Tomáš Oberhuber

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

Klasifikace

- jednou z nejčastějších úloh strojového učení je **klasifikace**
- jde o rozdělení vstupních objektů do několika tříd podle jejich vlastností
- objekty reprezentujeme pomocí vektorů, jejichž jednotlivé složky vyjadřují různé **vlastnosti/rysy objektů**, tzv. *feature vector*
- vhodný výběr vlastností je důležitý pro úspěšnou klasifikaci

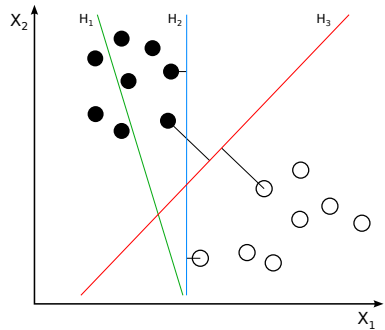
Příklady vlastností u pacientů:

- výška, věk, váha
- muž/žena
- tep, krevní tlak
- hladina cukru a cholesterolu v krvi
- prodělané nemoci, ...

Různé vlastnosti mívají často velice odlišný význam a často je potřeba je nějakým způsobem **předzpracovat**, např. normalizací nebo jinou vhodnou transformací.

Binární klasifikace

- jde o klasifikaci objektů do dvou tříd
  - nemocný/zdravý pacient
  - útočník/běžný uživatel
- použijeme stejný postup jako u regrese
- na rozdíl od regrese budeme přepokládat, že $y_i \in \{-1, 1\}$



- zde bychom mohli použít následující model

$$y_i = \text{sign}(\vec{x}_i^T \vec{w}),$$

kde

$$\text{sign}(t) = \begin{cases} 1 \text{ pro } t \geq 0, \\ -1 \text{ pro } t < 0 \end{cases}$$

- ztrátovou funkci můžeme definovat takto

$$L(\vec{w}, \mathbb{X}) = \frac{1}{2} \sum_{i=1}^{N} \left(\text{sign}(\vec{x}_i^T \vec{w}) - y_i \right)^2 + \frac{\lambda}{2} \left\| \vec{w} \right\|_2^2,$$

kde předpokládáme **implicitně zahrnutý bias**

- potíž je ale v definici funkce sign
  - není diferencovatelná v bodě $t = 0$
  - **tam, kde je diferencovatelná, je její gradient nulový**

Klasifikace metodou nejmenších čtverců

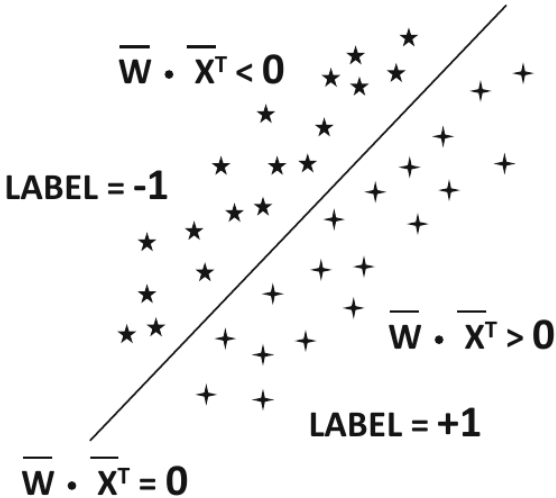
- zkusíme výraz sign ze ztrátové funkce odstranit a použít stejnou ztrátovou funkci jako u lineární regrese

$$L(\vec{w}, \mathbb{X}) = \frac{1}{2} \sum_{i=1}^{N} \left(\vec{x}_i^T \vec{w} - y_i \right)^2 + \frac{\lambda}{2} \left\| \vec{w} \right\|_2^2$$

- pokud se podaří najít parametry $\vec{w}$ tak, aby nadrovina daná vztahem

$$V \equiv \{ \vec{x} \in \mathbb{R}^n \mid \vec{x}^T \vec{w} = 0 \}$$

separovala třídu, kde je $y_i = -1$, od třídy bodů, kde je $y_i = 1$, pak by mohl náš model fungovat


$\overline{w} \cdot \overline{x}^T < 0$
LABEL = -1
$\overline{w} \cdot \overline{x}^T > 0$
LABEL = +1
$\overline{w} \cdot \overline{x}^T = 0$

Klasifikace metodou nejmenších čtverců

- pro ztrátovou funkci

$$L(\vec{w}, \mathbb{X}) = \frac{1}{2} \sum_{i=1}^{N} \left(\vec{x}_i^T \vec{w} - y_i \right)^2 + \frac{\lambda}{2} \left\| \vec{w} \right\|_2^2, \text{ kde } y_i \in \{-1, 1\},$$

