Pokročilé metódy analýzy dát 1

úvod do hlbokého učenia

Peter Bednár

Prerekvizity

Prerekvizity 1

Označenia

- $\sum_{i=1}^{N} a_i$ – suma, $\prod_{i=1}^{N} a_i$ – súčin
- $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ – vektor reálnych čísel, $\mathbf{A}$ – matica (riadky x stĺpce), $\boldsymbol{x}^T, \mathbf{A}^T$ – transponovanie vektora alebo matice
- $\boldsymbol{x}^T \boldsymbol{y} = \sum_{i=1}^{N} x_i y_i$ – skalárny súčin
- N – počet príkladov, M – počet atribútov, K – počet tried pri klasifikácii, $(\boldsymbol{x}_i, y_i)$ – trénovacie príklady

Vlastnosti logaritmu

- $\log\left(\prod_{i=1}^{N} a_i\right) = \sum_{i}^{N} \log(a_i)$, $\log\left(a^b\right) = b \log(a)$

Prerekvizity 2

Numerická optimalizácia

- Pre zadanú funkciu $f(\boldsymbol{x})$ hľadáme také hodnoty parametrov $\boldsymbol{x}$, pre ktoré nadobúda funkcia maximálnu, alebo minimálnu hodnotu
  - $\mathrm{argmax}_{\boldsymbol{x}} f(\boldsymbol{x})$ – maximalizácia
  - $\mathrm{argmin}_{\boldsymbol{x}} f(\boldsymbol{x})$ – minimalizácia
- Metódy sú najčastejšie navrhnuté pre minimalizáciu, pre maximalizovanie platí:

$$\mathrm{argmax}_{\boldsymbol{x}} f(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{x}} - f(\boldsymbol{x})$$

Prerekvizity 3

- Ak vynásobíme funkciu kladným číslom a, optimálne parametre sa nezmenia:

$$\text{argmax}_{\boldsymbol{x}} af(\boldsymbol{x}) = \text{argmax}_{\boldsymbol{x}} f(\boldsymbol{x})$$

$$\text{argmin}_{\boldsymbol{x}} af(\boldsymbol{x}) = \text{argmin}_{\boldsymbol{x}} f(\boldsymbol{x})$$

- Podobne, ak transformujeme funkciu monotónne rastúcou funkciou, napr. logaritmom:

$$\text{argmax}_{\boldsymbol{x}} \log(f(\boldsymbol{x})) = \text{argmax}_{\boldsymbol{x}} f(\boldsymbol{x})$$

Prerekvizity 4

Derivácia a gradient

- Gradient funkcie $f(\boldsymbol{x}), \boldsymbol{x} = (x_1, x_2, \dots, x_M)$ je vektor parciálnych derivácií $\nabla f(\boldsymbol{x}) = \left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \frac{\partial f(\boldsymbol{x})}{\partial x_2}, \dots, \frac{\partial f(\boldsymbol{x})}{\partial x_M}\right)$, tzn. gradient je vektorová funkcia a vieme ho vypočítať pre daný bod $\boldsymbol{x}$

Príklad

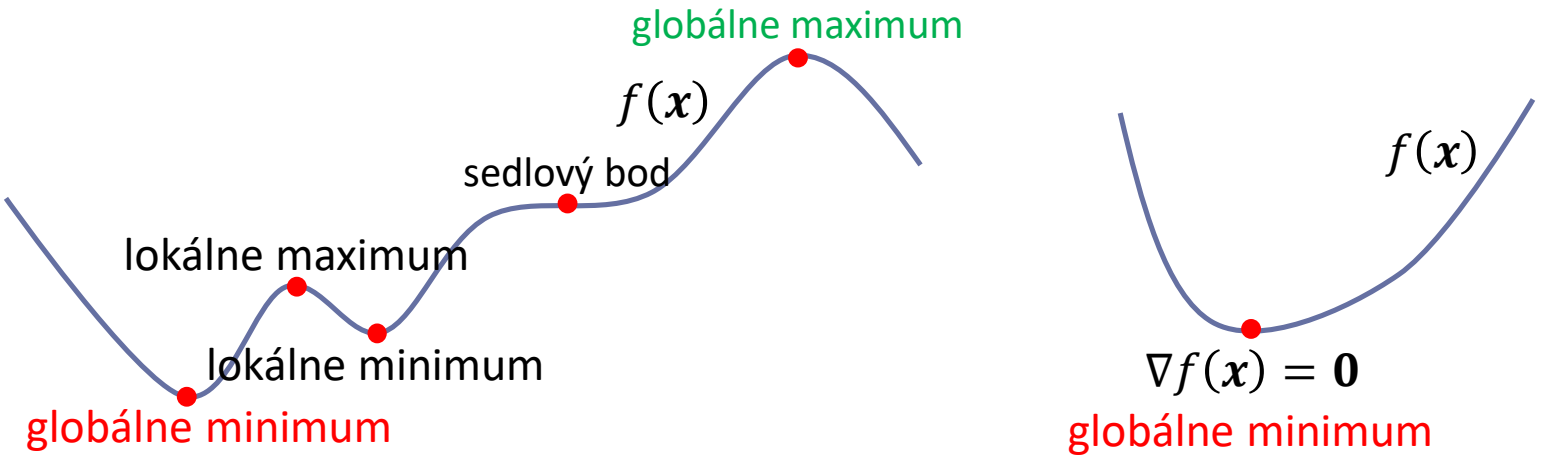
$$f(\boldsymbol{x}) = x_1^2 + 3x_2^2 + 2x_1x_2$$
$$\nabla f(\boldsymbol{x}) = (2x_1 + 2x_2, 6x_2 + 2x_1)$$

Po dosadení gradient v bode $\boldsymbol{x} = [1, 2]$

$$\nabla f([1, 2]) = (5, 10)$$

Prerekvizity 5

- Body, pre ktoré sa gradient rovná nulovému vektoru $\nabla f(\boldsymbol{x}) = \mathbf{0}$ sa označujú ako stacionárne – globálne minimum je jeden z nich



- Zvláštny význam pri optimalizácii majú konvexné funkcie, pre ktoré platí, že stacionárny bod zároveň určuje globálne minimum

Prerekvizity 6

Podmienená pravdepodobnosť

- P(Y|X) udáva pravdepodobnosť, že nastane jav Y za predpokladu, že nastal jav X
  - P(Y|X) = P(Y,X) / P(X)
- Podmienená pravdepodobnosť $P(Y=y \mid X_1=x_1, X_2=x_2, ..., X_M=x_M)$ má význam pre predpovedanie hodnoty cieľového atribútu $Y=y$ na základe známych hodnôt vstupných atribútov $X_1 = x_1, X_2 = x_2, ..., X_M = x_M$
  - Pre nový prípad zvolíme cieľovú hodnotu, pre ktorú je podmienená pravdepodobnosť maximálna

