Progresivní technologie v informatice I

Umělá inteligence a strojové učení 2

Fakulta informačních technologií

České vysoké učení technické v Praze

2023/2024

Financováno Evropskou unií
NextGenerationEU

Národní plán obnovy

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Hlavní body

1. Validační data: ladění hyperparametrů

2. Základy lineární regrese

Ladění hyperparametrů (1/3)

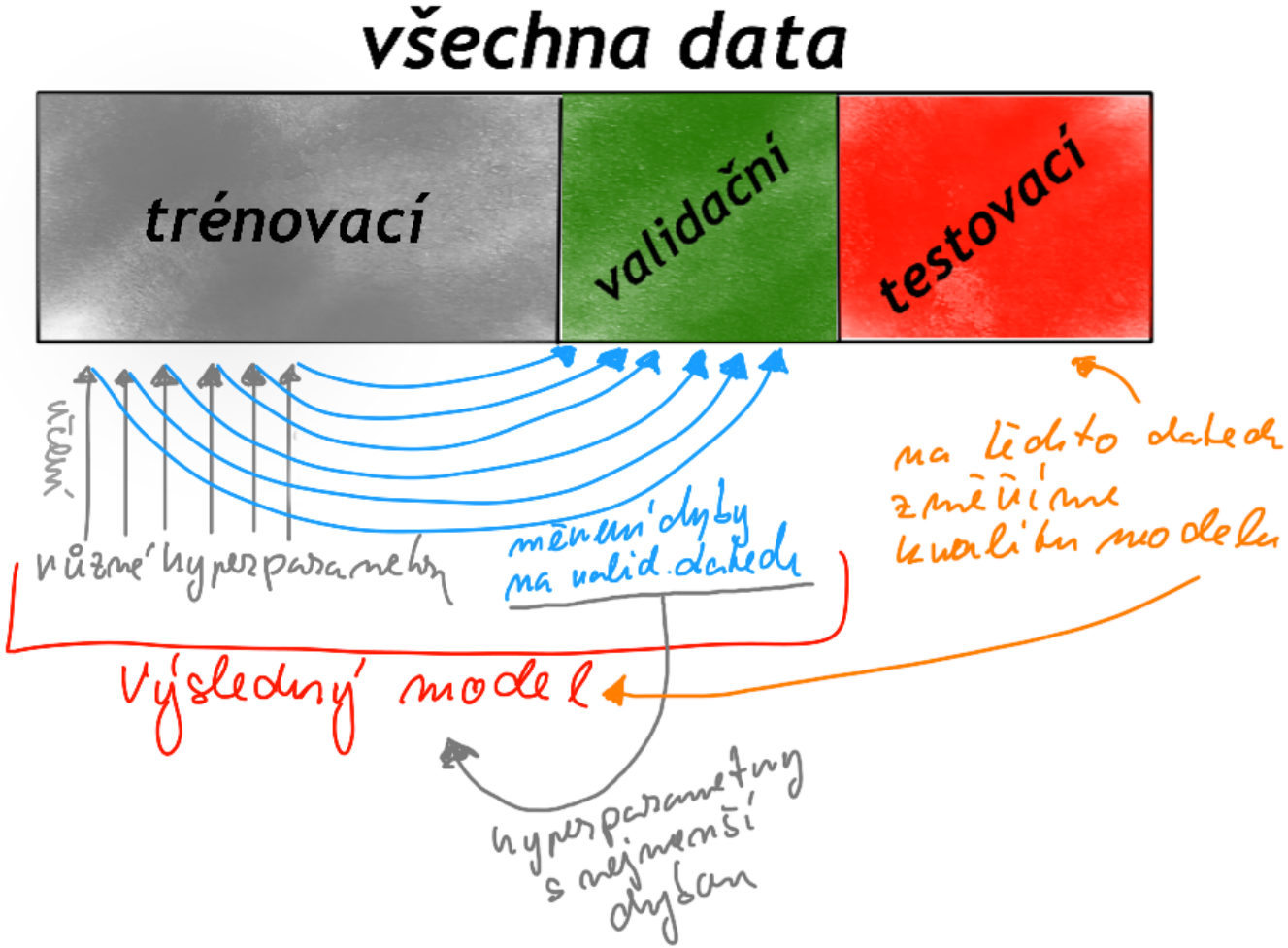
Teď se dostáváme ke třetí otázce: **Jak poznám, jakou mám zvolit hloubku stromu příp. jiné jeho parametry?**

