Metodika analýzy dat: Od základů po aplikace metod strojového učení

Výběr příznaků a vyhodnocení kvality modelů

Jakub Steinbach, Jan Vrba

Ústav počítačové a řídicí techniky
VŠCHT

3.10.2024

Obsah slajdů I

Bias a variance

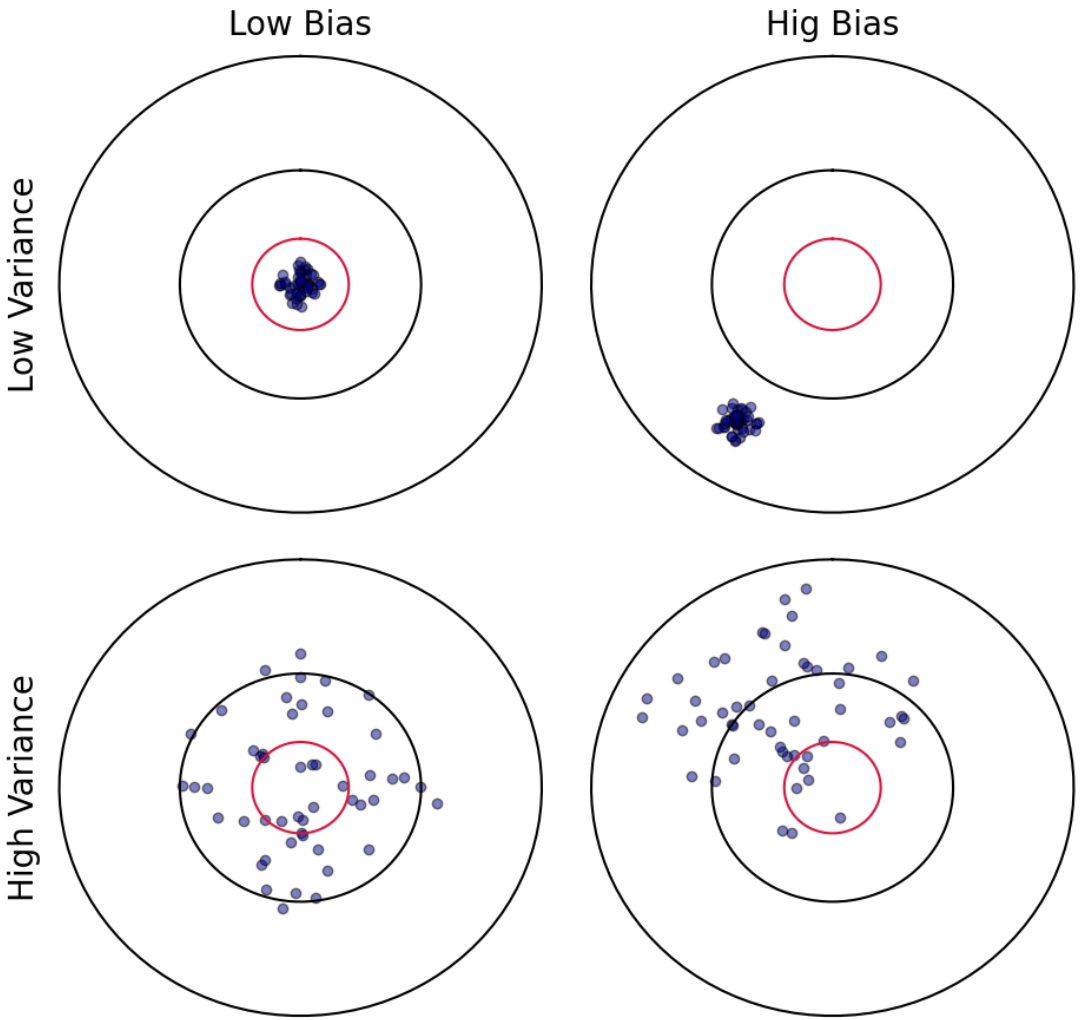
Bias (Zkreslení)

- ve statistice definována jako odchylka střední hodnoty odhadu parametru od očekávané hodnoty
- ve strojovém učení má význam schopnosti modelu predikovat hodnoty výstupu
- vysoký bias obvykle značí nedostatečnou komplexitu modelu
- s vysokým bias je spojen termín **underfitting**
- vysoký bias bychom si mohli představit i jako *"necitlivost parametrů modelu na změnu trénovacího datasetu"*

Variance (Rozptyl)

- ve statistice vyjadřuje variabilitu rozdělení souboru náhodných hodnot kolem její střední hodnoty
- ve strojovém učení má význam míry generalizace modelu
- vysoká variance obvykle značí příliš vysokou komplexitu modelu
- s vysokou mírou variance je spojen termín **overfitting**
- varianci bychom si mohli představit i jako *"přecitlivělost parametrů modelu na změnu trénovacího datasetu"*

Bias a variance



Variance - bias dekompozice

Model:

$$y = h(\boldsymbol{x}, \boldsymbol{\theta}) + \epsilon$$

- $h(\boldsymbol{x}, \boldsymbol{\theta}) = h$ - model
- y - výstupní hodnota (target) sledující nějakou funkční závislost na příznacích
- ϵ - chyba způsobena šumem ($\mathrm{E}[\epsilon] = 0$)