- pak má SGD tvar - tzv. Widrowův-Hoffův předpis (Widrow-Hoff update)

$$\vec{w}^{(k+1)} = \vec{w}^{(k)} - \gamma \sum_{i \in S_k} \left(\vec{x}_i^T \vec{w}^{(k)} - y_i \right) \vec{x}_i - \lambda \vec{w}^{(k)},$$

- počáteční odhad se často volí jako

$$\vec{w}^{(0)} = \vec{c}_1 - \vec{c}_{-1},$$

  kde $\vec{c}_{-1}$ a $\vec{c}_1$ jsou těžiště obou tříd
- pak je

$$0 < \|\vec{w}_0\|_2^2 = \vec{w}_0^T \vec{w}_0 = (\vec{c}_1 - \vec{c}_{-1})^T \vec{w}_0 \Rightarrow \vec{c}_{-1}^T \vec{w}^{(0)} < \vec{c}_1^T \vec{w}^{(0)}$$

Klasifikace metodou nejmenších čtverců

- předpokládejme, že pro nějaké i takové, že $y_i = 1$ nastane

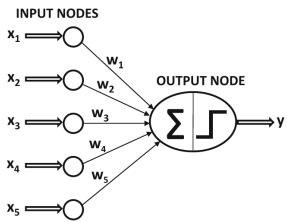
$$\vec{x}_i^T \vec{w} = 101, \text{ tj. } \left(\vec{x}_i^T \vec{w} - y_i\right) = 100.$$

- pak x_i je bod, který je správně klasifikován, ale přesto výrazně ovlivňuje konvergenci SGD
- zároveň se ukazuje, že nejvíce konvergenci ovlivňují body daleko od dělící nadroviny, i když jsou již správně klasifikované
- body, které jsou blízko dělicí nadroviny konvergenci ovlivňují méně a to i přesto, že mohou být klasifikovány špatně
- pokusíme se ztrátovou funkci navrhnout lépe

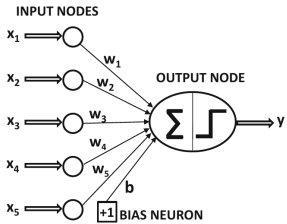
Perceptron

- preceptron (Rosenblatt,1960) je nejjednodušší neuronovou síťí
- je definovaný jako

$$y = \text{sign}(\vec{x}_i^T \vec{w} + b)$$



(a) Perceptron without bias



(b) Perceptron with bias

- vidíme, že využívá nespojitou funkci, na kterou nelze použít GD
- v době svého publikování ale tento přístup k učení nebyl známý
- ztrátová funkce je navržena tak, aby počítala špatně klasifikované vektory

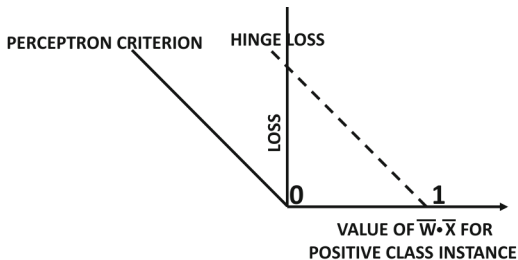
$$L_i^{(0/2)}(\vec{w}, \mathbb{X}) = \frac{1}{2}\left(y_i - \text{sign}(\vec{x}_i^T \vec{w})\right)^2 = 1 - y_i \text{sign}(\vec{x}_i^T \vec{w})$$

- je tedy $L_i^{(0/2)}(\vec{w}, \mathbb{X}) \in \{0, 2\}$
- a takto je ztrátová funkce také nespojitá

- spojitou variatnou je tzv. *perceptron criterion*

$$L_i(\vec{w}, \mathbb{X}) = \max\left\{-y_i(\vec{x}_i^T \vec{w}), 0\right\}$$

- tato funkce je spojitá a **penalizuje pouze špatně klasifikované vektory**



- pokud definujeme

$$f(z) = \max(z, 0), \text{ pak } f'(z) = \begin{cases} 1 & \text{pro } z > 0 \\ 0 & \text{pro } z \leq 0 \end{cases}$$

- pak má ztrátová funkce tvar

$$L_i(\vec{w}, \mathbb{X}) = f(-y_i(\vec{x}_i^T \vec{w}))$$

- a gradient lze zapsat takto

$$\nabla_{\vec{w}} L_i = -y_i \vec{x}_i^T f'(-y_i \vec{x}_i^T \vec{w}) = \begin{cases} -y_i \vec{x}_i^T & \text{pro } y_i(\vec{x}_i^T \vec{w}) < 0, \text{ tj. } \vec{x}_i \text{ špatně klasifikované} \\ 0 & \text{jinak} \end{cases}$$

