

# Vytěžování dat, cvičení 10: Rozhodovací stromy

Miroslav Čepek, Michael Anděl



Evropský sociální fond  
Praha & EU: Investujeme do vaší budoucnosti

*Fakulta elektrotechnická, ČVUT*

- ▶ Cílem této úlohy je zjistit, jak se projevuje prořezávání stromu na úspěšnosti klasifikace na testovací množině.
- ▶ Rozhodovací stromy už znáte z přednášky.

- ▶ Dataset je uměle generovaný na začátku souboru cv10.m, který si stáhnete ze stránek cvičení. Úkolem bude tento soubor doplnit.
- ▶ Dataset se skládá z "naměřených" hodnot (proměnná obs) a z tříd do kterých se snažíme klasifikovat (proměnná class).
- ▶ Třídy jsou 2: "green" a "blue".

- ▶ V m-file se nejprve vygeneruje soubor dat. Poté se v cyklu opakovaně použít náhodné rozřazení na trénovací a testovací množinu. Vaším úkolem je uvnitř cyklu...
- ▶ Vytvořit rozhodovací strom z trénovací množiny
- ▶ Zjistit chybu na testovací množině pro různé úrovně prořezání.
- ▶ Zarovnat chyby po prořezání podle hloubky zbylého stromu
- ▶ Tabulku chyb vykreslit pomocí grafu boxplot

- ▶ K vytvoření stromu použijte funkci `classregtree`.
- ▶ **DULEŽITÉ:** nastavte parametry "prune" na "off" a "splitmin" (resp. "minparent" v případě verzí Matlabu > 2014) na 2
- ▶ K vytvoření stromu použijte trénovací množinu

## Zjištění chyby (misclassification cost)

- ▶ Ke zjištění chyby použijte funkci `test` (je jich v matlabu více, tahle je metoda u třídy `@classregtree`)
- ▶ Pusťte ji na testovací množině.
- ▶ Funkce vrátí vektor chyb pro různé prořezání stromu. Na prvním místě je chyba pro strom bez prořezání, na posledním místě chyba pro strom s hloubkou 1.

- ▶ Chyby srovnejte do tabulky costs podle hloubky stromu po prořezání, ne podle úrovně prořezání
- ▶ Příklad: během dvou cyklů naleznete 2 různě hluboké stromy, hloubky 4 a hloubky 6. Metoda test vám vrátí vektor s délkou 4 resp. 6. Vaším cílem je, aby vždy poslední indexy byly v tabulce pod sebou. Poslední indexy totiž reprezentují chybu pro strom hloubky 1, předposlední 2 atd..

- ▶ Pro učení stromu fci `fitctree`
- ▶ Pro vyčíslení chyby pro jednotlivé úrovně prořezání je třeba použít nejprve fci `prune(treeObject, 'level', l)`, kde `l` je úroveň prořezání v intervalu `0 : max(treeObject.PruneList)`ř
- ▶ Pro každý prořezaný strom je pak třeba zavolat fci `predict` a následně porovnat predikci se skutečnými třídami.



Váš protokol by měl obsahovat:

- ▶ Výsledky v podobě boxplotu daných úrovní rozhodovacích stromů
- ▶ Závěry pozorovatelné z výsledků