- Můžeme postupovat takto:
  1. Data si rozdělíme na trénovací a testovací.
  2. Pro různé hodnoty hloubky stromu (parametr `max_depth`) naučíme rozhodovací strom na trénovacích datech
  3. a pro každou hloubku stromu změříme přesnost (classification accuracy) na testovacích datech.
  4. Vyberme tu hloubku, která dává nejmenší testovací chybu.
  5. Tuto hodnotu vezmeme také jako odhad chyby na reálných datech, a tedy jako objektivní míru kvality modelu.
- Je někde problém? Ano, je!
- Výsledná testovací chyba bude zpravidla příliš optimistickým odhadem skutečnosti! Výsledný model (konkrétně parametr hloubka stromu) byl vybrán na základě těchto dat a model je tedy těmto datům přizpůsobený.
- Takto bychom porušili zásadu, že **při učení modelu nesaháme na testovací data**, pokud má být testovací chyba rozumným odhadem skutečné chyby modelu.

Ladění hyperparametrů (2/3)

- Jak se s tímto problémem vypořádat?
- Obvykle se to dělá tak, že se data rozdělí ne na dvě, ale na tři podmnožiny: trénovací, **validační** (angl. **validation**) a testovací:
  1. Pro různé hodnoty hloubky stromu `max_depth` naučíme rozhodovací strom
  2. a změříme jeho chybu (přesnost) na validačních datech.
  3. Jako optimální hodnotu vybereme tu s nejmenší chybou (tj. s nejvyšší přesností).
  4. Chyba na testovacích datech, které dosud ležely ladem, je pak rozumným odhadem chyby modelu.
- Parametrům, jako je `max_depth`, které určují *tvar* nebo *komplexitu* modelu, se říká **hyperparametry modelu**.
- Určování těchto parametrů pomocí validačních dat se říká **ladění** (angl. **tuning**).
- Tento parametr nemusí být jeden, ale může jich být více. Pokud jsou spojité, může být výpočetně složité jich otestovat *reprezentativní* množství a je třeba dělat kompromisy.

Ladění hyperparametrů (3/3)



Ladění hyperparametrů: poznámky (1/2)

- Pro představu, jaké hyperparametry jsou dostupné pro rozhodovací stromy v knihovně `sklearn`:

```
Init signature: DecisionTreeClassifier(criterion='gini', splitter='best', max_de
pth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0,
 max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decreas
e=0.0, min_impurity_split=None, class_weight=None, presort=False)
```

- Význam těchto parametrů si vysvětlíte (a ukážete) na cvičení.
- Dělení dat na trénovací/validační/testovací podmnožiny je dobré dělat náhodně. Tj. nevybrat první půlku dat jako trénovací, další čtvrtinu jako validační a zbytek vzít jako testovací.
- V pořadí dat by mohla být nějaká zákonitost, která by znamenala, že trénovací a testovací data nebudou reprezentativní.
- Proto se postupuje tak, že se data vybírají náhodně. V `sklearn` je na to metoda:

```
from sklearn.model_selection import train_test_split
Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size=0.25, random_state=33)
```

Ladění hyperparametrů: poznámky (2/2)

- Neexistuje žádný optimální poměr velikosti trénovací/validační/testovací množiny dat.
- Obvyklé poměry jsou takové, že 20 % dat vezmeme jako testovací množinu a ze zbytku vezmeme 20 % jako validační množinu. Co zbude, jsou trénovací data.
- Místo 20 % lze použít 25 %, nebo 30 %.
- Lze použít i sofistikovanější strategie, např. měnit velikost validační množiny a koukat se, co to dělá.
- Často používanou metodou je také **cross-validace**, o které budeme mluvit později.