Support Vector Machines

- SVM byly navrženy v 90. letech V. Vapnikem zejména pro účely klasifikace
- vhodnou volbou tzv. *jádrové funkce* (kernel function) je lze zobecnit i na nelineární klasifikaci - tím se teď ale zabývat nebudeme
- SVM pro lineární klasifikaci je navrženo tak, aby nepenalizovalo správně klasifikované body, pro které je

$$y_i \left(\vec{x}_i^T \vec{w} \right) > 0$$

- penalizujeme tedy ty, pro které platí

$$y_i \left(\vec{x}_i^T \vec{w} \right) \leq 0$$

- zároveň ale penalizuje i dobře klasifikované body, které jsou blízko dělící nadrovině, pro které platí

$$y_i \left(\vec{x}_i^T \vec{w} \right) < 1 \Leftrightarrow 0 < 1 - y_i \left(\vec{x}_i^T \vec{w} \right)$$

R. Berwick, Village Idiot, *An Idiot's Guide to SVM*, MIT.

SVM - Support Vector Machines

- L_2 **SVM** (Hinton,1989) ztrátová funkce je dána jako

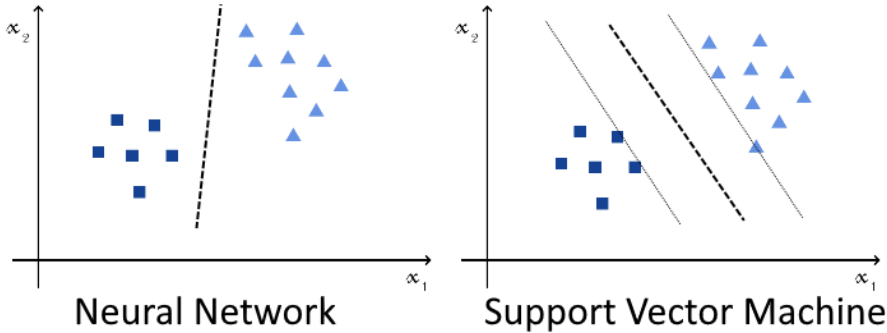
$$L(\vec{w}, b, \mathbb{X}) = \frac{1}{2} \sum_{i=1}^{N} \max \left\{ 0, \left(1 - y_i \left(\vec{x}_i^T \vec{w} \right) \right) \right\}^2 + \frac{\lambda}{2} \left\| \vec{w} \right\|_2^2$$

- **Hinge-loss SVM** (sklápěcí) ztrátová funkce je dána jako

$$L(\vec{w}, b, \mathbb{X}) = \sum_{i=1}^{N} \max \left\{ 0, \left(1 - y_i \left(\vec{x}_i^T \vec{w} \right) \right) \right\} + \frac{\lambda}{2} \left\| \vec{w} \right\|_2^2$$

- snahou SVM je, aby dělící nadrovina neprocházela blízko nekterých trénovacích bodů, což by mohlo vést k chybém při klasifikaci

SVM - Support Vector Machines



https://www.baeldung.com/cs/svm-vs-neural-network

- pomineme-li regularizaci, pak lze psát

$$L_i(\vec{w}, \mathbb{X}) = f\left(1 - y_i\left(\vec{x}_i^T \vec{w}\right)\right),$$

kde

$$f(a) = \begin{cases} \max\{0, a\} & \text{pro hinge loss,} \\ \frac{1}{2}\max\{0, a\}^2 & \text{pro } L_2 \text{ loss} \end{cases}$$

a

$$f'(a) = \begin{cases} \delta(a > 0) & \text{pro hinge loss,} \\ \max\{0, a\} & \text{pro } L_2 \text{ loss,} \end{cases}$$

kde $\delta(\cdot)$ je rovna 1, pokud je podmínka uvnitř splněna, jinak je rovno 0.

- gradient pro L_2 SVM lze potom psát jako

$$\nabla_{\vec{w}} L_i = -y_i \vec{x}_i^T f' \left(1 - y_i(\vec{x}_i^T \vec{w})\right) = \begin{cases} -y_i \vec{x}_i^T \left(1 - y_i(\vec{x}_i^T \vec{w})\right) & \text{pro } y_i(\vec{x}_i^T \vec{w}) < 1 \\ 0 & \text{jinak} \end{cases} .$$

- a pro hinge loss jako

$$\nabla_{\vec{w}} L_i = -y_i \vec{x}_i^T f' \left(1 - y_i(\vec{x}_i^T \vec{w})\right) = \begin{cases} -y_i \vec{x}_i^T & \text{pro } y_i(\vec{x}_i^T \vec{w}) < 1 \\ 0 & \text{jinak} \end{cases} .$$