Prerekvizity 7

Príklad

- Chceme predpovedať, či študent príde na cvičenie ráno keď prší

Študent	Počasie	Rozvrh	Počet prípadov
príde	slnečno	ráno	65
príde	slnečno	po obede	60
príde	prší	ráno	20
príde	prší	po obede	45
nepríde	slnečno	ráno	10
nepríde	slnečno	po obede	5
nepríde	prší	ráno	30
nepríde	prší	po obede	15
		celkovo	250

Prerekvizity 8

$$P(\text{Študent} = \text{príde} \mid \text{Počasie} = \text{prší}, \text{Rozvrh} = \text{ráno})$$
$$= \frac{P(\text{Študent} = \text{príde}, \text{Počasie} = \text{prší}, \text{Rozvrh} = \text{ráno})}{P(\text{Počasie} = \text{prší}, \text{Rozvrh} = \text{ráno})}$$
$$= \frac{20/250}{(20+30)/250} = 0.4$$

$$P(\text{Študent} = \text{nepríde} \mid \text{Počasie} = \text{prší}, \text{Rozvrh} = \text{ráno})$$
$$= 1 - P(\text{Študent} = \text{príde} \mid \text{Počasie} = \text{prší}, \text{Rozvrh} = \text{ráno})$$
$$= 0.6$$

- Predpovedáme, že študent nepríde s pravdepodobnosťou 0.6

Učenie prediktívnych modelov

Učenie prediktívnych modelov (1)

- Pri predikcii sa snažíme na základe známych atribútov **X** = (X_1, X_2, ..., X_M) predpovedať hodnotu cieľového atribútu Y

- Predpokladáme, že medzi atribútmi **X** a Y existuje nejaká závislosť, ktorú však vo všeobecnosti nepoznáme, k dispozícii máme iba tzv. trénovaciu množinu príkladov $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, ..., $(\boldsymbol{x}_N, y_N)$ pre ktorú poznáme hodnoty vstupných atribútov $\boldsymbol{x}$ a skutočnú hodnotu cieľového atribútu y

Učenie prediktívnych modelov (2)

- **Úloha:** na základe trénovacej množiny príkladov nájsť funkciu $f(\boldsymbol{x})$, ktorá pre zadané hodnoty vstupných atribútov $\boldsymbol{x}$ vypočíta čo najlepší odhad skutočnej hodnoty cieľového atribútu y vo všeobecnosti pre všetky nové prípady

Príklad regresie

- Máme realitnú kanceláriu a na základe vstupných atribútov:
  - X_1 – veľkosť úžitkovej plochy v m^2
  - X_2 – počet izieb
  - X_3 – navrhovaná cena
- Chceme predpovedať, za ako dlho predáme pre nového zákazníka jeho nehnuteľnosť na trhu:
  - Y – počet dní od zverejnenia ponuky do predaja
- Máme historické dáta napr. za posledný rok

Chybová funkcia

- Pre danú úlohu sa vyhodnocuje zhoda medzi skutočnou hodnotou y a vypočítanou predikciou podľa danej chybovej funkcie $L(y, f(\boldsymbol{x}))$
- Väčšinou sa rovná 0 ak sa y rovná $f(\boldsymbol{x})$ a rastie s narastajúcim rozdielom medzi y a $f(\boldsymbol{x})$
- **Príklady:**
- 1/0 chyba (klasifikačná chyba):

$$L(y, f(\boldsymbol{x})) = 0\ ak\ y = f(\boldsymbol{x}), inak\ 1$$

- Kvadratická chyba:

$$L(y, f(\boldsymbol{x})) = \frac{1}{2}(y - f(\boldsymbol{x}))^2$$

Ako navrhnúť algoritmus učenia?

- **Základný princíp:**
  - Nevieme spočítať chybu predikcie vo všeobecnosti, pretože nepoznáme cieľovú hodnotu pre všetky prípady
  - Preto sa pokúsime nájsť funkciu, ktorá bude mať minimálnu chybu aspoň na trénovacích dátach $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, ..., $(\boldsymbol{x}_N, y_N)$
  - tzn. zovšeobecnenú chybu sme nahradili trénovacou - tzv. empirickou chybou:

$$J(f) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(\boldsymbol{x}_i))$$

Algoritmus učenia = Chybová funkcia + Model + Optimalizačná metóda

1. Zvolíme si model:
   - Matematicky je to množina funkcií, z ktorých algoritmus vyberá výslednú funkciu $f(\boldsymbol{x})$

Učenie je teda optimalizačná úloha – z modelu vyberieme funkciu, pre ktorú bude chyba na trénovacích dátach minimálna

2. Zvolíme si optimalizačnú metódu
   - Napr. ak je model definovaný ako množina funkcií daného tvaru, ktoré majú okrem vstupných atribútov množinu parametrov W = {W_1, W_2, ...}, zvolíme si metódu, ktorá vypočíta optimálne parametre $\boldsymbol{w}$, pri ktorých je chyba funkcie $f(\boldsymbol{x}, \boldsymbol{w})$ minimálna

Príklad – lineárna regresia (1)

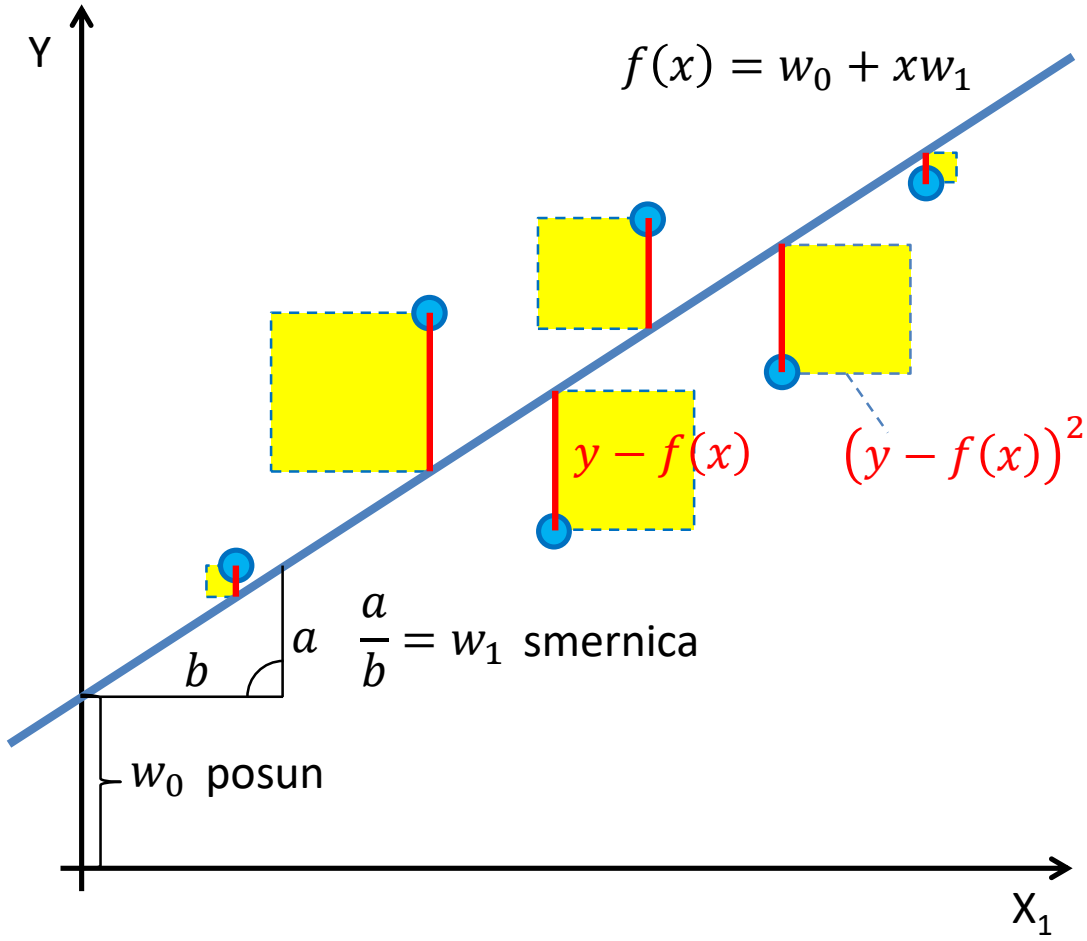
- Úlohou je regresia - cieľový atribút Y je číselný
- Minimalizujeme kvadratickú chybovú funkciu

