \dots, \alpha_n$ takav da $\sum_i \alpha_i = 1$ vrijedi:

$$\sum_{i=1}^n \alpha_i \mathbf{x}_i \in \text{dom}(f)$$

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

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

- Empirijska pogreška je konveksna \Leftrightarrow funkcija gubitka L je konveksna
- Dvije varijante gradijentnog spusta:
 - **Batch** (grupni): $\mathbf{w} \leftarrow \mathbf{w} - \eta \sum_{i=1}^N \nabla L(y^{(i)}, h(\mathbf{x}^{(i)}; \mathbf{w}))$
 - **Stohastički (SGD)**: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla L(y^{(i)}, h(\mathbf{x}^{(i)}; \mathbf{w}))$
- SGD je pogodan za on-line učenje (big data, data streams)

4 Gradijentni spust za logističku regresiju

- Gradijent funkcije gubitka i funkcije pogreške:

$$\begin{aligned} E(\mathbf{w}|\mathcal{D}) &= \frac{1}{N} \sum_{i=1}^N \left(-y^{(i)} \ln h(\mathbf{x}^{(i)}; \mathbf{w}) - (1 - y^{(i)}) \ln (1 - h(\mathbf{x}^{(i)}; \mathbf{w})) \right) \\ \nabla_{\mathbf{w}} E(\mathbf{w}|\mathcal{D}) &= \frac{1}{N} \sum_{i=1}^N \nabla L(y^{(i)}, h(\mathbf{x}^{(i)}; \mathbf{w})) \\ \nabla L(y, h(\mathbf{x})) &= \left(-\frac{y}{h(\mathbf{x})} + \frac{1-y}{1-h(\mathbf{x})} \right) h(\mathbf{x})(1-h(\mathbf{x})) \phi(\mathbf{x}) = (h(\mathbf{x}) - y) \phi(\mathbf{x}) \\ \nabla E(\mathbf{w}|\mathcal{D}) &= \sum_{i=1}^N (h(\mathbf{x}^{(i)}) - y^{(i)}) \phi(\mathbf{x}^{(i)}) \end{aligned}$$

(faktor $1/N$ može se apsorbirati u stopu učenja η)

Logistička regresija (grupni gradijentni spust)

- ```

1: $\mathbf{w} \leftarrow (0, 0, \dots, 0)$
2: ponavljam do konvergencije
3: $\Delta \mathbf{w} \leftarrow (0, 0, \dots, 0)$
4: za $i = 1, \dots, N$
5: $h \leftarrow \sigma(\mathbf{w}^T \phi(\mathbf{x}^{(i)}))$
6: $\Delta \mathbf{w} \leftarrow \Delta \mathbf{w} - (h - y^{(i)}) \phi(\mathbf{x}^{(i)})$
7: $\eta \leftarrow$ optimum linijskim pretraživanjem u smjeru spusta $\Delta \mathbf{w}$
8: $\mathbf{w} \leftarrow \mathbf{w} + \eta \Delta \mathbf{w}$

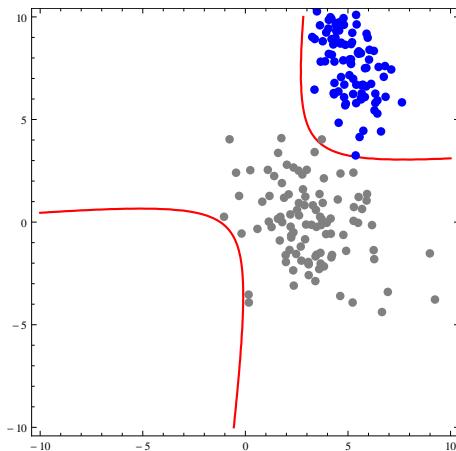
```

### Logistička regresija (stohastički gradijentni spust)

- 1:  $\mathbf{w} \leftarrow (0, 0, \dots, 0)$
- 2: **ponavlja** do konvergencije
- 3: slučajno permutiraj primjere u  $\mathcal{D}$
- 4: **za**  $i = 1, \dots, N$
- 5:  $h \leftarrow \sigma(\mathbf{w}^T \phi(\mathbf{x}^{(i)}))$
- 6:  $\Delta \mathbf{w} \leftarrow -(h - y^{(i)}) \phi(\mathbf{x}^{(i)})$
- 7:  $\eta \leftarrow$  optimum linijskim pretraživanjem u smjeru spusta  $\Delta \mathbf{w}$
- 8:  $\mathbf{w} \leftarrow \mathbf{w} + \eta \Delta \mathbf{w}$

## 5 Regularizirana regresija

- Prednosti regularizacije:
  - Sprječavanje pretjerane nelinearnosti
  - Suzbijanje nepotrebnih značajki
  - Sprječavanje otvrdnjavanja sigmoide kod linearno odvojivih problema
- Primjer prenaučenosti ( $n = 2$ ,  $\phi(\mathbf{x}) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$ ):



- L2-regularizirana pogreška:

$$E_R(\mathbf{w} | \mathcal{D}) = \sum_{i=1}^N \left( -y^{(i)} \ln h(\mathbf{x}^{(i)}) - (1 - y^{(i)}) \ln (1 - h(\mathbf{x}^{(i)})) \right) + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

- Ažuriranje težina:

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

ekvivalentno:

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

- Napomena: Težina  $w_0$  se ne regularizira

### L2-regularizirana logistička regresija (grupni gradijentni spust)

```

1: $\tilde{\mathbf{w}} \leftarrow (0, 0, \dots, 0)$ // $\tilde{\mathbf{w}}$ je prošireni vektor (w_0, \mathbf{w})
2: ponavljam do konvergencije
3: $\Delta w_0 \leftarrow 0$
4: $\Delta \mathbf{w} \leftarrow (0, 0, \dots, 0)$
5: za $i = 1, \dots, N$
6: $h \leftarrow \sigma(\tilde{\mathbf{w}}^T \phi(\tilde{\mathbf{x}}^{(i)}))$
7: $\Delta w_0 \leftarrow \Delta w_0 - (h - y^{(i)})$
8: $\Delta \mathbf{w} \leftarrow \Delta \mathbf{w} - (h - y^{(i)}) \phi(\mathbf{x}^{(i)})$
9: $\eta \leftarrow$ optimum linijskim pretraživanjem u smjeru spusta $\Delta \tilde{\mathbf{w}}$
10: $w_0 \leftarrow w_0 + \eta \Delta w_0$
11: $\mathbf{w} \leftarrow \mathbf{w}(1 - \eta \lambda) + \eta \Delta \mathbf{w}$