Theorem 1

Pro konvexní funkce platí:

1. *Suma konvexních funkcí je konvexní funkce.*
2. *Maximum konvexních funkcí je konvexní funkce.*
3. *Druhá mocnina nezáporné konvexní funkce je konvexní funkce.*
4. *Je-li* $f : \mathbb{R} \to \mathbb{R}$ *konvexní a* $g : \mathbb{R}^n \to \mathbb{R}$ *lineární funkce, pak* $f(g(\cdot))$ *je konvexní funkce.*
5. *Je-li* $f : \mathbb{R} \to \mathbb{R}$ *konvexní, nerostoucí a* $g : \mathbb{R}^n \to \mathbb{R}$ *konkávní funkce, pak* $f(g(\cdot))$ *je konvexní funkce.*
6. *Je-li* $f : \mathbb{R} \to \mathbb{R}$ *konvexní, neklesající a* $g : \mathbb{R}^n \to \mathbb{R}$ *konvexní funkce, pak* $f(g(\cdot))$ *je konvexní funkce.*

Theorem 2

Obě L_2 a hinge SVM ztrátové funkce jsou konvexní vůči parametrům $\vec{w}$ a b. Jsou-li použity i regularizační členy, pak jsou obě funkce striktně konvexní.

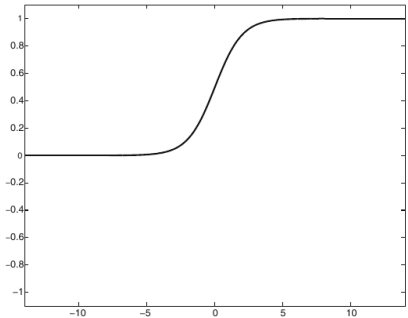
Toto tvrzení nám zaručuje **existenci globálního minima**.

Logistická regrese

- jde o pravděpodobnostní model
- budeme se snažit odhadnout pravděpodobnost $\hat{y}_i$, že $y_i = 1$
- k tomu použijeme hladkou funkci zvanou *sigmoid*

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

- chceme
  - $v \gg 0$ pro $y_i = 1$
  - $v \ll 0$ pro $y_i = -1$



(c) Sigmoid

- a budeme opět hledat ve tvaru $a = \vec{x}_i^T \vec{w}$
- dostáváme tak odhad pravděpodobnosti

$$p_i(\vec{w}) = P(y_i = 1|\vec{w}) = \frac{1}{1 + \exp(-\vec{x}_i^T \vec{w})}$$

- provedeme transformaci $t_i = 0.5(y_i + 1) \rightarrow t_i \in 0, 1$
- chceme tedy maximalizovat
  - $\max(p_i(\vec{w}))$ pokud $t_i = 1$
  - $\max(1 - p_i(\vec{w}))$ pokud $t_i = 0$
- toho lze dosáhnout pomocí následující věrohodnostní funkce

$$p(t|\vec{w}) = \prod_{i=1}^{N} p_i(\vec{w})^{t_i} \left[1 - p_i(\vec{w})\right]^{(1-t_i)}$$

tj.

$$(\vec{w}^*) = \arg\max_{\vec{w}} \prod_{i=1}^{N} p_i(\vec{w})^{t_i} \left[1 - p_i(\vec{w})\right]^{(1-t_i)}$$

- logaritmováním dostáváme

$$(\vec{w}^*) = \arg\min_{\vec{w}} \left[\underbrace{-\sum_{i=1}^{N} t_i \log p_i(\vec{w})}_{H(t,p)} \underbrace{-\sum_{i=1}^{N} (1-t_i) \log\left(1-p_i(\vec{w})\right)}_{H(1-t,1-p)} \right]$$

což je tzv. *cross entropy* ztrátová funkce.

Remark 3

Pro diskrétní náhodné distribuce p, q definované na množině $\mathcal{X}$ je ***křížová entropie*** *(cross entropy) definována jako*

$$H(p,q) = -\sum_{\vec{x}\in\mathcal{X}} p(\vec{x}) \log q(\vec{x}) = E_p\left[\log p\right]$$

- odvodíme gradient
- platí

$$\sigma(a) = \frac{1}{1+e^{-a}} = (1+e^{-a})^{-1} \Rightarrow \sigma'(a) = \frac{e^{-a}}{(1+e^{-a})^2}$$

- zároveň si všimneme, že platí

$$\sigma(1-\sigma) = \frac{1}{1+e^{-a}}\left(1 - \frac{1}{1+e^{-a}}\right) = \frac{1}{1+e^{-a}}\frac{1+e^{-a}-1}{1+e^{-a}} = \sigma'(a)$$