Bias:

$$\mathrm{bias}(h, y) = \mathrm{E}[h] - y$$

Definice rozptylu:

$$\mathrm{var}(h) = \mathrm{E}[h^2] - \mathrm{E}[h]^2$$

Kvadratická odchylka

$$\mathrm{S} = (y + \epsilon - h)^2$$

$$\mathrm{MSE} = \mathrm{E}[\mathrm{S}]$$

Variance - bias dekompozice

Dekompozice za předpokladu nezávislosti mezi h a ϵ:

1. $\mathrm{MSE} = \mathrm{E}[(y + \epsilon - h)^2]$
2. $\mathrm{MSE} = \mathrm{E}[y^2 + \epsilon^2 + h^2 + 2y\epsilon - 2h\epsilon - 2hy]$

všechny členy s ϵ v součinu $= 0$, rozepíšeme střední hodnoty

3. $\mathrm{MSE} = \mathrm{E}[h^2] + \mathrm{E}[y^2] - \mathrm{E}[2 \cdot h \cdot y] + \mathrm{E}[\epsilon^2]$

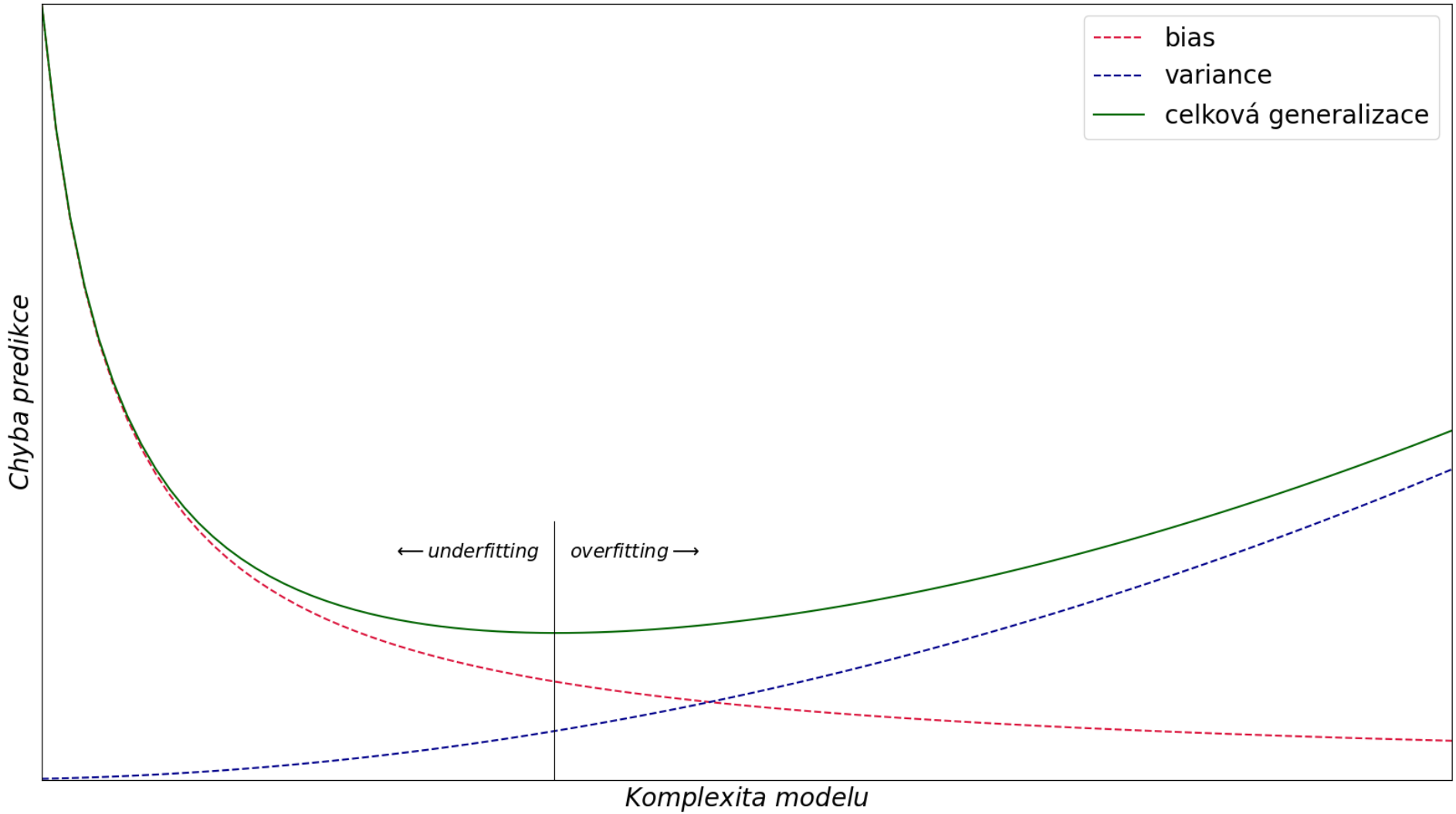
nahradíme $\mathrm{E}[x^2]$ z definice rozptylu, $\mathrm{var}(\epsilon) = \sigma^2$