```

### L2-regularizirana logistička regresija (stohastički gradijentni spust)

```

1: $\tilde{\mathbf{w}} \leftarrow (0, 0, \dots, 0)$ // $\tilde{\mathbf{w}}$ je prošireni vektor (w_0, \mathbf{w})
2: ponavljam do konvergencije:
3: slučajno permutiraj primjere u \mathcal{D}
4: za $i = 1, \dots, N$
5: $h \leftarrow \sigma(\tilde{\mathbf{w}}^T \phi(\tilde{\mathbf{x}}^{(i)}))$
6: $\Delta w_0 \leftarrow -(h - y^{(i)})$
7: $\Delta \mathbf{w} \leftarrow -(h - y^{(i)}) \phi(\mathbf{x}^{(i)})$
8: $\eta \leftarrow$ optimum linijskim pretraživanjem u smjeru spusta $\Delta \tilde{\mathbf{w}}$
9: $w_0 \leftarrow w_0 + \eta \Delta w_0$
10: $\mathbf{w} \leftarrow \mathbf{w}(1 - \eta \lambda) + \eta \Delta \mathbf{w}$

```

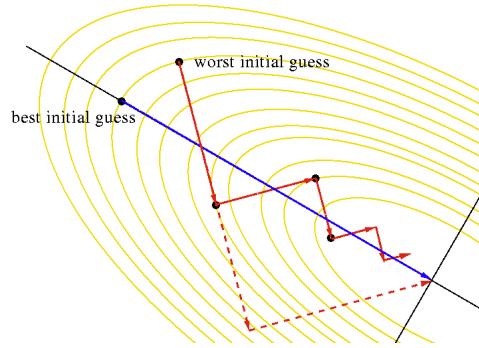
## 7. Logistička regresija II

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

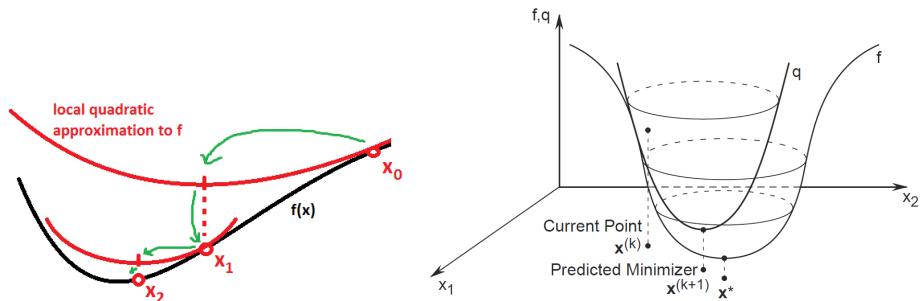
Jan Šnajder, natuknice s predavanja, v1.5

### 1 Alternative gradijentnom spustu

- Gradijentni spust s linijskim pretraživanjem ima cik-cak trajektoriju  $\Rightarrow$  sporo



- Alternativa: **optimizacija drugog reda**, npr. **Newtonov postupak**
- Ideja: skok iz trenutačnog minimuma do minimuma kvadratne aproks. funkcije



- Kvadratna aproksimacija  $f(\mathbf{x})$  u točki  $\mathbf{x}_t$  razvojem u **Taylorov red** drugog reda:

$$f(\mathbf{x}) \approx f_{\text{quad}}(\mathbf{x}) = f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^T (\mathbf{x} - \mathbf{x}_t) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_t)^T \mathbf{H}_t (\mathbf{x} - \mathbf{x}_t)$$

gdje je  $\mathbf{H}_t$  **Hesseova matrica** funkcije  $f(\mathbf{x})$  u točki  $\mathbf{x}_t$

$$\mathbf{H} = \nabla \nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

- $f(\mathbf{x})$  je konveksna  $\Leftrightarrow \mathbf{H}$  je pozitivno semi-definitna (ali ne nužno pozitivno definitna!)

- Ažuriranje parametara:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{H}_t^{-1} \nabla f(\mathbf{x}_t)$$

- Ne radi ako  $\mathbf{H}$  nije invertibilna  $\Rightarrow$  multikolinearnost  $\Rightarrow$  treba regularizirati
- Specifično, za logističku regresiju:

$$\mathbf{H} = \Phi^T \mathbf{S} \Phi$$

gdje  $\mathbf{S} = \text{diag}(h(\mathbf{x}^{(i)})(1 - h(\mathbf{x}^{(i)})))$

- Pravilo ažuriranja:

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

$\Rightarrow$  algoritam **iteratively reweighted least squares (IRLS)**

- Izračun  $\mathbf{H}_t$  u svakom koraku je potencijalno skup
- Alternativa: **kvazi-Newtonovi postupci** (BFSG, L-BFSG) – aproksimiraju  $\mathbf{H}_t$
- Uključivanje L2-regularizacije je jednostavno:

$$\nabla E_R(\mathbf{w} | \mathcal{D}) = \nabla E(\mathbf{w} | \mathcal{D}) + \lambda \mathbf{w}$$

$$\mathbf{H}_R = \mathbf{H} + \lambda I$$

- L1-regularizacija: **podgradijentne metode** (koordinatni spust, proksimalne metode)

## 2 Višeklasna logistička regresija

- OVO/OVR ne daje vjerojatnosnu distribuciju po klasama
- **Funkcija softmax:**  $\text{softmax} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , gdje za komponentu  $k$  vrijedi:

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

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

- **Multinomialna logistička regresija (MNR, maximum entropy classifier):**

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

gdje  $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_K)$

- Izlaz je **multinulijeva** (kategorička) varijabla  $\mathbf{y} = (y_1, y_2, \dots, y_K)^T$ , s distribucijom:

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

- Log-izglednost označenih primjera:

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

$\Rightarrow$  poopćena **pogreška unakrsne entropije**:

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

- Funkcija gubitka:

$$L(\mathbf{y}, h_k(\mathbf{x})) = - \sum_{k=1}^K y_k \ln h_k(\mathbf{x}; \mathbf{W})$$

- Gradijent za klasu  $k$ :

$$\nabla_{\mathbf{w}_k} E(\mathbf{W}|\mathcal{D}) = \sum_{i=1}^N (h_k(\mathbf{x}^{(i)}; \mathbf{W}) - y_k^{(i)}) \phi(\mathbf{x}^{(i)})$$

$\Rightarrow$  gradijent je isti kao i za binarnu logističku funkciju

- On-line ažuriranje:

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

$\Rightarrow$  algoritam **least-mean-squares (LMS)** ili **Widrow-Hoffovo pravilo**

- Isto dobivamo za on-line optimizaciju linearne regresije

### 3 Poopćeni linearni modeli i eksponencijalna familija

- Unificirani pogled na tri poopćena linearna modela koja smo razmatrali
- Linearna regresija:

$$h(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x})$$

$$P(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(\mu, \sigma^2) = \mathcal{N}(h(\mathbf{x}), \sigma^2)$$

$$L(y, h(\mathbf{x})) = (h(\mathbf{x}) - y)^2$$

$$\nabla_{\mathbf{w}} L(y, h(\mathbf{x})) = (h(\mathbf{x}) - y) \phi(\mathbf{x})$$

- Logistička regresija:

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

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

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

$$\nabla_{\mathbf{w}} L(y, h(\mathbf{x})) = (h(\mathbf{x}) - y) \phi(\mathbf{x})$$

- Multinomijalna logistička regresija:

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

$$P(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \prod_{k=1}^K \mu_k^{y_k} = \prod_{k=1}^K h_k(\mathbf{x})^{y_k}$$

$$L(\mathbf{y}, h_k(\mathbf{x})) = - \sum_{k=1}^K y_k \ln h_k(\mathbf{x}; \mathbf{W})$$

$$\nabla_{\mathbf{w}_k} L(y_k, h_k(\mathbf{x})) = (h_k(\mathbf{x}) - y_k) \phi(\mathbf{x})$$

- Sve tri korištene distribucije pripadaju **eksponencijalnoj familiji distribucija**:

$$p(\mathbf{x}|\boldsymbol{\theta}) = h(\mathbf{x}) \exp(\boldsymbol{\theta}^T \phi(\mathbf{x}) - A(\boldsymbol{\theta}))$$

- Ključno za poopćene linearne modele – distribucija određuje aktivacijsku funkciju:
  - Gauss  $\leftrightarrow$  funkcija identiteta, Bernoulli  $\leftrightarrow$  logistička, Multinoulli  $\leftrightarrow$  softmax

## 4 Adaptivne bazne funkcije

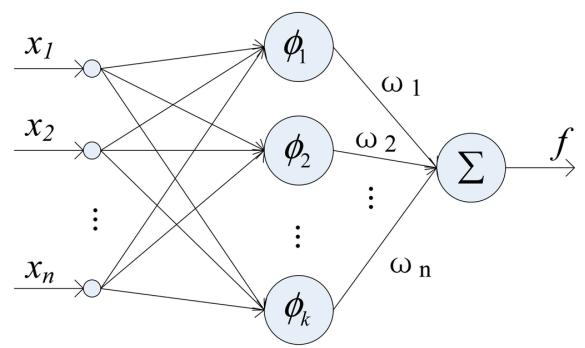
- Model s baznim funkcijama:

$$h(\mathbf{x}; \mathbf{w}) = f(\mathbf{w}^T \phi(\mathbf{x})) = f\left(\sum_{j=0}^m w_j \phi_j(\mathbf{x})\right)$$

- Fiksne (u obliku i broju) adaptivne funkcije mogu biti ograničavajuće
- **Parametrizirane bazne funkcije** – svaka bazna funkcija je poopćeni linearan model:

$$h(\mathbf{x}; \mathbf{w}) = f\left(\sum_{j=0}^m w_j^{(2)} \underbrace{f\left(\sum_{i=0}^n w_{ji}^{(1)} x_i\right)}_{=\phi_j(\mathbf{x})}\right) = f(\mathbf{w}^{(2)\text{T}} f(\mathbf{W}^{(1)} \mathbf{x}))$$

- Dobili smo dvoslojnju **neuronsku mrežu**



- Složeniji model, ali ga je lakše pretrenirati te optimizacija nije konveksna

## 8. Stroj potpornih vektora

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

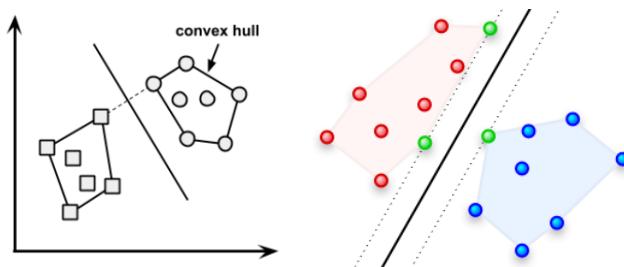
Jan Šnajder, natuknice s predavanja, v2.2

### 1 Problem maksimalne margine

- SVM je **linearan model**:

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

- Za nelinearnost možemo upotrijebiti preslikavanje  $\phi$
- **Margina** – udaljenost između hiperravnine i najbližeg primjera
- SVM nalazi **maksimalnu marginu**  $\Rightarrow$  dobra **generalizacija**
- Geometrijski: hiperravnina je simetrala spojice **konveksnih ljesaka** dviju klasa



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

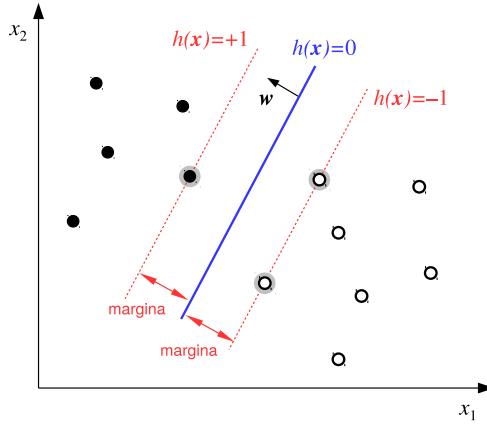
$$\forall (\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}. y^{(i)} h(\mathbf{x}^{(i)}) \geq 0$$

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

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

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

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



- Onda za sve primjere vrijedi:

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

- Optimizacijski problem svodi se na:

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

uz ograničenja:

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

- Ekvivalentno:

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

uz ograničenja:

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

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

## 2 Kvadratno programiranje

- Standardni oblik optimizacijskog problema uz ograničenja:

$$\begin{aligned} & \text{minimizirati} && f(\mathbf{x}) \\ & \text{uz ograničenja} && g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & && h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \end{aligned}$$

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$  – ciljna funkcija
- $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$  – ograničenja jednakosti
- $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$  – ograničenja nejednakosti

- Rješivo raznim metodama; mi radimo **Lagrangeovu dualnost + algoritam SMO**
- Omogućava optimizaciju u **dualnoj formi** ⇒ SMO, potporni vektori, jezgreni trik

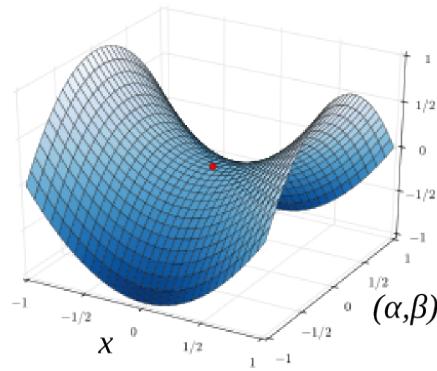
### 3 Lagrangeova dualnost

- Ograničenja kodiramo u **Lagrangeovu funkciju**:

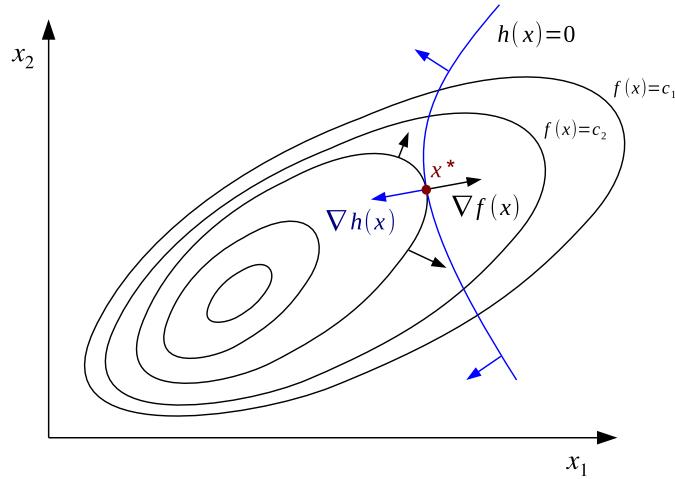
$$L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^m \alpha_i g_i(\mathbf{x}) + \sum_{i=1}^p \beta_i h_i(\mathbf{x})$$

gdje  $\alpha_i \geq 0$

- Rješenje originalnog problema je stacionarna točka u kojoj  $\nabla L = 0$
- $\nabla L = 0$  je u **točci sedla** funkcije  $L \Rightarrow$  minimum po  $\mathbf{x}$  a maksimum po  $(\boldsymbol{\alpha}, \boldsymbol{\beta})$



- Objašnjenje Lagrangeove funkcije za **ograničenje jednakosti**:



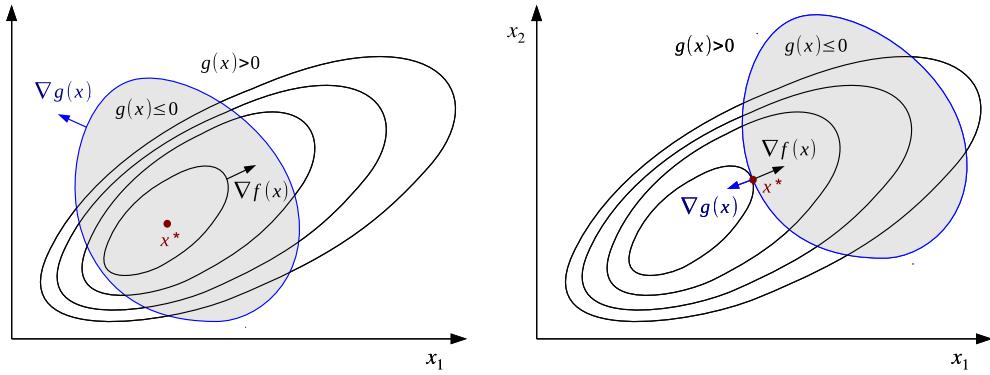
$\Rightarrow$  u stacionarnoj točci, vektori su kolinearni  $\Rightarrow$  postoji  $\beta$  za koju vrijedi:

$$\nabla f(\mathbf{x}) + \beta \nabla h(\mathbf{x}) = 0$$

što odgovara stacionarnoj točki (Lagrangeove) funkcije:

$$L(\mathbf{x}, \beta) = f(\mathbf{x}) + \beta h(\mathbf{x})$$

- Objašnjenje Lagrangeove funkcije za **ograničenje nejednakosti**:



- Moguća su dva slučaja:
  - minimum je unutar ostvarivog područja  $\Rightarrow$  ograničenje nije aktivno ( $\alpha = 0$ )
  - minimum je izvan ostvarivog područja  $\Rightarrow$  za neki  $\alpha > 0$  vrijedi:

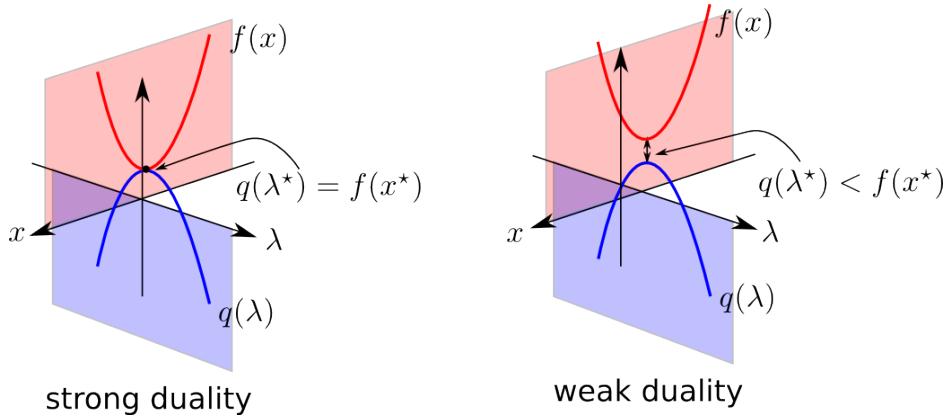
$$\nabla f(\mathbf{x}) = -\alpha \nabla g(\mathbf{x})$$

što odgovara stacionarnoj točki (Lagrangeove) funkcije:

$$L(\mathbf{x}, \alpha) = f(\mathbf{x}) + \alpha g(\mathbf{x})$$

$\Rightarrow$  u svakom slučaju, za točku rješenja vrijedi  $\alpha g(\mathbf{x}) = 0$

- Izvorna ograničenja i dva uvjeta za  $\alpha$  čine **Karush-Kuhn-Tuckerove (KKT) uvjete**
- Načelo dualnosti:** dualni problem je **donja ograda** primarnog problema



- Kod **jake dualnosti** ( $f(\mathbf{x})$  konveksna), primarno i dualno rješenje se poklapaju
- Dualna Lagrangeova funkcija:**

$$\tilde{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\alpha})$$

- Stacionarnu točku od  $L$  nalazimo **maksimizacijom** funkcije  $\tilde{L}$ , tj. dualni problem je:

$$\begin{aligned} &\text{maksimizirati} && \tilde{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &\text{uz ograničenja} && \alpha_i \geq 0, \quad i = 1, \dots, m \end{aligned}$$

## 4 Optimizacija maksimalne margine

- Lagrangeova funkcija za problem maksimalne margine:

$$L(\mathbf{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i \left\{ y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + w_0) - 1 \right\}$$

gdje  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N)$ ,  $\alpha_i \geq 0$ .

- Minimizacija funkcije  $L$  po primarnim varijablama:

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

- Uvrštavanjem u  $L$  dobivamo dualnu Lagrangeovu funkciju:

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

- Dualni optimizacijski problem SVM-a jest maksimizirati:

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

tako da:

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

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

$\Rightarrow$  kvadratni program rješiv algoritmom **SMO** (*sequential minimal optimization*)

- U točci rješenja vrijede uvjeti KKT:

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

- Od  $n + 1$  primarne varijable došli smo na  $N$  dualnih varijabli  $\Rightarrow$  nekad isplativo

## 5 Dualni model SVM-a

- Na temelju jednakosti:

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

izvodimo **dualnu formulaciju** modela:

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

⇒ umjesto težina hiperravnine  $\mathbf{w}$ , imamo dualne parametre  $\boldsymbol{\alpha}$

- Predikcija za  $\mathbf{x}^{(i)}$  temelji se na skalarnom umnošku  $\mathbf{x}^T \mathbf{x}^{(i)}$  ⇒ **sličnost vektora**
- Samo vektori za koje  $\alpha_i > 0$  utječu na predikciju ⇒ **potporni vektori**
- Težine hiperravnine (primarno) su linearna kombinacija potpornih vektora (dualno):