- derivujeme ztrátovou funkci

$$L(\vec{t}, \vec{w}) = -\sum_{i=1}^{N} \Big[t_i \log p_i(\vec{w}) + (1 - t_i) \log \left(1 - p_i(\vec{w})\right) \Big], \text{ kde } p_i(\vec{w}) = \sigma(\vec{x}_i^T \vec{w})$$

podle $\vec{w}$

$$\begin{aligned}
\frac{\partial L}{\partial w_j} &= -\sum_{i=1}^{N} \left(\frac{t_i}{p_i} - \frac{1 - t_i}{1 - p_i} \right) p_i' \frac{\partial (\vec{x}_i^T \vec{w})}{\partial w_j} \\
&= \sum_{i=1}^{N} \left(\frac{p_i - t_i}{p_i(1 - p_i)} \right) \underbrace{p_i(1 - p_i)}_{p_i'} x_{i,j} \\
&= \sum_{i=1}^{N} (p_i - t_i) x_{i,j}
\end{aligned}$$

Celkově tedy máme:

$$\nabla_{\vec{w}} L = \sum_{i=1}^{N} (p_i - t_i)\vec{x}_i,$$

A tedy:

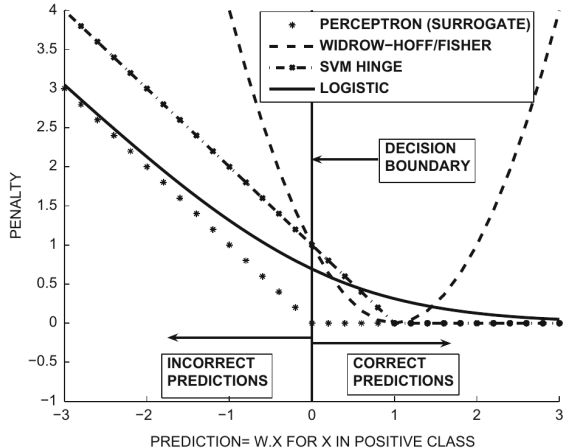
$$\vec{w}^{(k+1)} = \vec{w}^{(k)} - \gamma_1 \sum_{i=1}^{N} (p_i^{(k)} - t_i)\vec{x}_i,$$

- vidíme, že korekce parametrů z GD zavisí na

  $$p_i - t_i,$$

  což je rozdíl správné klasifikace a pravděpodobnosti odhadnuté modelem
- to je právě pro logistickou regresi typické

Porovnání ztrátových funkcí



Pozn.: Graf logistické regrese neodpovídá naší definici!!!

Example 4

- vygenerujte data pomocí skriptu

```
binary-classification-data.py
```

- pomocí parametru `overlap_parameter` lze měnit překryv dat
- proveďte výpočet lineární regrese s pomocí skriptu

```
binary-classification.py
```

Klasifikace do více tříd

- nyní budeme provádět klasifikaci do $K > 2$ tříd $\mathcal{C}_1, \ldots, \mathcal{C}_K$
- nabízí se dvě možnosti - z nichž ani jedna nefunguje
  - použít $K - 1$ klasifikátorů, každý říká, zda $x_i \in \mathcal{C}_k$ nebo ne - *one-versus-the-rest*
  - použít $K(K-1)/2$ klasifikátorů, každý klasifikuje jednu dvojici tříd - *one-versus-one*

Klasifikace do více tříd

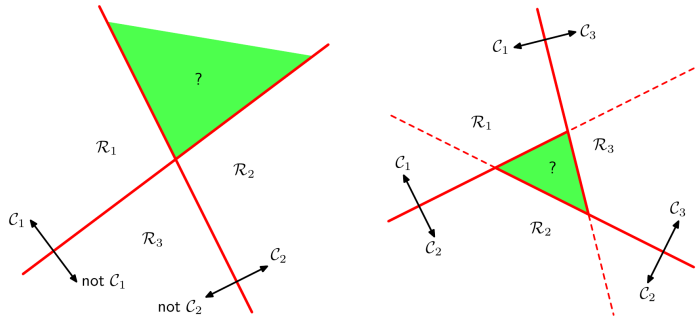


Figure: Nejednoznačnost při použití *one-versus-the-rest* klasifikátory (vlevo) a *one-versus-one* klasifikátory (vpravo).