4. $\mathrm{MSE} = \mathrm{var}(h) + \mathrm{E}[h]^2 + \mathrm{var}(y) + \mathrm{E}[y]^2 - \mathrm{E}[2 \cdot h \cdot y] + \mathrm{E}[\epsilon^2]$
5. $\mathrm{MSE} = \mathrm{var}(h) + \big(\mathrm{E}[h] - y\big)^2 + \sigma^2 + \mathrm{E}[\epsilon]^2$
6. $\mathrm{MSE} = \mathrm{variance} + \mathrm{bias}^2 + \sigma^2$

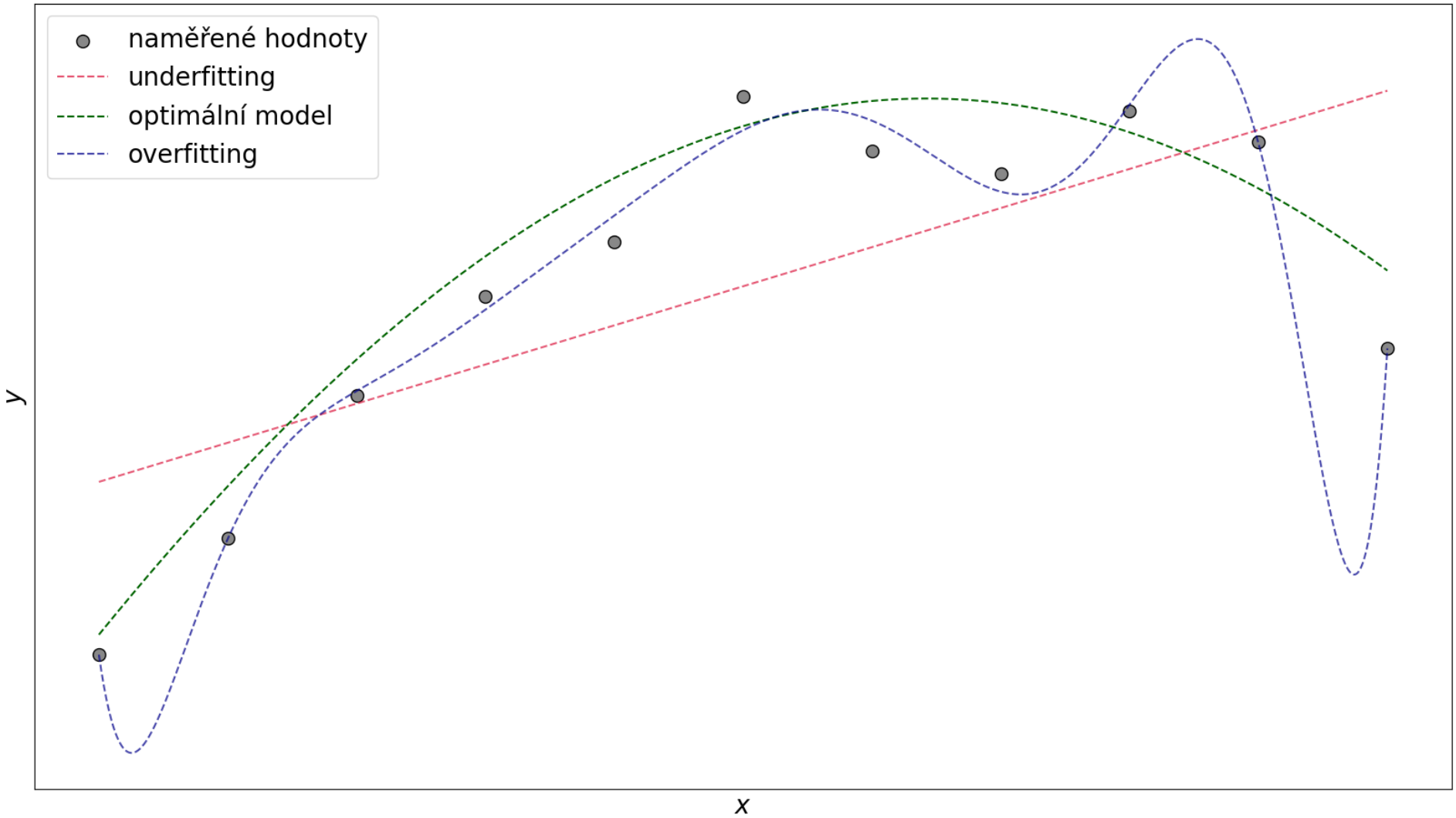
Jak bias a variance ovlivňují optimalizaci modelu?

- chyba predikce obsahuje tři složky - varianci, bias a chybu způsobenou šumem
- chybu predikce lze ovlivnit snížením bias či variance
- nízká míra bias obvykle vede k vysoké varianci a opačně $\rightarrow$ **bias-variance tradeoff**
- při optimalizaci hledáme *"extrém"*, kdy kombinace bias a variance vedou k nejnižší možné odchylce

Bias-variance tradeoff



Overfitting x underfitting



Rozdělování datasetu a výběr příznaků

Trénovací, testovací a validační datasety

- pro vyhodnocení kvality modelu je důležité použít jiný dataset než ten, na kterém je model natrénován
- výborné výsledky na trénovacím datasetu nevypovídají nic o generalizaci modelu
- v ideálním případě je vhodné mít k dispozici 3 datasety
  - **trénovací dataset** - používáme k nalezení parametrů modelu
  - **validační dataset** - průběžne používáme k vylepšení trénovaného modelu a optimalizaci hyperparametrů
  - **testovací dataset** - po dokončení optimalizace modelu a nalezení parametrů ověříme výsledky na datasetu, který model "neviděl"
- pokud má model hodně hyperparametrů, je obvykle potřeba větší validační dataset
- "rule of thumb"pro velikosti datasetů (trénovací/validační/testovací)
  1. 80-10-10
  2. 70-15-15
  3. 60-20-20