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

- I težina  $w_0$  se može izraziti pomoću potpornih vektora (v. jednadžbu 7.8 u skripti)

## 9. Stroj potpornih vektora II

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

Jan Šnajder, natuknice s predavanja, v2.2

### 1 Podsjetnik

- Problem maksimalne margine
- Kvadratno programiranje – primarna formulacija
- Lagrangeova dualnost – prijelaz u dualni problem
- Maksimalna margina – dualna formulacija:

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

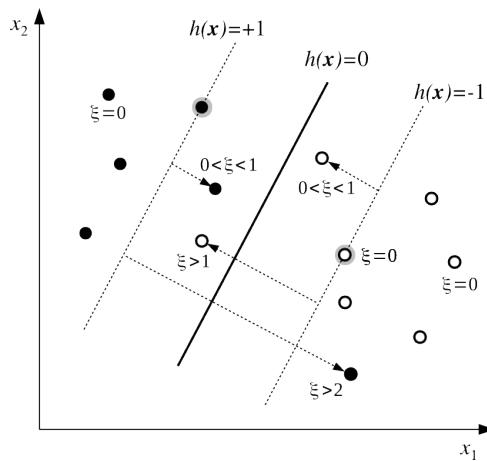
uz ograničenja:

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

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

### 2 Meka margina

- Gornja formulacija inzistira na **linearnoj odvojivosti**  $\Rightarrow$  uzrokuje **prenaučenost**
- Rješenje: dopustiti ulaske u marginu i pogrešne klasifikacije  $\Rightarrow$  **meka margina**



- Reformulacija ograničenja:

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

gdje  $\xi_i \geq 0$  govori koliko je primjer  $\mathbf{x}^{(i)}$  ušao u marginu,  $\xi_i = |y^{(i)} - h(\mathbf{x}^{(i)})|$

- $\xi_i = 0 \Rightarrow$  ispravno klasificiran i izvan margine
- $0 < \xi_i \leq 1 \Rightarrow$  ispravno klasificiran, ali unutar margine
- $\xi_i > 1 \Rightarrow$  pogrešno klasificiran

- Ciljnu funkciju proširujemo **kaznom** za primjere za koje  $\xi_i > 0$ :

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$$

- $C \uparrow \Rightarrow$  tvrda margina, složen model;  $C \downarrow \Rightarrow$  meka margina, jednostavan model

- Optimizacijski problem meke margine:

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

uz ograničenja:

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

- Pripadna **Lagrangeova funkcija**:

$$L(\mathbf{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i(y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + w_0) - 1 + \xi_i) - \sum_{i=1}^N \beta_i \xi_i$$

- Rješenje zadovoljava **uvjete KKT**:

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

- Minimum po primarnim parametrima:

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

- Uvrštavanjem u  $L$  dobivamo **dualnu Lagrangeovu funkciju**:

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

- Pripadni dualni optimizacijski problem:

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

uz ograničenja:

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

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

- Kao i za tvrdnu marginu, uz dodatno ograničenje  $\alpha_i \leq C$
- Vektori za koje  $0 < \alpha_i \leq C$  su **potporni vektori** (oni s  $\alpha_i = C$  su unutar margine)

### 3 Gubitak zglobnice

- Alternativna formulacija SVM-a: funkcija gubitka i minimizacija pogreške
- Vrijedi

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

pa kaznu po primjeru  $\xi_i$  možemo napisati kao funkciju gubitka:

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

⇒ **gubitak zglobnice (hinge loss)**

- Uvrštavanjem u ciljnu funkciju **primarnog** optimizacijskog problema:

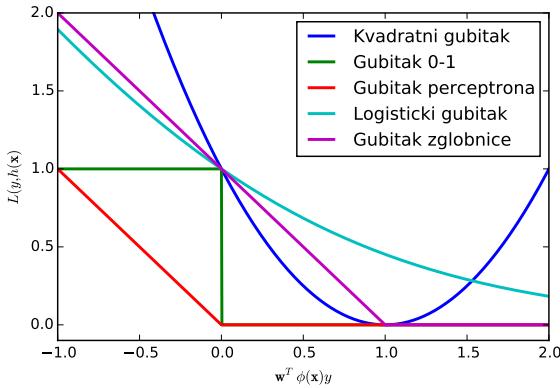
$$\operatorname{argmin}_{\mathbf{w}, w_0, \boldsymbol{\xi}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \right\}$$

dobivamo:

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

gdje  $\lambda = 1/C$

- Može se minimizirati npr. stohastičkim (pod)gradijentnim spustom
- Usporedba funkcija gubitaka:



## 4 Alternativni SVM algoritmi

- Primarna formulacija  $\Rightarrow$ 
  - Gubitak zglobnice  $\Rightarrow$  SGD, SGP, koordinatni spust, ...
  - Kvadratno programiranje (QP)  $\Rightarrow$ 
    - \* Lagrangeova dualnost  $\Rightarrow$  Dualno QP  $\Rightarrow$  SMO
    - \* Koordinatni spust, metode kazne, metode unutarnje točke, ...
- SVM rješavači: SMO, LibSVM, LibLinear, SVM<sup>light</sup>, Pegasos, ...

## 5 Napomene

- SVM regresija (SVR)
- Hiperparametar  $C$  – određuje složenost, odabrati unakrsnom provjerom
- Skaliranje – skalirati značajke, da ne dominiraju one s većim rasponom
- Višeklasna klasifikacija – preporuča se OVO zbog manje neuravnoveženosti klasa
- Probabilistički izlaz – može se aproksimirati Plattovom metodom (izlazna sigmoida)

$$P(y = 1|\mathbf{x}) = \sigma(ah(\mathbf{x}) + b)$$

- Nelinearnost – preslikavanjem  $\phi$  ili **jezgrenim trikom**  $\Rightarrow$  iduće predavanje

# 10. Jezgrene metode

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

Jan Šnajder, natuknice s predavanja, v1.3

## 1 Jezgrene funkcije

- Umjesto težina uz vektor značajki  $\mathbf{x}$ , izračunavamo **sličnost** dvaju primjera
- Naročito prikladno kada se primjeri teško vektoriziraju (npr. jer imaju strukturu)
- **Jezgrena funkcija** (*kernel function*):  $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$
- Jezgrena funkcija je **mjera sličnosti** ako zadovoljava:
  - $\kappa(\mathbf{x}, \mathbf{x}) = 1$
  - $0 \leq \kappa(\mathbf{x}, \mathbf{x}') \leq 1$
  - $\kappa(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x}', \mathbf{x})$
- Jezgre za strukturirane podatke: **string kernels**, **tree kernels**, **graph kernels**
- Tipične jezgrene funkcije za primjere u vektorskem prostoru:
  - **Linearna jezgra**:  $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$
  - **Radijalna bazna funkcija (RBF)**: općenito jezgra tipa  $\kappa(\mathbf{x}, \mathbf{x}') = f(\|\mathbf{x} - \mathbf{x}'\|)$
  - **Gaussova RBF-jezgra**:

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

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

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

- Umjesto euklidske, može se koristiti **Mahalanobisova udaljenost**:

$$\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Sigma^{-1} (\mathbf{x} - \mathbf{x}')\right)$$

gdje je  $\Sigma$  kovarijacijska matrica značajki

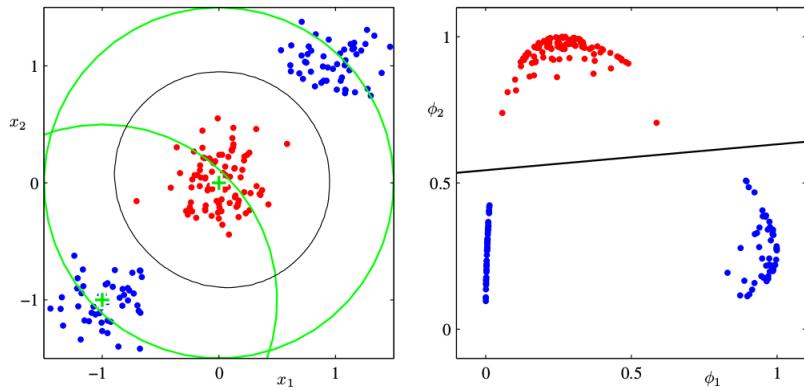
## 2 Jezgreni strojevi

- Preslikavanje  $\phi$  koje za bazne funkcije  $\phi_j$  koristi jezgrene funkcije:

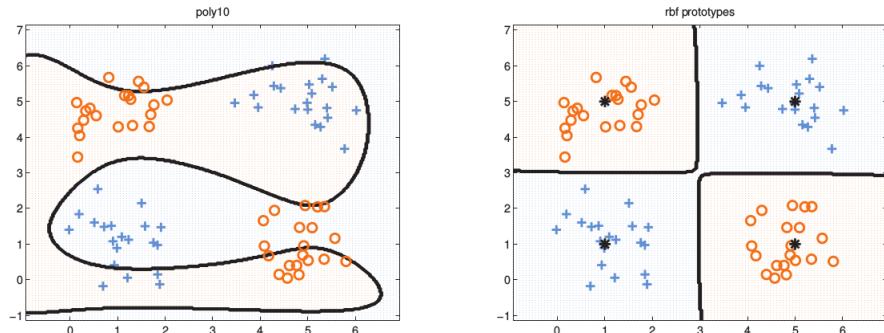
$$\phi(\mathbf{x}) = (1, \kappa(\mathbf{x}, \boldsymbol{\mu}_1), \kappa(\mathbf{x}, \boldsymbol{\mu}_2), \dots, \kappa(\mathbf{x}, \boldsymbol{\mu}_m))$$

gdje su  $\boldsymbol{\mu}_j \in \mathcal{X}$  odabrane točke u prostoru primjera

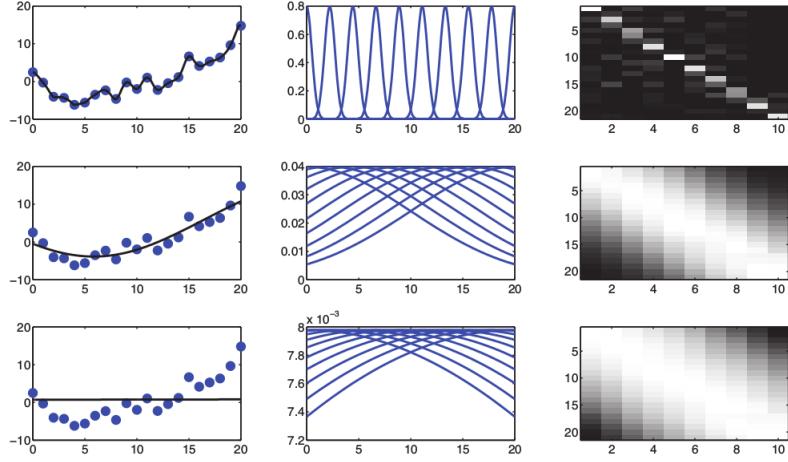
- **Jezgreni stroj** (*kernel machine*) – poopćeni linearni model s takvim preslikavanjem
- Primjer (iz MLPP) – klasifikacija:



- Primjer (iz MLPP) – klasifikacija:



- Primjer (iz MLPP) – regresija:



- Uniforman raspored  $\mu_j \Rightarrow$  neprilagođen podatcima, problem visokih dimenzija
- Alternativa:  $\mu_j$  su primjeri iz skupa za učenje:

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

- Problem: primjera može biti puno; rješenje: **L1-regularizacija**
- **Rijetki jezgreni strojevi:** L1VM, SVM

### 3 Jezgreni trik

- SVM je rijedak jezgreni stroj, no umjesto preslikavanja koristi jezgreni trik
- **Jezgreni trik** – skalarni produkt vektora zamjenjuje se jezgrenom funkcijom:

$$\kappa(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$$

- Model SVM:

$$h(\mathbf{x}) = \sum_{i=1}^N \alpha_i y^{(i)} \phi(\mathbf{x})^\top \phi(\mathbf{x}^{(i)}) + w_0 = \sum_{i=1}^N \alpha_i y^{(i)} \kappa(\mathbf{x}, \mathbf{x}^{(i)}) + w_0$$

- Ciljna funkcija (kvadratno programiranje):

$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^{(i)} y^{(j)} \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

- **Inverzno oblikovanje** – odabiremo jezgru i time implicitno definiramo  $\phi$
- Prednosti:
  - Manja računalna složenost (izračun  $\kappa$  je često jeftiniji od izračuna  $\phi$ )
  - Nekad je lakše definirati  $\kappa$  nego  $\phi$  (strukturirani podatci: nizovi, stabla, grafovi)

- Prostor koji inducira  $\kappa$  može biti visoko (potencijalno beskonačno) dimenzijski
- Uvjet:  $\kappa$  odgovara skalarnom produktu u nekom vekt. prostoru  $\Rightarrow$  **Mercerova jezgra**
- Jezgrena matrica (*kernel matrix*):

$$\mathbf{K} = \begin{pmatrix} \kappa(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) & \kappa(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) & \dots & \kappa(\mathbf{x}^{(1)}, \mathbf{x}^{(N)}) \\ \kappa(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}) & \kappa(\mathbf{x}^{(2)}, \mathbf{x}^{(2)}) & \dots & \kappa(\mathbf{x}^{(2)}, \mathbf{x}^{(N)}) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}^{(N)}, \mathbf{x}^{(1)}) & \kappa(\mathbf{x}^{(N)}, \mathbf{x}^{(2)}) & \dots & \kappa(\mathbf{x}^{(N)}, \mathbf{x}^{(N)}) \end{pmatrix}$$