$$J(f) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(y_i - f(\boldsymbol{x}_i))^2$$

- Lineárny model – funkcie majú tvar lineárnej funkcie s parametrami $\boldsymbol{w} = (w_0, w_1, \ldots, w_M)$

$$f(\boldsymbol{x}, \boldsymbol{w}) = w_0 + x_1 w_1 + \cdots + x_M w_M$$

Príklad – lineárna regresia (2)



Príklad – lineárna regresia (3)

- Optimalizujeme parametre $\boldsymbol{w}$, hodnoty trénovacích príkladov $\boldsymbol{x}$, y sú konštanty
- Pri lineárnej regresii vieme priamo vypočítať optimálne parametre aj analyticky:
  1. Zderivujeme $J(f)$ podľa $w_0, w_1, \ldots, w_M$ a dostaneme M+1 funkcií
  2. Položíme derivácie = 0 a dostaneme sústavu M+1 rovníc, z ktorých si vypočítame optimálne parametre

- Algoritmus učenia lineárnej regresie = algoritmus riešenia sústavy rovníc (nie však vo všeobecnosti, pozor na lineárne závislé atribúty!)

Príklad – logistická regresia (1)

- Úlohou je klasifikácia do dvoch tried Y = {0, 1}
- Minimalizujeme 1/0 chybovú funkciu:

$$L\big(y, f(\boldsymbol{x})\big) = 0 \; ak \; y = f(\boldsymbol{x}), inak \; 1$$

- Trénovacia chyba $J(f)$ teda udáva pravdepodobnosť chybnej klasifikácie
- Problém je, že 1/0 chybová funkcia sa nedá dobre derivovať (nedá sa priamo numericky optimalizovať)
- Preto budeme minimalizovať jej aproximáciu

Príklad – logistická regresia (2)

- Model definujeme pomocou podmienených pravdepodobností
  - P(Y = 1 | X = $\boldsymbol{x}$) – pravdepodobnosť, že má byť príklad s hodnotami vstupných atribútov $\boldsymbol{x}$ zaradený do triedy 1
  - P(Y = 0 | X = $\boldsymbol{x}$) = 1 - P(Y = 1 | X = $\boldsymbol{x}$) – pravdepodobnosť, že má byť zaradený do triedy 0

- 1/0 chybová funkcia bude minimálna, ak zaradíme príklad do najpravdepodobnejšej triedy, tzn. pre dve triedy {0,1} zaradíme príklad do triedy 1 ak P(Y = 1 | X = $\boldsymbol{x}$) > 0.5, inak do triedy 0

Príklad – logistická regresia (3)

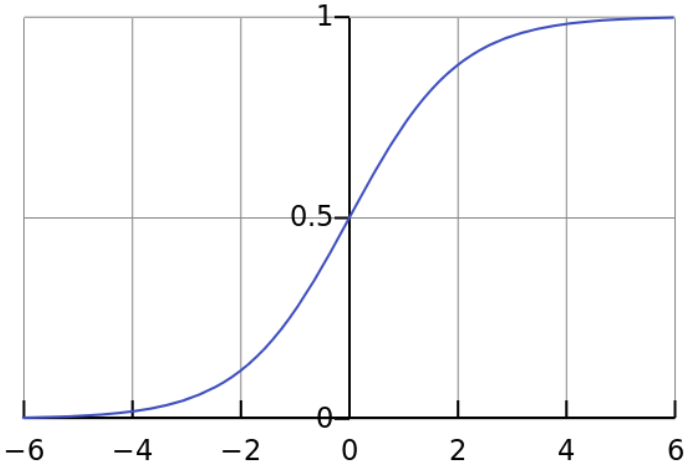
- Pravdepodobnosť P(Y = 1 | X = $\boldsymbol{x}$) (a teda ani P(Y = 0 | X = $\boldsymbol{x}$)) nepoznáme
- Model logistickej regresie si navrhneme ako funkciu $p(\boldsymbol{x})$, pomocou ktorej je možné priamo odhadnúť pravdepodobnosť zaradenia do triedy 1, $p_1(\boldsymbol{x})$ ≈ P(Y = 1 | X = $\boldsymbol{x}$), pričom chceme, aby bol tento odhad založený na lineárnej funkcii

$$f(\boldsymbol{x}, \boldsymbol{w}) = w_0 + x_1 w_1 + \cdots + x_M w_M$$

- Pravdepodobnosť pre triedu 0 potom môžeme odhadnúť ako $p_0(\boldsymbol{x}) = 1 - p_1(\boldsymbol{x})$

Príklad – logistická regresia (4)

- Keďže lineárna funkcia môže nadobúdať ľubovoľnú reálnu hodnotu
  - Pre odhad pravdepodobnosti potrebujeme hodnotu predikcie $f(\boldsymbol{x}, \boldsymbol{w})$ previesť do intervalu (0,1)
  - Pri logistickej regresii sa na transformovanie predikcie používa logistická funkcia



$$g(t) = \frac{1}{1+e^{-t}}$$

Príklad – logistická regresia (5)

- Odhad pravdepodobnosti pre triedu 1 dostaneme po dosadení lineárnej funkcie do logistickej

$$p_1(\boldsymbol{x}) = g\big(f(\boldsymbol{x}, \boldsymbol{w})\big) = g(w_0 + x_1 w_1 + \cdots + x_M w_M)$$

- Ako vypočítame parametre $w_0, w_1, \ldots, w_M$?
- Keďže $p_1(\boldsymbol{x})$ je odhad pravdepodobnosti, môžeme použiť metódu maximálnej vierohodnosti (*max. likelihood*)