Hlavní body

1. Validační data: ladění hyperparametrů
2. Základy lineární regrese

Motivační příklad

- Chceme prodat nemovitost (řekněme v Praze) a netušíme, za kolik ji máme potenciálním zájemcům nabídnout.
- Zároveň si nechceme platit žádného „realitního“ odborníka, který by nám se stanovením ceny poradil.
- Zkusíme tedy udělat vlastní průzkum realitního trhu a vytvořit model, který nám pomůže cenu stanovit.
- Napíšeme skript, který stáhne data z realitních serverů a uloží je v nějaké strukturované podobě (ideálně tabulce či více tabulkách).
- Pro jednoduchost uvažujme, že budeme znát u každé nabídky toto:
  - Y – cenu, za kterou se prodává,
  - X_1 – užitnou plochu,
  - X_2 – počet místností,
  - X_3 – vzdálenost od nejbližší zastávky metra.
- Jako vysvětlovanou proměnnou Y jsme označili veličinu, kterou pro naši nemovitost neznáme a chceme ji predikovat.
- Příznaky $X_1, \ldots X_3$ označují veličiny, které pro naši nemovitost známe a o kterých věříme, že cenu Y ovlivňují.

Formalizace úlohy

Obecně tedy chceme na základě p příznaků $X_1, \ldots, X_p$ predikovat hodnotu vysvětlované proměnné Y.

V modelu lineární regrese předpokládáme lineární závislost vysvětlované proměnné na hodnotách příznaků.

Jelikož nedoufáme, že tato závislost je perfektní v tom smyslu, že pro stejné hodnoty $x_1, \ldots, x_p$ příznaků $X_1, \ldots, X_p$ dostaneme vždy stejnou hodnotu vysvětlované proměnné Y, modelujeme tuto závislost následovně:

$$Y = w_1 x_1 + \ldots + w_p x_p + \varepsilon,$$

kde $w_1, \ldots, w_p$ jsou nějaké neznámé koeficienty a ε je náhodná veličina.

Poznámky:

- Veličina ε odpovídá části Y, která je nevysvětlitelná pomocí hodnot příznaků a je tedy z našeho pohledu náhodná.
- Do náhodné veličiny ε se tak „schovají“ vlivy, které neznáme nebo cíleně nezahrnujeme do našeho modelu (např. stáří budovy, počet koupelen, počet oken) ale např. i chyby, nekonzistence dat a jiné podivnosti v měření příznaků.

Obvykle ještě oddělujeme střední hodnotu náhodných vlivů a dostáváme tak:

Model lineární regrese

Hodnota vysvětlované proměnné Y v bodě $(x_1, \ldots, x_p)^T$ je

$$Y = w_0 + w_1 x_1 + \ldots + w_p x_p + \varepsilon,$$

kde $\mathbb{E}\varepsilon = 0$.

- Koeficient w_0 se nazývá *intercept* a odpovídá (očekávané) výchozí hodnotě Y při nulových příznacích.
- Zavedeme-li nový konstantní příznak $X_0 = x_0 = 1$ a vektorové značení

  $$\vec{x} = (x_0, x_1, \ldots, x_p)^T \quad \text{a} \quad \vec{w} = (w_0, w_1, \ldots, w_p)^T,$$

  můžeme zkráceně psát

  $$Y = \vec{w}^T \vec{x} + \varepsilon.$$

- Vektor $\vec{w} = (w_0, w_1, \ldots, w_p)^T$ koeficientů také někdy nazýváme vektor vah.