Jak rozdělit dataset

- **náhodný výběr** - vhodné pro vyvážené datasety, jinak hrozí, že výsledný model bude mít bias
- **stratifikovaný výběr** - počet zástupců tříd v každém datasetu (trénovací/validační/testovací) odpovídá celkovému poměru tříd v původním datasetu
  - problematické pokud jeden vzorek dat náleží do více tříd (např. na obrázku je několik objektů z různých tříd)
  - pro velký počet tříd a daty, která náležejí vícero třídám příliš komplikovaná metoda
  - pro třídy s malým počtem dat hrozí, že model sice bude s velkou přesností klasifikovat majoritní třídu, ale minoritní špatně (v binární klasifikaci nepoužitelný model)
- **vyvážený výběr** - vybereme náhodně zvolený počet zástupců z minoritní třídy a doplníme stejným počtem náhodně vybraných zástupců z majoritní třídy

Feature selection (výběr příznaků) - kategorizace metod

1. filtrační metody (filter methods)
2. wrapper metody (wrapper methods)
3. vnořené metody (embedded methods)

Feature selection

Feature selection - filtrační metody

Aplikace na dataset

- odstranění vstupních proměnných s konstantní nebo quasi-konstantní hodnotou (99% hodnot stejných)
- korelace mezi vstupními proměnnými a cílovou proměnnou - vysoká hodnota korelace = vhodná feature
- korelace mezi vstupními proměnnými - hledáme proměnné, které mezi sebou nejsou korelované
- výběr features podle vzájemné informace s cílovou proměnnou

$$I(X,Y) = \sum\sum P_{(X,Y)}(x,y) \log\left(\frac{P_{(X,Y)}(x,y)}{P_X(x)P_Y(y)}\right)$$

- mutual information v Pythonu

```
from sklearn.feature_selection import mutual_info_classif as MIC
mi_score = MIC(X,y)
```

- χ^2-test - vhodný pro kategorické proměnné (H_0 - proměnné jsou nezávislé)

Feature selection - wrapper methods

Algoritmy, které ubírají/přidávají feature na základě výsledků po trénovaní

- **dopředný výběr** - začneme s 1 featurou a přidáváme feature, které nejvíc zlepší výsledky modelu do té doby, dokud další přidaná featura zlepšuje model (výpočetně náročné) - přídáváme feature která má
  1. nejmenší p-hodnotu
  2. nejvíc zvýší R^2
  3. nejvíc sníží RSS
- **zpětná eliminace** - začneme s n featurama a ubíráme featury postupně features s největší p-hodnotou, obvykle dokud zbývají features s $p > 0.05$
- **vyčerpávající výběr** - vytvoříme všechny možné podmnožiny featur a vybereme tu, na které má model nejlepší výsledky (výpočetně náročné)^n
- **rekurzivní výběr** - stanovíme požadovaný počet featur a postupně ubíráme featury s nejmenší důležitostí (koeficienty u škálované lin. regrese, Gini nebo entropie u decision trees,...)

Feature selection - embedded methods

Metody, které jsou přímo součástí učícího algoritmu:

- regularizace (Lasso, Ridge, Elastic net) - použitelná pro klasifikační i regresní problémy
- rozhodovací stromy
- náhodný les
- XGBoost
- Extremely Randomized Trees Classifier

Vyhodnocení binárních klasifikátorů

Vyhodnocení přesnosti klasifikátorů - základní pojmy

- **P** - počet skutečně (empiricky ověřitelné) pozitivních vzorků v datasetu
- **N** - počet skutečně (empiricky ověřitelné) negativních vzorků v datasetu
- **TP** - výsledek klasifikátoru, který správně klasifikuje pozitivní vzorek
- **TN** - výsledek klasifikátoru, který správně klasifikuje negativní vzorek
- **FP** - pozitivní výsledek klasifikátoru pro negativní vzorek (chybně klasifikovaný negativní vzorek)
- **FN** - negativní výsledek klasifikátoru pro pozitivní vzorek (chybně klasifikovaný pozitivní vzorek)
- h^+ - pozitivní výsledek klasifikátoru
- h^- - negativní výsledek klasifikátoru
- g^+ - skutečně pozitivní vzorek
- g^- - skutečně negativní vzorek