1. Vyjadríme si pravdepodobnosť pre celú trénovaciu množinu (tzv. funkciu vierohodnosti $l(\boldsymbol{w})$) ako súčin pravdepodobností jednotlivých príkladov, tzn. pre príklady z triedy 1 dosadíme priamo $p_1(\boldsymbol{x})$ a pre príklady z triedy 0 dosadíme $1 - p_1(\boldsymbol{x})$
2. Zvolíme také parametre, pre ktoré je funkcia vierohodnosti maximálna

Príklad – logistická regresia (6)

- Keďže $y_i = \{0, 1\}$, celkovú funkciu vierohodnosti môžeme skrátene vyjadriť ako súčin:

$$l(\boldsymbol{w}) = \prod_{i=1}^{N} p_1(\boldsymbol{x}_i)^{y_i}(1 - p_1(\boldsymbol{x}_i))^{1-y_i}$$

  - Ak $y_i = 1$ $p_1(\boldsymbol{x}_i)^1(1 - p_1(\boldsymbol{x}_i))^0 = p_1(\boldsymbol{x}_i)$
  - Ak $y_i = 0$ $p_1(\boldsymbol{x}_i)^0(1 - p_1(\boldsymbol{x}_i))^1 = 1 - p_1(\boldsymbol{x}_i) = p_0(\boldsymbol{x}_i)$

- Parametre $\boldsymbol{w}$ môžeme vypočítať ľubovoľnou metódou ktorá maximalizuje funkciu vierohodnosti $l(\boldsymbol{w})$

Príklad – logistická regresia (7)

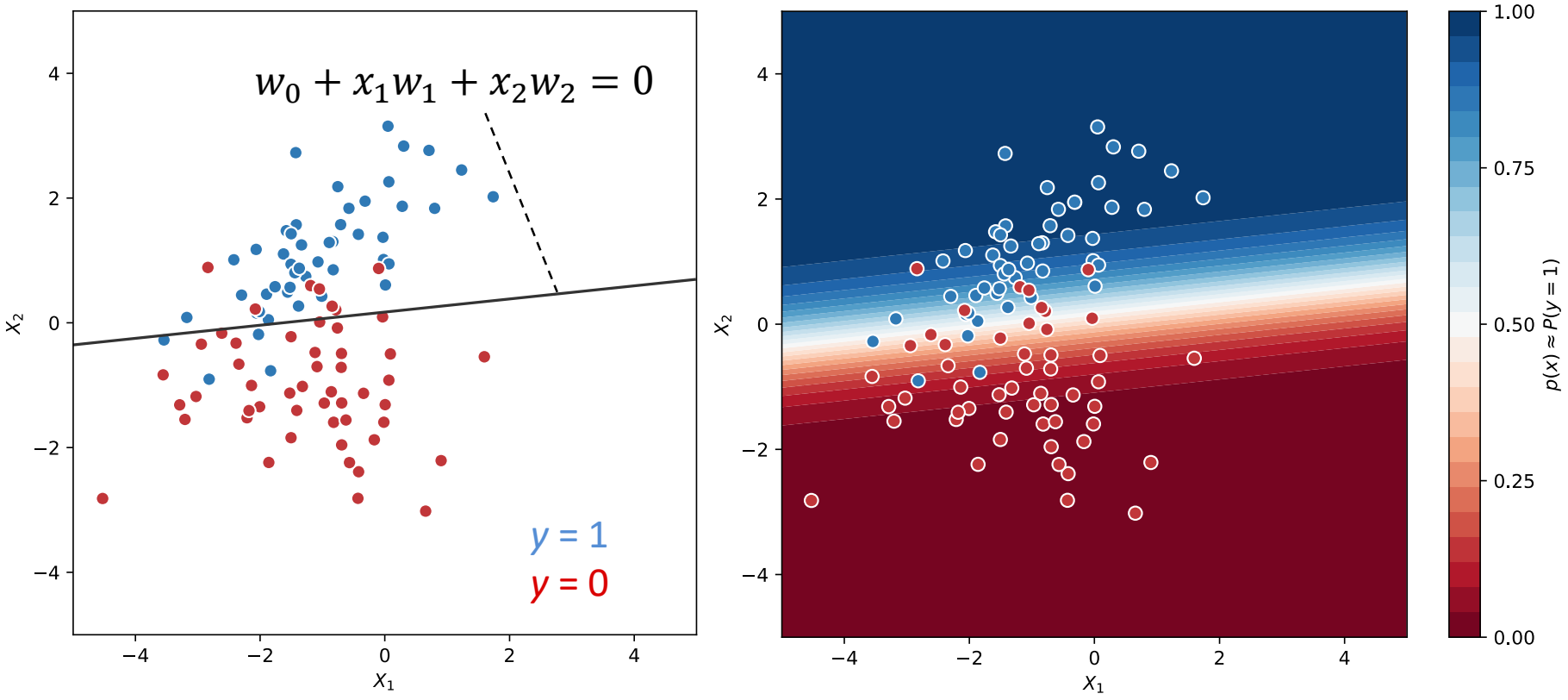
- Pre prahovú hodnotu 0.5 sa dá funkcia $p_1(\boldsymbol{x})$ zobraziť aj geometricky ako deliaca (hyper)plocha oddeľujúca príklady jednotlivých tried
- Z priebehu logistickej funkcie platí, že $g(t) = 0.5$ ak $t = 0$, tzn.

$$g(w_0 + x_1 w_1 + \cdots + x_M w_M) = 0.5 \text{ ak}$$

$$w_0 + x_1 w_1 + \cdots + x_M w_M = 0$$

- Pre logistickú regresiu je deliaca hyperplocha lineárna (priamka pre X_1 X_2, rovina pre X_1 X_2 X_3, ... hyperrovina vo všeobecnosti) - lineárny klasifikátor

Príklad – logistická regresia (8)



Zobrazenie deliacej priamky modelu logistickej regresie a kontúr funkcie $p_1(\boldsymbol{x})$ - odhadu pravdepodobnosti pre triedu 1

Príklad – logistická regresia (9)

- Akú chybovú funkciu minimalizuje logistická regresia?
  - V $l(\boldsymbol{w})$ máme súčin členov pre jednotlivé príklady, pre chybovú funkciu potrebujeme súčet
  - Chybovú funkciu minimalizujeme, nie maximalizujeme

1. Súčin pravdepodobností v $l(\boldsymbol{w})$ pre jednotlivé príklady sa dá nahradiť za sumu pomocou logaritmu
   - $[\log(a.b.c\ \ldots) = \log(a) + \log(b) + \log(c) + \ldots]$
   - Funkcia $\log\big(l(\boldsymbol{w})\big)$ má maximum v rovnakom bode ako $l(\boldsymbol{w})$, takže sa optimálne riešenie $\boldsymbol{w}$ nezmení
2. Pre minimalizovanie zmeníme znamienko funkcie na
   - $\log\big(l(\boldsymbol{w})\big)$