Predikce v modelu lineární regrese

Předpokládejme nyní, že už máme odhad $\hat{\vec{w}}$ vektoru koeficientů $\vec{w}$.

Hodnotu Y v konkrétním bodě $\vec{x}$ predikujeme vztahem

$$\hat{Y} = \hat{\vec{w}}^T \vec{x} = \hat{w}_0 + \hat{w}_1 x_1 + \dots \hat{w}_p x_p.$$

Skutečná hodnota Y v bodě $\vec{x}$ je přitom určena vztahem

$$Y = \vec{w}^T \vec{x} + \varepsilon$$

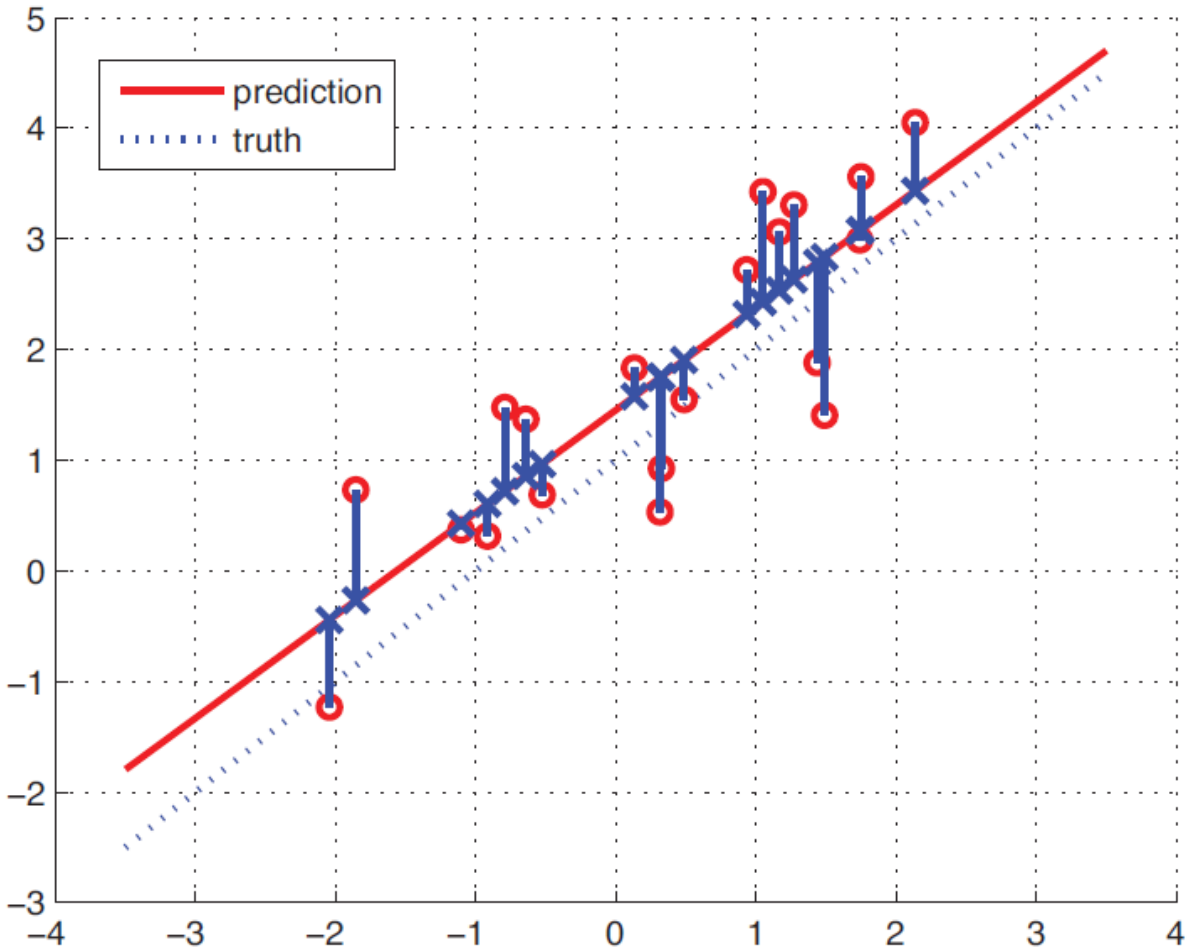
a je tedy náhodnou veličinou.