Sensitivita, specifita

Senzitivita

- senzitivita vyjadřuje schopnost klasifikátoru úspěšně klasifikovat danou třídu
- je to pravděpodobnost, že klasifikace bude pozitivní pro vzorek dat který je skutečně pozitivní
- v AJ literatuře: sensitivity, recall, hit rate, true positive rate

$$TPR = P(h^+|g^+) = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR$$

Specificita

- senzitivita vyjadřuje schopnost klasifikátoru úspěšně zamítat danou třídu
- je to pravděpodobnost, že klasifikace bude negativní pro vzorek dat, který je skutečně negativní
- v AJ literatuře: specificity, selectivity, true negative rate

$$TNR = P(h^-|g^-) = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

Prediktivní hodnoty

Pozitivní prediktivní hodnota

- pozitivní prediktivní hodnota vyjadřuje pravděpodobnost, že vzorek dat pozitivně klasifikován pro skutečně pozitivní vzorek
- v AJ literatuře: precision, positive predictive value

$$PPV = P(g^+|h^+) = \frac{TP}{TP + FP} = 1 - FDR$$

Negativní prediktivní hodnota

- negativní prediktivní hodnota vyjadřuje pravděpodobnost, vzorek dat je negativně klasifikován pro skutečně negativní vzorek
- v AJ literatuře: negative predictive value

$$NPV = P(g^-|h^-) = \frac{TN}{TN + FN} = 1 - FOR$$

Falešně pozitivní míra, falešně negativní míra

Falešně pozitivní míra

- falešně pozitivní míra vyjadřuje pravděpodobnost falešně pozitivního výsledku klasifikace pro negativní vzorek
- v AJ literatuře: fall-out, false positive rate

$$FPR = P(h^+|g^-) = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - TNR$$

Falešně negativní míra

- falešně negativní míra vyjadřuje pravděpodobnost falešně negativního výsledku klasifikace pro positívní vzorek
- v AJ literatuře: miss rate, false negative rate

$$FPR = P(h^-|g^+) = \frac{FN}{P} = \frac{FN}{FN + TP} = 1 - TPR$$

Přesnost

- přesnost vyjadřuje pravděpodobnost správné klasifikace
- vhodná metrika pouze pro vyvážené datasety
- v AJ literatuře: accuracy

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

- pro klasifikaci do více tříd je to poměr správných klasifikací k celkovému počtu klasifikací

$$ACC_m = \frac{\text{správná klasifikace}}{\text{celkový počet klasifikací}}$$

F_1-skóre

- metrika kombinující positivní prediktivní hodnotu se sensitivitou

$$F_1 = \frac{2}{TPR^{-1} + PPV^{-1}} = 2 \cdot \frac{TPR \cdot PPV}{TPR + PPV} = \frac{2TP}{2TP + FP + FN}$$

- harmonický průměr se blíží aritmetickému průměru pro hodnoty které jsou si blízké
- F_1 skóre ignoruje TN!
- pro silně nevyvážené datasety je jeho interpretace zavádějící
- problém může způsobit apriorní stejná váha PPV i TPR

Matice záměn (confusion matrix)

Tabulka, ve které:

- **sloupec** - skutečná třída klasifikace
- **řádek** - výstupní třída klasifikátoru
- **hodnota** - četnost kombinací skutečné třídy a výstupní třídy klasifikátoru

		Skutečná třída				
	Třídy	A	B	C	D	TPR
Výstup klasifikátoru	A	10	1	1	3	$\frac{10}{13}$
	B	2	15	0	1	$\frac{15}{22}$
	C	1	5	16	4	$\frac{16}{20}$
	D	0	1	3	11	$\frac{11}{19}$
	PPV	$\frac{10}{15}$	$\frac{15}{18}$	$\frac{16}{26}$	$\frac{11}{14}$	

Vliv prevalence na výsledky klasifikace

- prevalence je podíl pozitivních dat v celkovém datasetu

$$P_r = \frac{P}{P + N}$$

- prevalence má zásadní vliv na výsledky klasifikátorů čím víc se její hodnoty blíží 0 nebo 1
- rozumné datasety by měli být vyvážené, tzn. stejný počet N a P dat

Vliv prevalence na výsledky klasifikace - příklad

Příklad

Klasifikátor má senzitivitu 95% a specificitu 98%. V datasetu je zastoupeno 2% pozitivních vzorků a 98% negativních. Jedná se o nevyvážený dataset. Pro preditivní hodnoty:

$$PPV = P(g^+|h^+) = \frac{P(h^+|g^+)P(g^+)}{P(h^+|g^+)P(g^+) + P(h^+|g^-)P(g^-)}$$

$$PPV = \frac{0.95 \cdot 0.02}{0.95 \cdot 0.02 + (1 - 0.98) \cdot 0.98} = 0.4922$$

$$NPV = P(g^-|h^-) = \frac{P(h^-|g^-)P(g^-)}{P(h^-|g^-)P(g^-) + P(h^-|g^+)P(g^+)}$$

$$NPV = \frac{0.98 \cdot 0.98}{0.98 \cdot 0.98 + (1 - 0.95) \cdot 0.02} = 0.9990$$

Pro dataset s 2% prevalancí je pro pozitivně detekovaný vzorek pravděpodobnost pouze 49.22% že je skutečně pozitivní!