Príklad – logistická regresia (10)

- Tzn. logistická regresia minimalizuje záporná logaritmická funkciu vierohodnosti $-\log\big(l(\boldsymbol{w})\big)$ – *negative log likelihood*
- Pre jeden príklad má po úprave výsledná chybová funkcia tvar:

$$L\big(y_i, p(\boldsymbol{x}_i)\big) = -\big[y_i \log(p_1(\boldsymbol{x}_i)\,) + (1 - y_i)\log\big(1 - p_1(\boldsymbol{x}_i)\big)\big]$$

– $[\log(a^b) = b.\log(a)]$

- Táto funkcia sa označuje aj ako krížová entropia
  - Pri jej minimalizovaní na celej trénovacej množine sa snažíme čo najviac zvýšiť odhad pravdepodobnosti $p_1(\boldsymbol{x}_i)$ pre príklady z triedy 1 a $1 - p_1(\boldsymbol{x}_i) = p_0(\boldsymbol{x}_i)$ pre príklady z triedy 0

Príklad – logistická regresia (11)

- Krížovú entropiu je možné zovšeobecniť pre ďalšie modely a pre klasifikáciu do K tried

$$L\big(y_i, f(\boldsymbol{x}_i)\big) = -\sum_{k=1}^{K} y_{i,k} \log(p_k(\boldsymbol{x}_i))$$

- Kde $y_{i,k} = 1$ ak príklad i patrí do triedy k, inak 0 a $p_k(\boldsymbol{x}_i)$ je odhad pravdepodobnosti modelu pre triedu k
- Ak môže byť príklad zaradený iba do jednej z K tried, pre odhad pravdepodobnosti musí platiť $0 \leq p_k(\boldsymbol{x}_i) \leq 1$, $\sum_{k=1}^{K} p_k(\boldsymbol{x}_i) = 1$

Príklad – logistická regresia (12)

- **Príklad**
- Priradenie triedy a predikciu môžeme považovať za dve pravdepodobnostné distribúcie pričom chceme dosiahnuť, aby sa čo najviac podobali pre všetky príklady.

$y_{i,1}$	0	0.001	$p_1(\boldsymbol{x}_i)$
$y_{i,2}$	0	0.0015	$p_2(\boldsymbol{x}_i)$
	...	...	
$y_{i,k}$	1	0.78	$p_k(\boldsymbol{x}_i)$
	...	...	
$y_{i,K}$	0	0	$p_K(\boldsymbol{x}_i)$

- Pri učení sa snažíme čo najviac zvýšiť odhad pre správnu triedu *k* (resp. znížiť odhad pre všetky ostatné triedy, keďže platí ohraničenie $\sum_{k=1}^{K} p_k(\boldsymbol{x}_i) = 1$)

Aký algoritmus vybrať? (1)

- Predpokladajme, že závislosť medzi **X** a Y najpresnejšie popisuje funkcia $F^*(\boldsymbol{x})$, tzn. funkcia, ktorá má minimálnu zovšeobecnenú chybu na všetkých možných prípadoch
- $F^*(\boldsymbol{x})$ je teda naša cieľová funkcia, ktorú hľadáme

- Ak si zvolíme model s obmedzenými funkciami (napr. lineárnymi), aj najlepšia možná funkcia $f^*(\boldsymbol{x})$, ktorá sa v ňom bude nachádzať bude veľmi odlišná od cieľovej funkcie $F^*(\boldsymbol{x})$ a teda bude mať veľkú chybu vo všeobecnosti

- Nedokážeme presne aproximovať zložité závislosti jednoduchou funkciou

Aký algoritmus vybrať? (2)

- Naopak, ak zvolíme veľkú množinu komplexných funkcií, je pravdepodobnejšie že sa medzi nimi bude nachádzať aj $F^*(\boldsymbol{x})$
  - Alebo sa aspoň nebude najlepšia funkcia modelu $f^*(\boldsymbol{x})$ veľmi odlišovať od $F^*(\boldsymbol{x})$ (tzn. zníži sa chyba aproximácie)
- Na druhej strane sa však zvýši variancia učenia a môže dochádzať k preučeniu

Variancia učenia a preučenie (1)

- Máme iba obmedzený počet trénovacích príkladov
- Ak zvolíme model s komplexnými funkciami - pre danú trénovaciu množinu môže existovať veľa navzájom odlišných funkcií, ktoré budú mať minimálnu chybu na trénovacích príkladoch
  - Niektoré z týchto funkcií sa však môžu výrazne odlišovať od cieľovej funkcie $F^*(\boldsymbol{x})$

- Ak algoritmus zvolí funkciu, ktorá síce má minimálnu chybu na trénovacích dátach, ale veľkú chybu vo všeobecnosti pre ostatné prípady, dochádza k tzv. preučeniu

Variancia učenia a preučenie (2)

- Možnosti preučenia sa nemôžeme nikdy principiálne vyhnúť s úplnou istotou!
- Pre danú konečnú trénovaciu množinu sa ju však môžeme pokúsiť znížiť regulovaním učenia – tzn. zvolíme si jednoduchší, viac ohraničený model

Koľko trénovacích príkladov potrebujeme? (1)

- Predpokladajme, že môžeme vybrať náhodne N trénovacích príkladov zo všetkých možných prípadov:
  - Vždy keď si zvolíme iné trénovacie príklady, algoritmus môže nájsť inú funkciu $f(\boldsymbol{x})$ s minimálnou trénovacou chybou

- Ak máme komplexný model – aj malá zmena trénovacích príkladov môže spôsobiť výber veľmi odlišnej funkcie $f(\boldsymbol{x})$ (tzn. veľkú varianciu učenia) – a teda narastá aj možnosť preučenia

Koľko trénovacích príkladov potrebujeme? (2)

- Ak budeme pridávať ďalšie trénovacie príklady:
  - Chybu bude algoritmus minimalizovať na väčšom počte príkladov, tzn. funkcia $f(\boldsymbol{x})$ by sa mala približovať k cieľovej funkcii $F^*(\boldsymbol{x})$ a chyba aproximácie by sa mala znížiť
  - Viac príkladov viac obmedzí výber rôznych funkcií $f(\boldsymbol{x})$ s minimálnou trénovacou chybou, tzn. variancia učenia sa zníži (a teda sa aj zníži možnosť preučenia)

Zovšeobecnená chyba funkcie $f(\boldsymbol{x})$ - zhrnutie

- Je chyba určená na všetkých možných prípadoch
- Môžeme ju vyjadriť rozdielom medzi naučenou funkciou $f(\boldsymbol{x})$ a neznámou cieľovou funkciou $F^*(\boldsymbol{x})$
- Nedá sa priamo určiť lebo nepoznáme $F^*(\boldsymbol{x}) = y$, ale vieme že je ovplyvnená dvoma protichodnými zložkami:

- Zovšeobecnená chyba = Chyba aproximácie (Bias) + Chyba variancie učenia

Koľko trénovacích príkladov potrebujeme? (3)