- $\mathbf{K} = \Phi\Phi^T \Leftrightarrow \mathbf{K}$  je **Gramova matrica** (matrica skalarnih produkata)
- Gramova matrica je uvijek pozitivno semidefinitna ( $\forall \mathbf{x}, \mathbf{x}^T \mathbf{K} \mathbf{x} \geq 0$ )
- **Mercerov teorem:**  $\mathbf{K}$  je pozitivno semidefinitna  $\Leftrightarrow \kappa$  je Mercerova jezgra
- Inducirani prostor: skalarni produkt + proizvoljna dimenzija  $\Rightarrow$  **Hilbertov prostor**
- Mercerove jezgre: linearna, polinomijalna, RBF-jezgra, string kernels, ...
- Preslikavanje polinomijalne jezgre
  - Općenito:  $\kappa(\mathbf{x}, \mathbf{x}') = (\gamma \mathbf{x}^T \mathbf{x}' + c)^d$
  - Primjer za  $n = 2, d = 2, c = 0, \gamma = 1$ :

$$\begin{aligned} \kappa(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^T \mathbf{z})^2 = ((x_1, x_2)^T (z_1, z_2))^2 = (x_1 z_1 + x_2 z_2)^2 \\ &= (x_1 z_1)^2 + 2(x_1 z_1)(x_2 z_2) + (x_2 z_2)^2 = x_1^2 z_1^2 + \sqrt{2} x_1 x_2 \sqrt{2} z_1 z_2 + x_2^2 z_2^2 \\ &= (x_1^2, \sqrt{2} x_1 x_2, x_2^2)^T (z_1^2, \sqrt{2} z_1 z_2, z_2^2) = \phi(\mathbf{x})^T \phi(\mathbf{z}) \\ \Rightarrow \phi(\mathbf{x}) &= (x_1^2, \sqrt{2} x_1 x_2, x_2^2) \end{aligned}$$

- Preslikavanje RBF-jezgre:
  - $\mathbf{K}$  je punog ranga  $\Leftrightarrow \phi(\mathbf{x})$  su lin. nezavisni  $\Rightarrow$  **beskonačnodimenzijski prostor**
  - $\gamma \rightarrow \infty \Leftrightarrow \kappa(\mathbf{x}, \mathbf{x}') \rightarrow 0 \Leftrightarrow \phi(\mathbf{x})^T \phi(\mathbf{x}') = 0 \Leftrightarrow$  primjeri su ortonormirani  
 $\Leftrightarrow$  primjeri su vrhovi višedimenzijskog simpleksa  $\Leftrightarrow$  linearno su odvojivi
- Složenije Mercerove jezgre gradimo operacijama koje zadržavaju to svojstvo:

$$\begin{aligned} \kappa(\mathbf{x}, \mathbf{x}') &= \alpha \kappa_1(\mathbf{x}, \mathbf{x}') & \alpha > 0 \\ \kappa(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x}) \kappa_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') & f - \text{bilo koja funkcija} \\ \kappa(\mathbf{x}, \mathbf{x}') &= q(\kappa_1(\mathbf{x}, \mathbf{x}')) & q - \text{polinom s poz. koef.} \\ \kappa(\mathbf{x}, \mathbf{x}') &= \exp(\kappa_1(\mathbf{x}, \mathbf{x}')) \\ \kappa(\mathbf{x}, \mathbf{x}') &= \kappa_1(\phi(\mathbf{x}), \phi(\mathbf{x}')) & \phi : \mathbb{R}^n \rightarrow \mathbb{R}^{m+1} \\ \kappa(\mathbf{x}, \mathbf{x}') &= \kappa_1(\mathbf{x}, \mathbf{x}') + \kappa_2(\mathbf{x}, \mathbf{x}') \\ \kappa(\mathbf{x}, \mathbf{x}') &= \kappa_1(\mathbf{x}, \mathbf{x}') \kappa_2(\mathbf{x}, \mathbf{x}') \end{aligned}$$

$\Rightarrow$  **multiple kernel learning (MKL)**

## 4 Napomene

- Odabir modela kod SVM-a
  - RBF-jezgra: hiperparametri  $C$  i  $\gamma$  su međuovisni ( $\gamma \uparrow \Leftrightarrow C \downarrow$ )
    - odabir modela najčešće se radi **pretraživanjem po rešetci** (*grid search*)
- Linearna jezgra – ne daje nelinearnost, ali daje rijetka rješenja (potporni vektori)
- Jezgeni trik primjenjiv je na druge algoritme (npr. kernelizirana linearna regresija)
- **Aproksimacija kernela** (kada je  $N$  velik) – aproksimacija preslikavanja  $\phi$  + SGD

# 11. Neparametarske metode

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

Jan Šnajder, natuknice s predavanja, v1.3

## 1 Parametarske vs. neparametarske metode

- **Parametarske metode** – hipoteza je definirana do na parametre  $\theta$ 
  - broj parametara modela  $n$  (složenost modela) ne ovisi o broju primjera  $N$
  - pretpostavljaju da se podatci ravnaju po nekom modelu (distribuciji)
  - primjeri imaju **globalan** utjecaj na izgled hipoteze
- **Neparametarske metode** – hipoteza nije eksplizitno definirana
  - broj parametara ovisi o broju primjera
  - ne pretpostavljaju model (distribuciju) podataka
  - **lokalna** aproksimacija hipoteze u okolini pohranjenih primjera
- NB: Neparametarski modeli imaju parametre (ali nemaju parametre distribucije)!
- Predikcija se ne radi unaprijed nego na zahtjev  $\Rightarrow$  **lijene metode** (*lazy methods*)
- **Induktivna pristranost** neparametarskih metoda: slični primjeri imaju slične oznake
- Preporuke:
  - malo podataka i/ili poznat model/distribucija  $\Rightarrow$  parametarski postupci
  - mnogo podataka i nepoznat model/distribucija  $\Rightarrow$  neparametarski postupci

## 2 SVM

- SVM model:

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

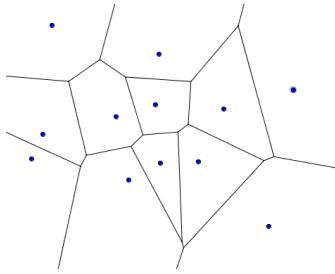
- Primarna formulacija  $\Rightarrow$  parametarski; dualna formulacija  $\Rightarrow$  neparametarski
- Broj parametara proporcionalan broju potpornih vektora, koji ovisi o  $N$
- Prikladno kada  $N \ll n$  (algoritam SMO ima složenost  $\mathcal{O}(N^2)$ )