Vliv prevalence na výsledky klasifikace - příklad

Příklad

Klasifikátor má senzitivitu 95% a specificitu 98%. V datasetu je zastoupeno 50% pozitivních vzorků a 50% negativních. Jedná se o vyvážený dataset. Pro preditivní hodnoty:

$$PPV = P(g^+|h^+) = \frac{P(h^+|g^+)P(g^+)}{P(h^+|g^+)P(g^+) + P(h^+|g^-)P(g^-)}$$

$$PPV = \frac{0.95 \cdot 0.5}{0.95 \cdot 0.5 + (1 - 0.98) \cdot 0.5} = 0.9794$$

$$NPV = P(g^-|h^-) = \frac{P(h^-|g^-)P(g^-)}{P(h^-|g^-)P(g^-) + P(h^-|g^+)P(g^+)}$$

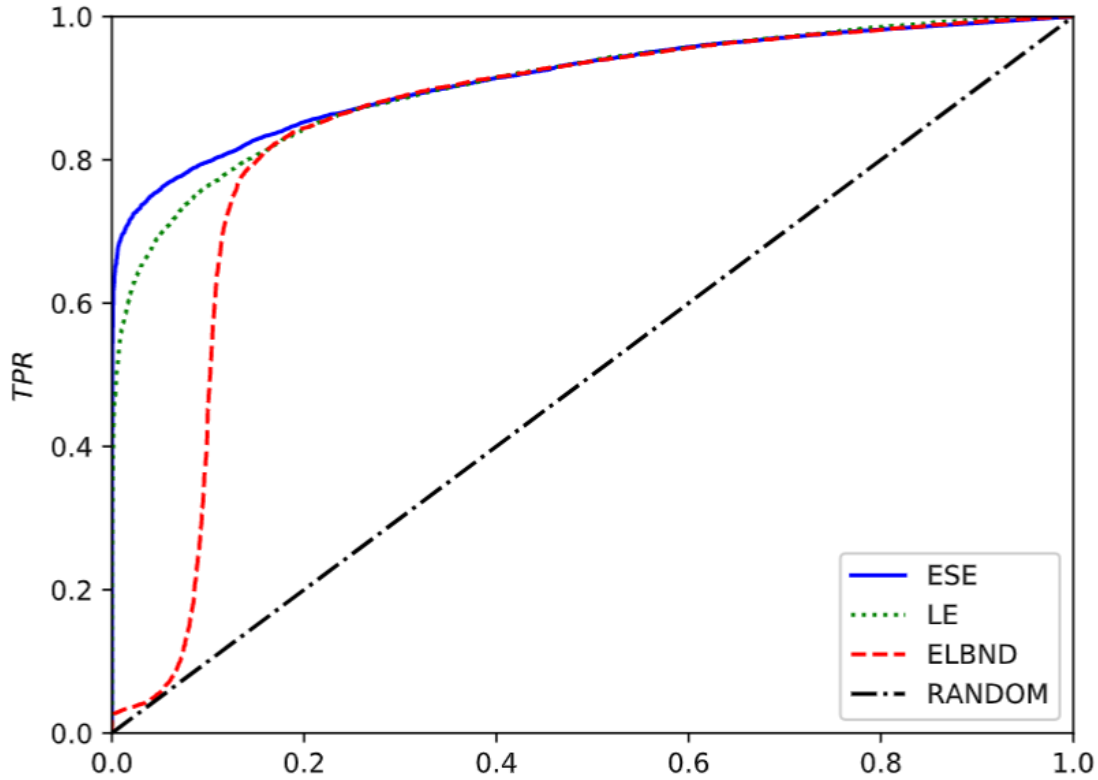
$$NPV = \frac{0.98 \cdot 0.5}{0.98 \cdot 0.5 + (1 - 0.95) \cdot 0.5} = 0.9515$$

Vhodné metriky pro nevyvážené datasety

- pozitivní prediktivní hodnota - penalizuje falešně pozitivní detekce, klasifikátor, který sice detekuje všechny P vzorky pozitivně, ale všechny negativní vzorky také pozitivně je nepoužitelný
- senzitivita - v kritických oblastech (např. medicína) potřebujeme, by klasifikátor detekoval kritickou třídu správně co nejčastěji a s co nejmenším počtem falešne negativních detekcí
- F_1 skóre - harmonický průměr pozitivní prediktivní hodnoty a senzitivity $\implies$ nezahrnuje TN detekce

ROC křivka

- ROC křivka (receiver operating characterestic curve) vypovídá o schopnosti binárního klasifikátoru správně klasifikovat vzorky v závislosti na velikosti rozhodovacího prahu
- osa x reprezentuje hodnotu *FPR*
- osa y reprezentuje hodnotu *TPR*



AUC - plocha pod křivkou ROC

- plocha pod křivkou ROC (AUC - area under ROC Curve) odpovídá podílu správných klasifikací k celkovému počtu klasifikací
- ideální klasifikátor má hodnotu $AUC = 1$
- náhodný klasifikátor má hodnotu $AUC = 0.5$
- nejhorší klasifikátor (vše klasifikuje obráceně) má hodnotu $AUC = 0$