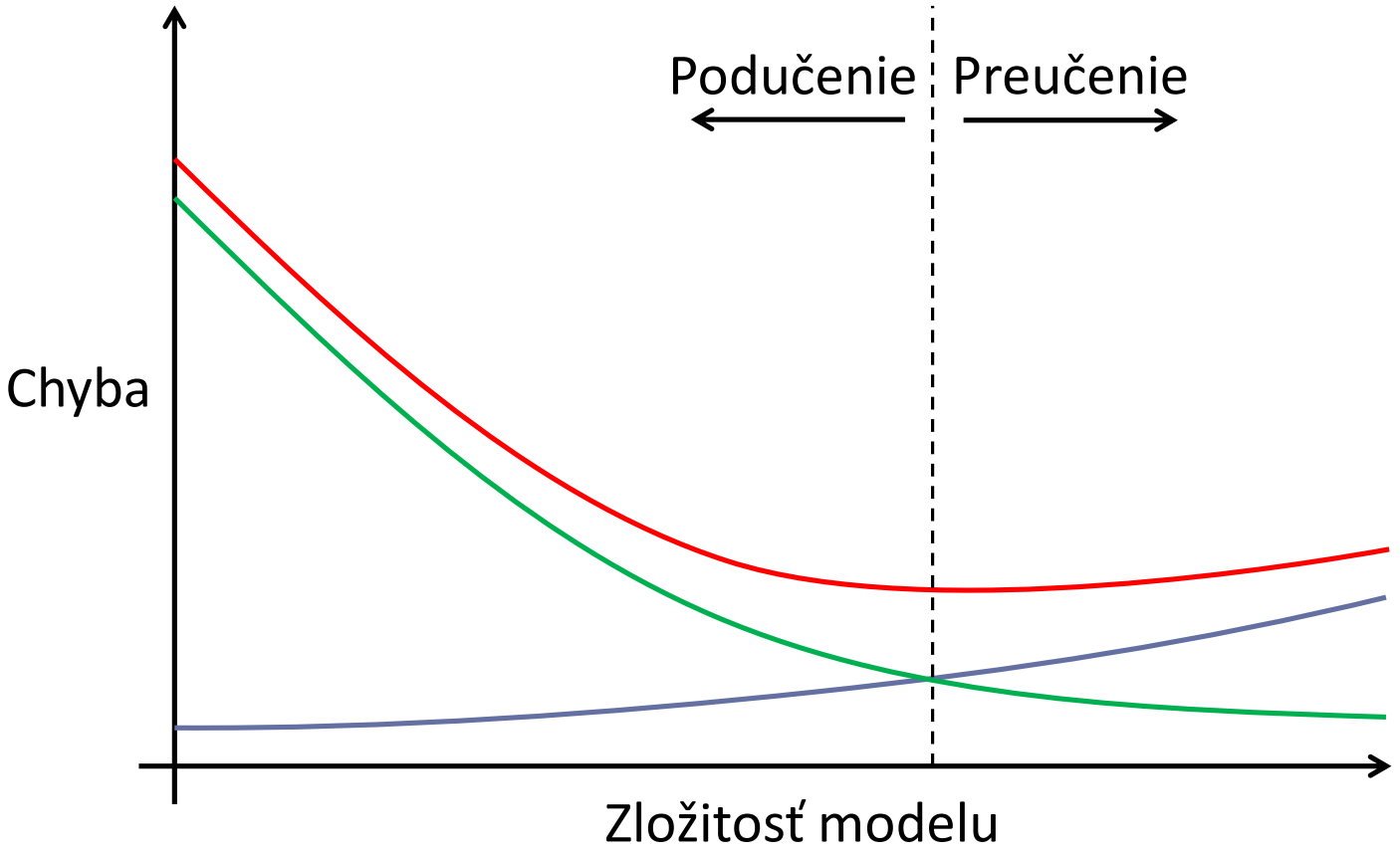
- Zovšeobecnená chyba = Chyba aproximácie (Bias) + Chyba variancie učenia

- Pre malú trénovaciu množinu môže prevládať vplyv variancie učenia a skôr môže dôjsť k preučeniu – preto uprednostníme jednoduchší model

- Pri narastajúcom počte príkladov sa bude variancia zmenšovať, takže zvolíme zložitejší model, aby sme čo najviac minimalizovali chybu aproximácie

Chyba = Bias + Variancia

Zovšeobecnená chyba = Chyba aproximácie (Bias) + Chyba variancie



Aký dobrý je náš model? (1)

- Nikdy nebudeme vedieť úplne vyhodnotiť zovšeobecnenú chybu
  - A teda nevieme ani presne určiť, ako sa odlišuje niektorá funkcia modelu $f(\boldsymbol{x})$ od $F^*(\boldsymbol{x})$

- Podľa štatistickej teórie učenia však vieme s veľkou istotou určiť, s akou chybou sa môže funkcia $f(\boldsymbol{x})$ naučená na N trénovacích príkladoch maximálne odlišovať od $f^*(\boldsymbol{x})$ – tzn. od najpresnejšej funkcie patriacej do modelu
  - Tento odhad je platný nezávisle od $F^*(\boldsymbol{x})$ a od vybranej trénovacej množiny

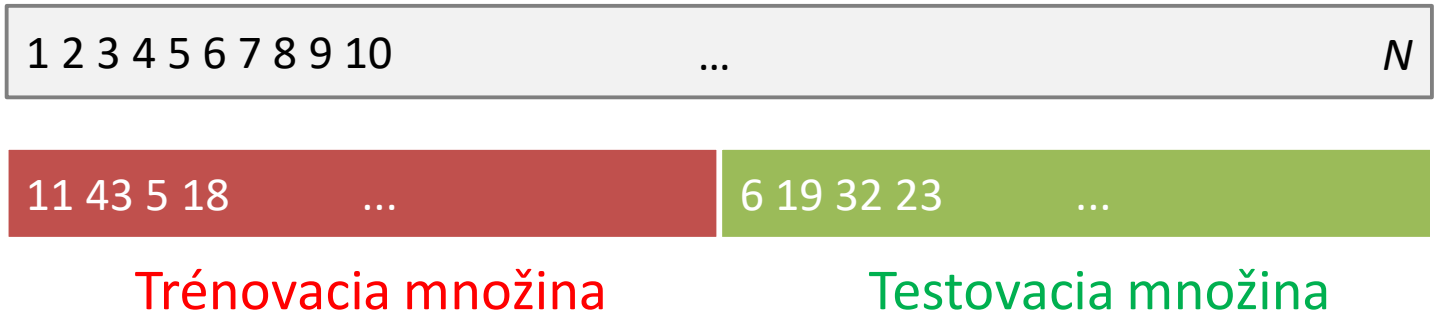
Aký dobrý je náš model? (2)