Klasifikace do více tříd

- řešením je použít K klasifikátorů

$$c_k(\vec{x}) = \vec{x}^T \vec{w}_k + b_k,$$

  kde platí, že $\vec{x} \in \mathcal{C}_k$, je-li $c_k(\vec{x}) > c_j(\vec{x})$ pro všechna $j \neq k$
- hranice mezi třídami pak **není** dána nadrovinami $c_k(\vec{x}) = 0$
- v následujícím textu opět **nebudeme** bias vypisovat explicitně, tj. budeme psát pouze

$$c_k(\vec{x}) = \vec{x}^T \vec{w}_k,$$

Budeme používat následující značení:

- $\mathcal{C}_1 \dots \mathcal{C}_K$ jsou jednotlivé třídy
- predikce k-tého klasifikátoru je

$$c_k(\vec{x}) = \vec{x}^T \vec{w}_k = c_k(\mathbb{W}, \vec{x}_i), k = 1, \dots K$$

- index třídy, do které patří prvek $\vec{x}_i$ budeme značit jako $c(i)$

Klasifikace do více tříd - metoda nejmenších čtverců

- hledáme tedy vektory $(\vec{w}_1, \ldots, \vec{w}_K) \equiv \mathbb{W}$ tak, aby

$$c_k(\vec{x}) - c_j(\vec{x}) > 0 \text{ pro všechna } j \neq k,$$

  je-li $\vec{x} \in \mathcal{C}_k$
- například můžeme použít tzv. *1-of-K encoding*

$$\vec{t}_j = (t_{j,1}, \ldots, t_{j,K)}) = \left(0, \ldots, 0, \underbrace{1}_{j-\text{tá pozice}}, 0 \ldots, 0\right) = \vec{y}_i$$

- a následně budeme hledat $\mathbb{W}$, aby

$$\left(c_1(\mathbb{W}, \vec{x}_i), \ldots, c_K(\mathbb{W}, \vec{x}_i)\right) \approx \vec{t}_k \Leftrightarrow \vec{x} \in \mathcal{C}_k.$$

Klasifikace do více tříd - metoda nejmenších čtverců

To lze také přeformulovat jako optimalizační úlohu

$$L(\mathbb{W}, \mathbb{X}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \vec{x}_i^T \mathbb{W} - \vec{y}_i \right\|^2 + \frac{\lambda}{2} \|\mathbb{W}\|_F^2 = \frac{1}{N} \|\mathbb{X}\mathbb{W} - \mathbb{Y}\|^2 + \frac{\lambda}{2} \|\mathbb{W}\|_F^2 ,$$

kde

$$\mathbb{Y} = (\vec{y}_i, \ldots, \vec{y}_N) ,$$

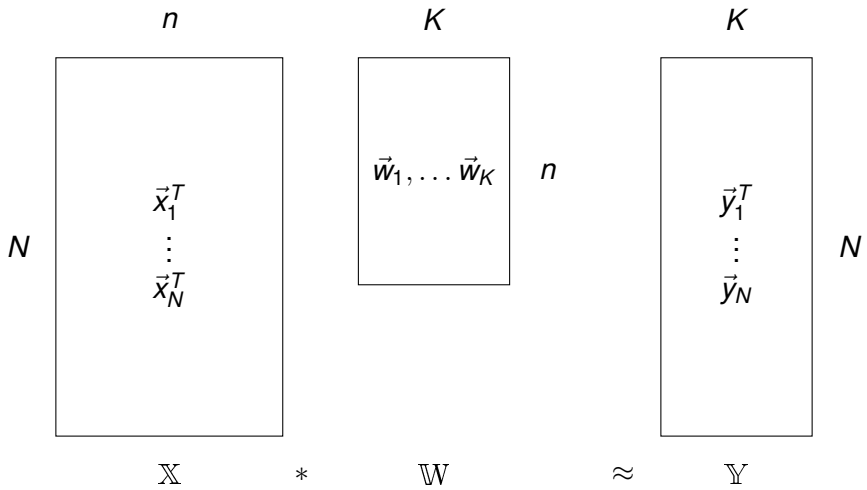
a hledáme

$$\min_{\mathbb{W}} L(\mathbb{W}, \mathbb{X}) .$$

Gradient má tvar (odvozovali jsme již v lineární regresi)

$$\nabla_{\mathbb{W}} L = \frac{1}{N} \mathbb{X}^T (\mathbb{X}\mathbb{W} - \mathbb{Y}) ,$$

Klasifikace do více tříd - metoda nejmenších čtverců



Klasifikace do více tříd - perceptron

- při konstrukci perceptronu opět vycházíme z podmínky

$$c_k(\vec{x}) - c_j(\vec{x}) > 0 \text{ pro všechna } j \neq k,$$

- pokud $\vec{x} \in \mathcal{C}_k$, tj.

$$\vec{x}^T \left(\vec{w}_k - \vec{w}_j\right) > 0 \text{ pro všechna } j \neq k,$$