Validace modelů

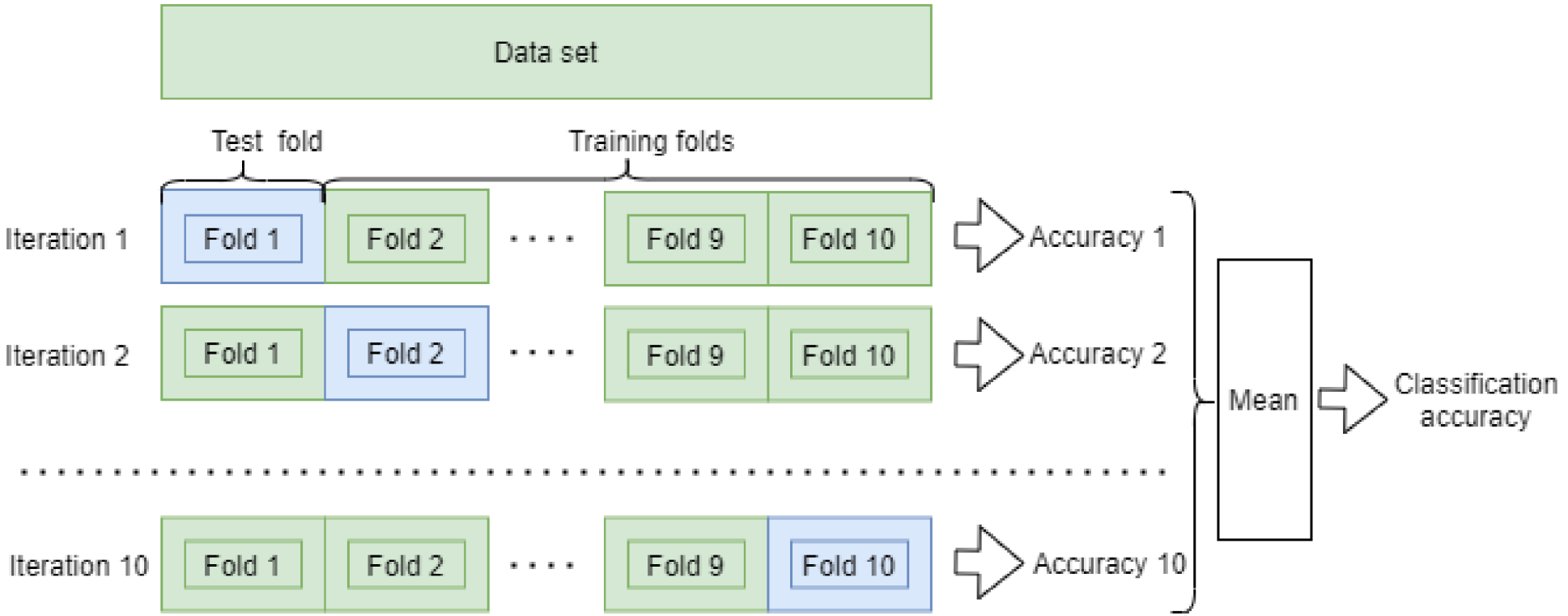
Metoda opakované zádrže (repeated holdout)

1. vygenerování trénovacího a testovacího datasetu
2. nalezení parametrů modelu
3. výpočet chybového kritéria
4. návrat do kroku 1., opakovat k-krát
5. výpočet celkové chyby jako

$$E = \frac{E_1 + E_2 + \ldots E_k}{k}$$

- nevýhodou je, že některé vzorky z datasetu se nemusí nikdy objevit v trénovací množině dat!
- vhodné pro datasety s velkým množstvím dat pro získání prvotních modelů, které budou dále optimalizaovány
- v porovnání s křižovou validací výrazně méně výpočetně náročné

K- fold křížová validace



K-fold křížová validace

1. náhodně rozdělíme dataset na k-podmnožin stejné velikosti
2. postupně vybíráme jednu podmnožinu jako testovací a zbylých $k-1$ podmnožin jako trénovací
3. pro každý i-tý výběr vypočítáme parametry modelu a chybové kritérium E_i
4. vypočítáme výsledné chybové kritérium jako průměrnou chybu

$$E = \frac{E_1 + E_2 + \ldots E_k}{k}$$

- v trénovacích datasetech budou postupně zastoupeny všechna data
- v testovacích datasetech budou postupně zastoupeny všechna data
- výpočetně náročná metoda
- rule of thumb pro volbu K: $K = 10$

Leave-one-out křížová validace

1. rozdělíme dataset na m podmnožin obsahujících právě jedno datum
2. postupně vybíráme jednu podmnožinu jako testovací a zbylých $m-1$ podmnožin jako trénovací
3. pro každý i-tý výběr vypočítáme parametry modelu a chybové kritérium E_i
4. vypočítáme výsledné chybové kritérium jako průměrnou chybu

$$E = \frac{E_1 + E_2 + \ldots E_k}{m}$$

- extrémně výpočetně náročná metoda
- vhodné pro datasety s malým počtem dat
- poskytuje velmi dobrý odhad kvality modelu
- nelze zajistit stratifikovaná trénovací a testovací data!
- je to k-fold křížová validace, s velikostí foldu 1

Feature extraction

Přehled feature extraction algoritmů

- feature extraction je technika, která původní dataset transformuje na dataset jiných featur se stejnou nebo nižší dimenzionalitou (interpretovatelnost features nemusí být zachována!)
- některé vybrané algoritmy:
  1. PCA (principal component analysis)
  2. LDA (linear discriminant analysis)
  3. ICA (independent component analysis)
  4. LLE (localy linear embedding)
  5. AE (autoencoders)
  6. t-SNE (t-distributed Stochastic Neighbor Embedding)