- Pre daný model však nemusíme mať istotu, že cieľová funkcia $F^*(\boldsymbol{x})$ určite patrí do modelu, resp. že sa $f^*(\boldsymbol{x})$ od nej výrazne neodlišuje
- Navyše ak chceme spresniť odhad chyby podľa štatistickej teórie, musíme uvažovať veľký počet trénovacích príkladov
- A tak isto nás väčšinou zaujíma presnosť algoritmu na danom probléme, nie na všetkých možných problémoch

- Preto sa pri praktickom riešení používa iná metóda na odhad zovšeobecnej chyby

Aký dobrý je náš model? (3)

- Náhodne si rozdelíme dáta na trénovaciu a testovaciu množinu, na trénovacej množine naučíme funkciu $f(\boldsymbol{x})$ a vypočítame jej chybu na testovacej množine



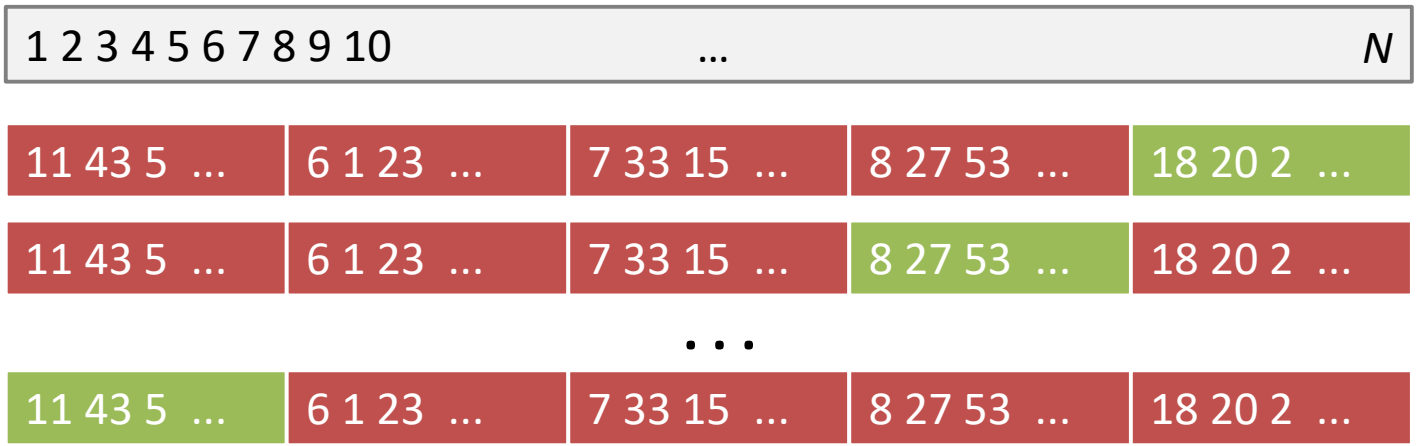
Aký dobrý je náš model? (4)

- Testovacia chyba je odhadom zovšeobecnenej chyby
- Pri inom náhodnom rozdelení sa môže testovacia chyba zmeniť:
  - Prejaví sa variancia učenia, pretože sme zmenili trénovacie dáta
  - Prejaví sa variancia náhodného výberu testovacích príkladov

- Tzn., aby sme mali čo najpresnejší odhad, musíme mať dostatočný počet príkladov aj v testovacej aj v trénovacej množine

Aký dobrý je náš model? (5)

- Krížová validácia
  - Učenie na väčšej trénovacej množine – zníži sa vplyv variancie učenia
  - Testovanie vyhodnotené na všetkých dátach – zníži sa vplyv variancie testovacej množiny



Učenie prediktívnych modelov – zhrnutie (1)

- Úloha je na základe trénovacej množiny nájsť funkciu, pomocou ktorej vieme čo najpresnejšie predpovedať hodnotu cieľového atribútu podľa vstupných atribútov vo všeobecnosti pre všetky prípady

- Algoritmus učenia = Chybová funkcia + Model + Optimalizačná metóda
- Algoritmus učenia zvolí funkciu zo zvolenej množiny funkcií – modelu tak aby sa minimalizovala chyba na trénovacích príkladoch

Učenie prediktívnych modelov – zhrnutie (2)

- Zovšeobecnená chyba = chyba aproximácie + chyba variancie učenia
  - Príliš jednoduchý model – veľká chyba aproximácie – podučenie
  - Príliš zložitý model – veľká varianca učenia – preučenie
- Zovšeobecnenú chybu vieme len odhadnúť:
  - Testovacia chyba na nezávislej náhodne vybranej testovacej množine – pozor na varianciu učenia a varianciu výberu testovacích príkladov
  - Krížová validácia

Lineárne modely na nelineárnych dátach

- Väčšina pozorovaných závislostí je nelineárna
- Transformácia pomocou bázových funkcií
  - Jednoduchý spôsob ako rozšíriť lineárny model na nelineárne dáta:
  - Zvolíme si nelineárne (tzv. bázové) funkcie B_1, B_2, …, B_K, pomocou ktorých transformuje pôvodné vstupné atribúty
  - Na transformovaných dátach naučíme lineárny model, keď ho však aplikujeme na pôvodné dáta, vieme reprezentovať aj nelineárne závislosti