- ztrátovou funkci tak můžeme definovat takto

$$\begin{aligned} L_i\left(\mathbb{W}, \mathbb{X}\right) &= \sum_{j \neq c(i)} -\min\left\{\vec{x}_i^T \left(\vec{w}_{c(i)} - \vec{w}_j\right), 0\right\} \\ &= \sum_{j \neq c(i)} \max\left\{\vec{x}_i^T \left(\vec{w}_j - \vec{w}_{c(i)}\right), 0\right\}, \end{aligned}$$

  kde $c(i)$ označuje třídu, do které patří $\vec{x}_i$
- takto penalizujeme všechny špatně klasifikované vektory $\vec{x}$

Klasifikace do více tříd - perceptron

- pak lze psát

$$L_i\left(\mathbb{W}, \vec{b}, \mathbb{X}\right) = \sum_{j \neq c(i)} \max\left\{v_{ji}, 0\right\} \text{pro } v_{ji} = \vec{x}_i^T\left(\vec{w}_j - \vec{w}_{c(i)}\right)$$

- a pro gradient dostáváme

$$\frac{\partial L_i}{\partial \vec{w}_r} = \underbrace{\frac{\partial L_i}{\partial v_{ji}}}_{\delta(v_{ji}>0)} \underbrace{\frac{\partial v_{ji}}{\partial \vec{w}_r}}_{\vec{x}_i^T} \text{ pro } r \neq c(i)$$

$$\frac{\partial L_i}{\partial \vec{w}_r} = \sum_{j \neq c(i)} \underbrace{\frac{\partial L_i}{\partial v_{ji}}}_{\delta(v_{ji}>0)} \underbrace{\frac{\partial v_{ji}}{\partial \vec{w}_r}}_{-\vec{x}_i^T} \text{ pro } r = c(i)$$

- a tedy

$$\frac{\partial L_i}{\partial \vec{w}_r} = \begin{cases} \delta(v_{ri} > 0)\vec{x}_i^T & \text{pro } r \neq c(i) \\ -\sum_{j \neq r} \delta(v_{ji} > 0) x_i^T & \text{pro } r = c(i) \end{cases}$$

Klasifikace do více tříd - perceptron

- pro perceptron se také uvažuje ztrátová funkce tvaru

$$L_i\left(\mathbb{W}, \vec{b}, \mathbb{X}\right) = \max_{j \neq c(i)} \max\left\{\vec{x}_i^T\left(\vec{w}_j - \vec{w}_{c(i)}\right) + (b_j - b_{c(i)}), 0\right\},$$

- gradient pak má jednodušší tvar

$$\frac{\partial L_i}{\partial \vec{w}_r} = \begin{cases} \vec{x}_i^T & \text{pro nejchybnější klasifikaci } r \neq c(i) \\ -x_i^T & \text{pro } r = c(i) \\ 0 & \text{jinak} \end{cases}$$

Klasifikace do více tříd - SVM

- podstata SVM je v tom, že penalizují i správně klasifikované, ale jsou blízko dělící hranice
- ztrátovou funkci tedy opravíme na

$$L_i(\mathbb{W}, \mathbb{X}) = \sum_{j \neq c(i)} \max\left\{\vec{x}_i^T\left(\vec{w}_j - \vec{w}_{c(i)}\right) + 1, 0\right\}$$

- a celkově máme

$$L(\mathbb{W}, \mathbb{X}) = \sum_{i=1}^{N} \sum_{j \neq c(i)} \max\left\{\vec{x}_i^T\left(\vec{w}_j - \vec{w}_{c(i)}\right) + 1, 0\right\}$$

- jde o tzv. Westonův-Watkinsonův SVM, obdoba hinge-loss SVM
- obdoba L_2-loss SVM pak je

$$L(\mathbb{W}, \mathbb{X}) = \sum_{i=1}^{N} \sum_{j \neq c(i)} \max\left\{\vec{x}_i^T\left(\vec{w}_j - \vec{w}_{c(i)}\right) + 1, 0\right\}^2$$

Klasifikace do více tříd - SVM

- označme

$$v_{ji} = \vec{x}_i^T \left(\vec{w}_j - \vec{w}_{c(i)} \right) + 1$$

- pak lze psát

$$L_i \left(\mathbb{W}, \vec{b}, \mathbb{X} \right) = \sum_{j \neq c(i)} \max \left\{ v_{ji}, 0 \right\}$$

- a pro gradient dostáváme

$$\frac{\partial L_i}{\partial \vec{w}_r} = \sum_{j=1}^{k} \underbrace{\frac{\partial L_i}{\partial v_{ji}}}_{\delta(v_{ji}>0)} \underbrace{\frac{\partial v_{ji}}{\partial \vec{w}_r}}_{\vec{x}_i^T}$$