### 3 Algoritam k-NN

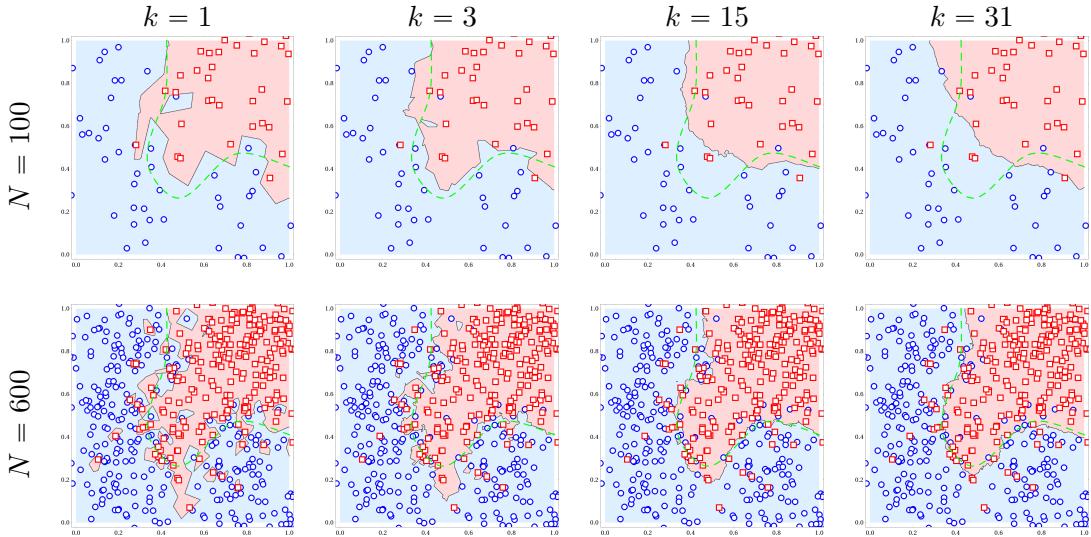
- Neparametarski klasifikacijski algoritam
- Predikcija na temelju većinske oznake  $k$  **najbližih susjeda** (*nearest neighbors*):

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

- $k$  je **hiperparametar** algoritma  $\Rightarrow$  manji  $k$  daje složeniji model
- $k = 1 \Rightarrow$  ulazni prostor particioniran u **Voronoijev dijagram**:



- Primjer: binarna klasifikacija u  $n = 2$  u ovisnosti o  $k$  za  $N = 100$  i  $N = 600$ :

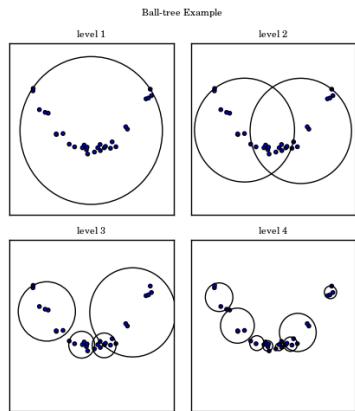


- **Težinski k-NN** – utjecaj primjera ovisi o udaljenosti/sličnosti  $\Rightarrow$  **kernel**:

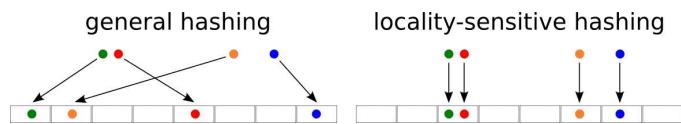
$$h(\mathbf{x}) = \operatorname{argmax}_{j \in \{0, \dots, K-1\}} \sum_{i=1}^N \kappa(\mathbf{x}^{(i)}, \mathbf{x}) \mathbf{1}\{y^{(i)} = j\}$$

- Mjera udaljenosti ne mora biti euklidska (npr. Mahalonbisova udaljenost)
- Računalni problem: **nalaženje nabližeg susjeda** (*nearest neighbor search*)
- Alternative iscrpnom pretraživanju (bitno za velike skupove podataka):

- egzaktne metode: indeksiranje prostora primjera (npr. **ball tree**)



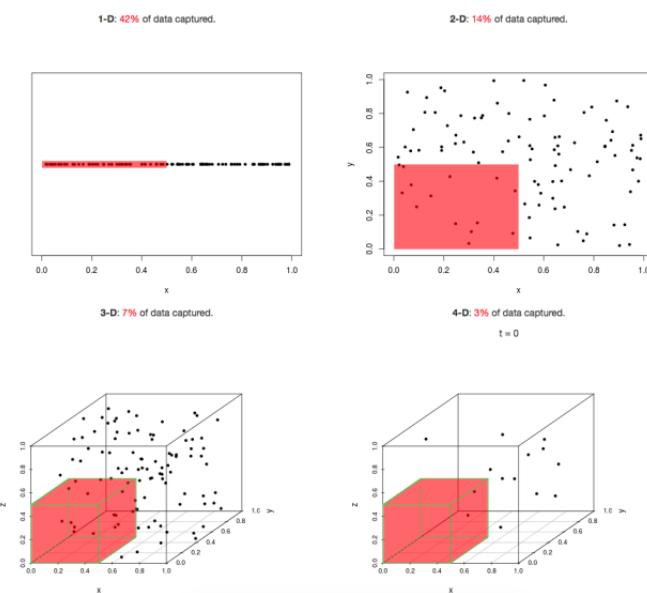
- aproksimativne metode: **locally sensitive hashing (LSH)**



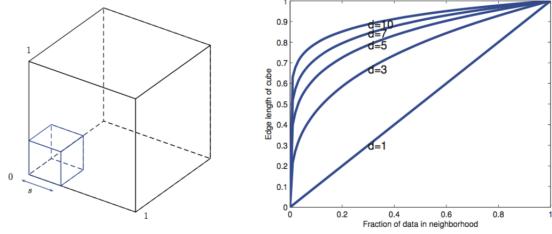
- **Prokletstvo dimenzionalnosti (curse of dimensionality):**

- s porastom dimenzije  $n$  sve točke postaju međusobno vrlo udaljene
- udaljenosti postaju nediskriminative
- općenit problem svih algoritama u visokodimenzijskim prostorima

- Primjer: s porastom broja dimenzija udio podataka u jediničnoj hiperkocki opada:



- Primjer: s porastom broja dimenzija, udaljenost između susjeda raste:



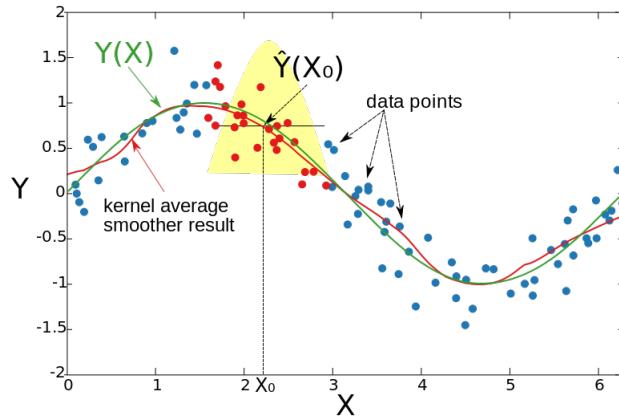
## 4 Neparametarska regresija

- Neparametarska regresija = **modeli zaglađivanja** (*smoothing models*)
- **$k$ -nn smoother** - prosjek vrijednosti  $k$  najbližih susjeda:

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

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

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



## 5 Stabla odluke

- Neparametarski model jer broj parametara ( $\propto$  broj razina) raste s brojem primjera
- Ulazni prostor rekurzivno dijeli na lokalna područja (dva potprostora)