Linear Discriminant Analysis

- chceme minimalizovat varianci v rámci jednotlivých tříd a maximalizovat rozdíl jejich středních hodnot $\Longrightarrow$ lepší separovatelnost dat
- hledáme zobrazení

$$y_i = \boldsymbol{w}^T \boldsymbol{x}_i$$

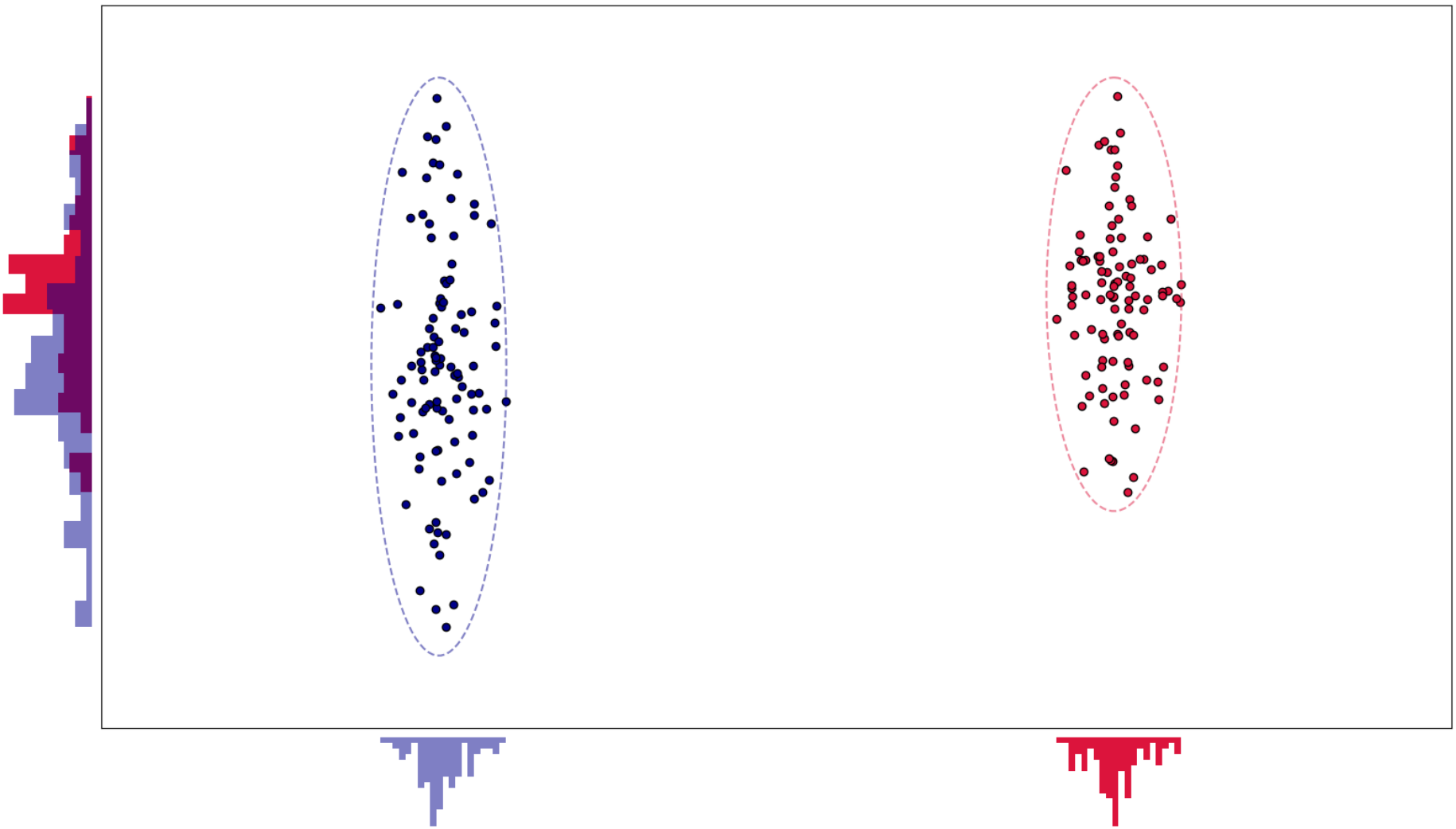
- střední hodnota v rámci j-té třídy

$$\mu_j = \frac{1}{m_j} \sum_{y_i \in c_j}^{m_j} y_i, j = 1, 2$$

- scatter

$$s_j^2 = \sum_{y_i \in c_j}^{m_j} (y_i - \mu_j)^2, j = 1, 2$$

Linear Discriminant Analysis



Linear Discriminant Analysis - optimlizační probém

- řešíme maximalizační úlohu

$$\max_{\mathbf{w}} J(\mathbf{w}) = \max_{\mathbf{w}} \frac{|\mu_1 - \mu_2|}{s_1^2 + s_2^2}$$

- pro kterou existuje řešení

$$\mathbf{w} = \mathbf{S}_W^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$\hat{\boldsymbol{\mu}}_j = \frac{1}{m_j} \sum_{\mathbf{x}_i \in c_j}^{m_j} \mathbf{x}_i, j = 1, 2$$

- $\mathbf{S}_W$ je tzv. within-class scatter matrix

$$\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

$$\mathbf{S}_j = \sum_{\mathbf{x}_i \in c_j}^{m_j} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j)^T, j = 1, 2$$

Linear Discriminant Analysis - shrnutí

1. spočítáme průměry v rámci třídy
2. spočítáme scatter matice
3. určíme $\mathbf{S}_W$ a její inverzi
4. vypočítáme vektor zobrazení $\mathbf{w}$
5. vypočítáme nově features $y_i = \mathbf{w}^T \mathbf{x}_i$

- LDA v Pythonu

`sklearn.discriminant_analysis.LinearDiscriminantAnalysis`