Z předpokladu $\mathbb{E}\varepsilon = 0$ plyne, že

$$\mathbb{E}Y = \vec{w}^T \vec{x}$$

a $\hat{Y}$ je tedy vlastně bodovým odhadem střední hodnoty $\mathbb{E}Y$ v bodě $\vec{x}$.

Vizualizace modelu lineární regrese



Modré body jsou body trénovací množiny. Červené body jsou predikce. Modré křížky odpovídají středním hodnotám bodů trénovací množiny, $(\vec{x}_i, \mathbb{E}\, Y_i)$. Modrá čerchovaná čára je skutečná regresní přímka daná rovnicí $y = \vec{w}^T \vec{x}$ a červená čára je přímka $\hat{y} = \hat{\vec{w}}^T \vec{x}$ určující naše predikce.

Měření chybovosti predikce pomocí ztrátové funkce

Zaměřme se nyní na problematiku odhadu vektoru parametrů modelu $\vec{w}$.

Následující úvahy jsou obecně platné pro supervizované učení nějakého modelu s parametry.

- Naším cílem je najít takovou hodnotu $\vec{w}$, aby chyba modelu byla co nejmenší.
- Tuto hodnotu pak použijeme jako odhad $\hat{\vec{w}}$.
- K tomu musíme specifikovat, co se myslí chybou modelu a v jakém smyslu má být nejmenší.
- Chybu modelu nejčastěji měříme pomocí nějaké nezáporné funkce $L : \mathbb{R}^2 \to \mathbb{R}$, nazývané *ztrátová funkce* (angl. *loss function*), kterou aplikujeme na skutečnou hodnotou proměnné Y a odpovídající predikci $\hat{Y}$.
- Obvyklou volbou v případě spojité vysvětlované veličiny bývá *kvadratická ztrátová funkce*,

$$L(Y, \hat{Y}) = (Y - \hat{Y})^2.$$

Metoda nejmenších čtverců

- Velikost chyby modelu v bodě $\vec{x}$ je tedy $L(Y, \hat{Y})$, kde Y je skutečná hodnota vysvětlované Y v bodě $\vec{x}$ a $\hat{Y} = \vec{w}^T \vec{x}$ je predikce v bodě $\vec{x}$.
- Otázkou je, v jakém bodě $\vec{x}$ bychom měli hodnotu $L(Y, \hat{Y})$ vyhodnocovat a následně minimalizovat vzhledem k $\vec{w}$.
- Zřejmě to musí být bod $\vec{x}$ z trénovací množiny, protože jinak bychom neznali skutečnou hodnotu Y v tomto bodě.
- Abychom co nejvíc využili trénovací data, budeme minimalizovat součet chyb přes všechny body trénovací množiny, tj. přes všechny dvojice $(\vec{x}_i, Y_i)$ pro $i = 1, \ldots, N$.
- Součet chyb přes všechny tyto body pro kvadratickou ztrátovou funkci je

$$\mathrm{RSS}(\vec{w}) = \sum_{i=1}^{N} L(Y_i, \vec{w}^T \vec{x}_i) = \sum_{i=1}^{N} (Y_i - \vec{w}^T \vec{x}_i)^2$$

  a nazýváme ho *reziduální součet čtverců* (angl. *residual sum of squares*).
- Minimalizací tohoto výrazu získáme odhad $\hat{\vec{w}}$. Tento postup se nazývá *metoda nejmenších čtverců* (angl. the method of *least squares*).