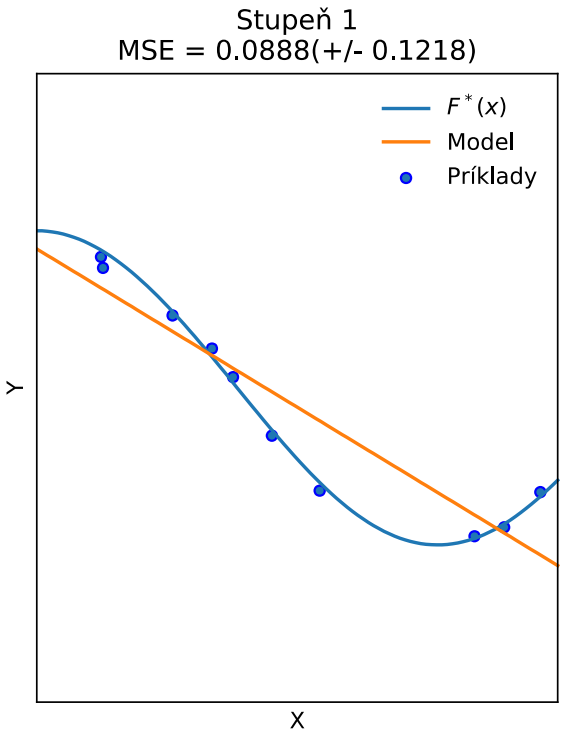
Lineárne modely na nelineárnych dátach

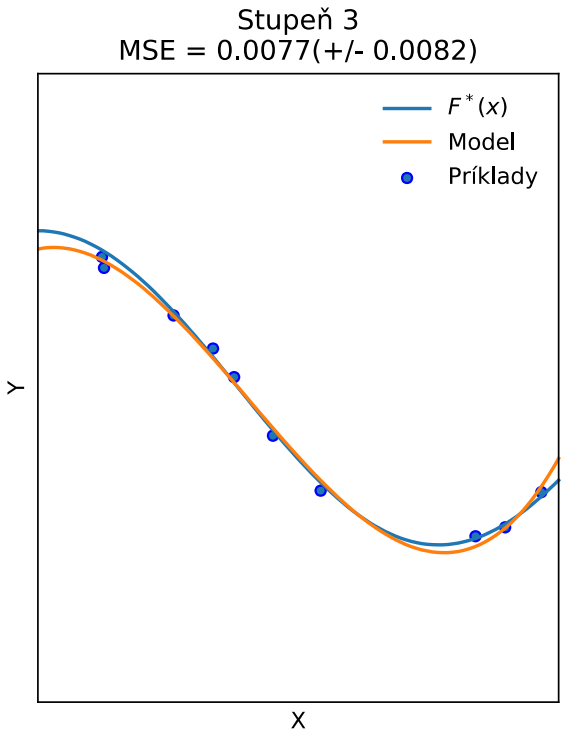
- **Príklad**:
- Polynomiálna regresia
  - Ako transformačné funkcie použijeme členy polynómu zvoleného stupňa, tzn. napr. pre dva atribúty X_1, X_2 a stupeň polynómu 2 budú transformačné funkcie $B_1 = X_1$, $B_2 = X_2$, $B_3 = X_1X_2$, $B_4 = X_1^2$, $B_5 = X_2^2$
  - Výsledný model bude mať tvar:

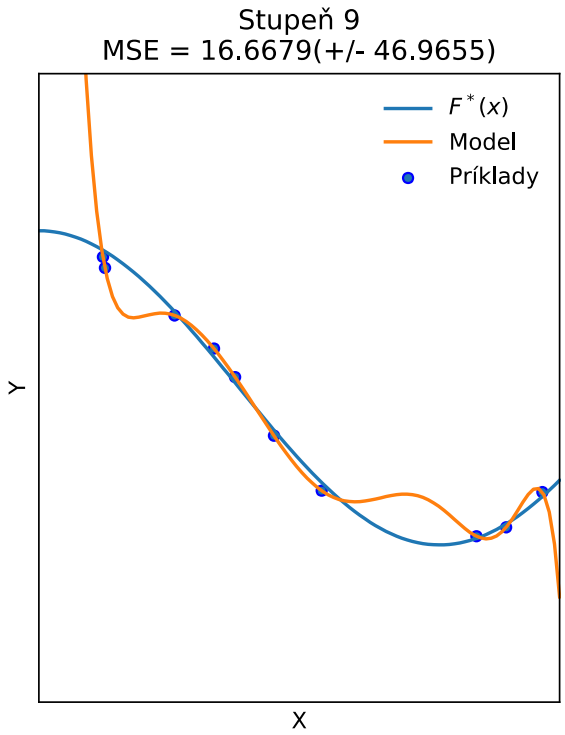
$$f(\boldsymbol{x}, \boldsymbol{w}) = w_0 + x_1 w_1 + x_2 w_2 + x_1 x_2 w_3 + x_1^2 w_4 + x_2^2 w_5$$

- Okrem členov polynómu môžeme použiť aj iné nelineárne funkcie, napr.: $x_i/x_j, \log(x_i)$, ...

Polynomiálna regresia – príklad (1)

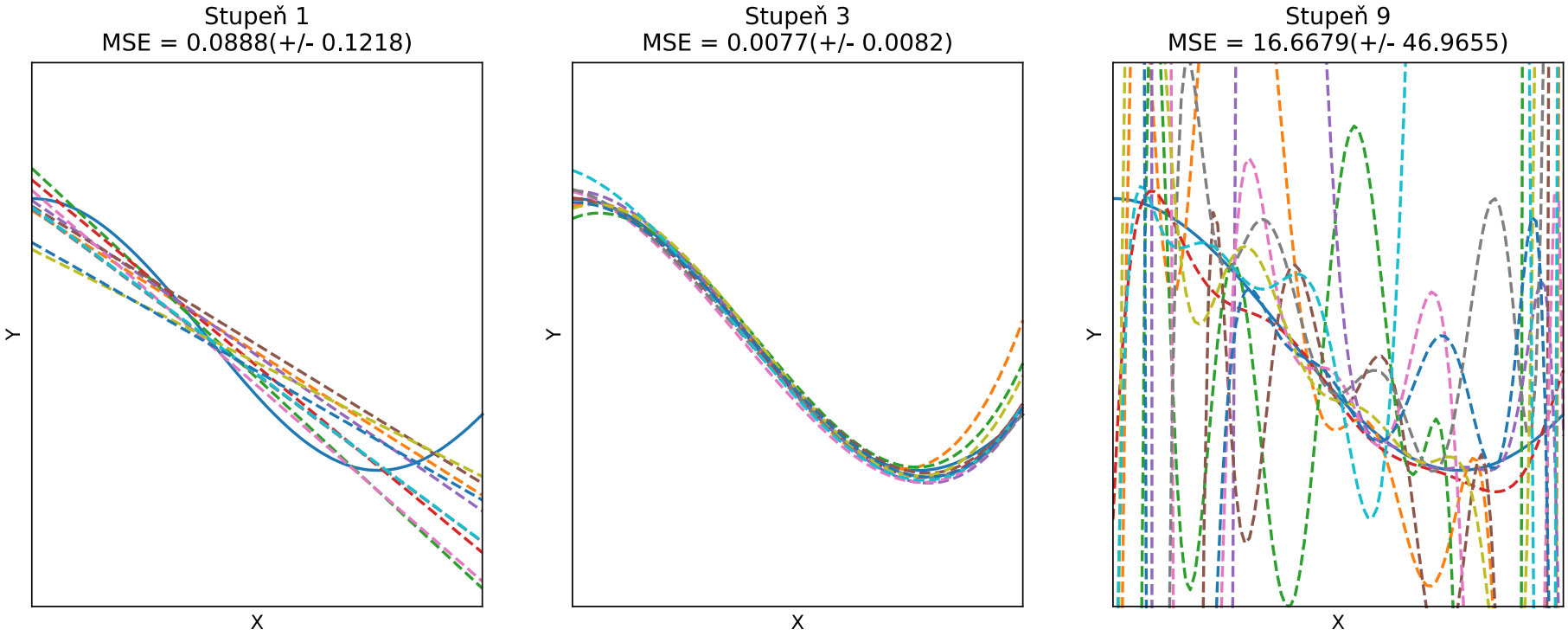






MSE je priemerná kvadratická chyba pri 10-násobnej krížovej validácii, štandardná odchýlka chyby udáva odhad variancie učenia

Polynomiálna regresia – príklad (2)



Priebeh 10 modelov naučených na náhodne vybraných trénovacích množinách s 10 príkladmi. Pri jednoduchom lineárnom modeli (stupeň 1) došlo k podučeniu, pri polynóme 9. stupňa k výraznému preučeniu