- a tedy

$$\frac{\partial L_i}{\partial \vec{w}_r} = \begin{cases} \delta(v_{ri} > 0) \vec{x}_i^T & \text{pro } r \neq c(i) \\ -\sum_{j \neq r} \delta(v_{ji} > 0) x_i^T & \text{pro } r = c(i) \end{cases}$$

Klasifikace do více tříd - logistická regrese

- u logistické regrese počítáme pravděpodobnost, že $\vec{x}_i$ patří do třídy $\mathcal{C}_k$ takto

$$P(\mathcal{C}_k \mid \mathbb{W}, \vec{x}_i) = \frac{\exp\left(\vec{x}_i^T \vec{w}_k\right)}{\sum_{j=1}^{K} \exp\left(\vec{x}_i^T \vec{w}_j\right)}$$

- věrohodnostní funkce pak má tvar

$$p(\mathbb{T} \mid \mathbb{W}, \mathbb{X}) = \prod_{k=1}^{K} \prod_{i=1}^{N} p(\mathcal{C}_k | \mathbb{W}, \vec{x}_i)^{t_{k,i}},$$

kde

$$\mathbb{T} = (\vec{t}_1, \ldots, \vec{t}_K)$$

Klasifikace do více tříd - logistická regrese

- po logaritmování a změně znaménka [1] dostáváme

$$L(\mathbb{W}, \mathbb{X}) = -\sum_{k=1}^{K}\sum_{i=1}^{N} t_{k,i} \log p(\mathcal{C}_k|\mathbb{W}, \vec{x}_i) = -\sum_{k=1}^{K} E_{t_k} \log p(\mathcal{C}_k|\mathbb{W}, \vec{x}_i)$$

- jde tedy opět o ztrátovou funkci
- odvodíme nyní vztah pro gradient

[1] od maximalizace chceme přejít k minimalizaci

Klasifikace do více tříd - logistická regrese

- označíme-li $a_{ki} = \vec{x}_i^T \vec{w}_k$, pak lze psát

$$P(\mathcal{C}_k \mid \mathbb{W}, \vec{x}_i) = p_{ki} = \frac{\exp(a_{ki})}{\sum_{j=1}^{K} \exp(a_{ji})}$$

- dále platí

$$\begin{aligned}
\frac{\partial p_{ki}}{\partial a_{ki}} &= \frac{\exp(a_{ki}) \sum_{j=1}^{K} \exp(a_{ji}) - \exp(a_{ki}) \exp(a_{ki})}{\left(\sum_{j=1}^{K} \exp(a_{ji})\right)^2} \\
&= \frac{\exp(a_{ki})}{\sum_{j=1}^{K} \exp(a_{ji})} \frac{\sum_{j=1}^{K} \exp(a_{ji}) - \exp(a_{ki})}{\sum_{j=1}^{K} \exp(a_{ji})} \\
&= p_{ki}(1 - p_{ki})
\end{aligned}$$

Klasifikace do více tříd - logistická regrese

- a pro $j \neq k$ pak platí

$$\frac{\partial p_{ki}}{\partial a_{ji}} = \frac{0 - \exp(a_{ki}) \exp(a_{ji})}{\left(\sum_{j=1}^{K} \exp\left(a_{ji}\right)\right)^2} = -p_{ki} p_{ji}$$

- nyní označíme

$$L_i\left(\mathbb{W}, \vec{b}, \vec{x}_i\right) = -\sum_{k=1}^{K} t_{k,i} \log p(\mathcal{C}_k | \mathbb{W}, \vec{x}_i)$$

Klasifikace do více tříd - logistická regrese

- dále dostáváme

$$\begin{aligned}\frac{\partial L_i}{\partial a_{ji}} &= -\sum_{k=1}^{K} \frac{t_{k,i}}{p_{ki}} \frac{\partial p_{ki}}{\partial a_{ji}} = \sum_{k=1,k\neq j}^{K} \frac{t_{k,i}}{p_{ki}} p_{ki} p_{ji} - \frac{t_{j,i}}{p_{ji}} p_{ji}(1-p_{ji}) \\ &= \sum_{k=1,k\neq j}^{K} t_{k,i} p_{ji} - t_{j,i}(1-p_{ji}) = \underbrace{\sum_{k=1}^{K} t_{k,i}}_{=1} p_{ji} - t_{j,i} = p_{ji} - t_{j,i}\end{aligned}$$

Example 5

- vygenerujte data pomocí skriptu

```
multiclass-classification-data.py
```

- pomocí parametru `overlap_parameter` lze měnit překryv dat
- proveďte výpočet lineární regrese s pomocí skriptu

```
multiclas-classification.py
```

Popis variant SVM zvané C-SVM a X-SVM lze najít v článku:
X-SVM: An Extension of C-SVM Algorithm for Classification of High-Resolution Satellite Imagery