Rozhodovací stromy a lesy

Lenka Matoušová, Radka Uhlířová, Jakub Opršal, Pavel Palát, Daniela Šollerová

Gym. Aloise Jiráska Litomyšl, Gym. Dr. K. Polesného Znojmo,

Gym. tř. Kpt. Jaroše Brno, SPŠE Kounicova Brno, Gym. Mozartova Pardubice

lenka9@centrum.cz, sagi-ta@email.cz, snek@centrum.cz, harry_x@babylon5.cz

Abstrakt

Rozhodovací stromy a lesy jsou jedněmi ze základních metod strojového učení dneška. Nalézají své použití např. při řešení klasifikačních problémů či AI (umělá inteligence).

1 Přehled algoritmů a technik

V současné technické/vědecké praxi neustále narůstá potřeba algoritmů strojového učení, popř. AI. Tyto metody řeší rozličné kategorie problémů, nejčastějším řešeným problémem je klasifikace určitých objektů do daných tříd, kterým se budeme zabývat. Velký význam má též použití těchto algoritmů v AI, kde se často využívá expertních systémů (většinou reprezentovaných pomocí rozhodovacích stromů) a/nebo neuronových sítí používaných na speciální problémy AI, kde běžné metody nejsou efektivní (typickým příkladem je např. algoritmus řízení auta). Další možnou aplikací jsou aproximace funkcí, kde jsou nejefektivnější neuronové sítě.

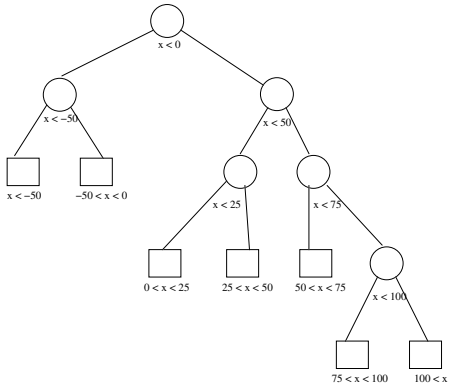
Obecně rozlišujeme dva druhy strojového učení:

- Učení s učitelem
- Učení bez učitele

Odlišnost spočívá především v tom, že učící množina je přítomna jen v případě učení s učitelem.
Při klasifikování objektu je nutné objekt popsat několika proměnnými (parametry). Tyto vstupní proměnné můžeme rozdělit do několika kategorii, např.

- Numerické
- Kategoriální (např. barva - černá, hnědá,... - jedná se o konečnou množinu)

V problémech klasifikace do tříd je často používaný též algoritmus zvaný *Nearest neighbour*, který spočívá v tom, že si měří vzdálenosti vstupního bodu k prvkům trénovací množiny v N (kde N je počet vstupních proměnných) rozměrném prostoru a v případě, že dostane nová vstupní data, snaží se nalézt jeho nejbližší možný bod. K tomu může používat euklidovskou vzdálenost nebo jinou metriku. Existuje několik modifikací tohoto algoritmu -



Obrázek 1: Rozhodovací strom

např. *K-NN* - kde se provádí hledání K nejbližších sousedů daného bodu. Takovýto algoritmus je přesnější, jelikož je méně náchylný k šumu. Algoritmus může též provádět vlastní učení, kdy při úspěšné klasifikaci objektu provede jeho přidání do trénovací množiny, ale nejedná se o příliš používanou techniku. NN má problém v případě narůstání počtu dimenzí (tzv. prokletí dimenzionality).
Další používanou technikou je LDA (Linear Discriminant Analysis) - která je vhodná v případě, že data jednotlivých tříd mají Gaussovské rozdělení. Metoda vyžaduje lineární separabilnost vstupních dat.
Dále se pak používá též neuronových sítí, které poskytují velmi velké možnosti, nicméně problémem je složitost vytrénování dané sítě.
Poslední metodou, kterou se budeme zabývat, jsou právě rozhodovací stromy. Mezi jejich výhodu patří především menší složitost učení (než je tomu např. u neuronových sítí), jednoduchá reprezentace a možnost sestavování pravidel z naučených dat, jelikož tyto rozhodovací stromy nejsou nic jiného než souborem hierarchicky řazených podmínek a fakt, že metoda nevyžaduje lin. separabilnost dat.

2 Rozhodovací stromy

2.1 Struktura rozhodovacího stromu

Rozhodovací strom se skládá z kořenového uzlu stromu (nejvyšší uzel na stromu). Dále pak každý uzel může obsahovat dvě nebo více větví vedoucí na další takovýto uzel, popř. na *leaf node*, který neobsahuje žádnou další větev.
Nejtypičtějším rozhodovacím stromem je binární strom, jež má u každého svého rozhodovacího uzlu danou podmínku závisející na jedné vstupní proměnné (je možno samozřejmě mít u uzlu i složitou funkci, ale tento postup se v praxi neosvědčil) a podle jejího výsledku (0,1) se vybere následující uzel, který se vyhodnotí identicky. Tak algoritmus pokračuje až do doby, než narazí na tzv. *leaf node*, který je zároveň jeho výsledkem.
Skutečným problémem je ovšem vytvoření binárního stromu a stanovení jednotlivých podmínek u jeho uzlů. K tomuto je používáno několik algoritmů, jimiž se budeme zabývat. Vstupními daty je množina uspořádaných dvojic vstupní proměnné a třídy (tzn. množina prvků se známou klasifikací). Podle těchto vstupních dat algoritmus stanovuje strukturu rozhodovacího stromu.

2.2 Algoritmy pro vytváření struktur rozhodovacích stromů a lesů

2.2.1 Principy algoritmů rozhodovacích stromů

Nejčastěji algoritmy používají techniky "Rozděl a panuj". Základní operací, kterou tyto algoritmy provádějí je rozdělování trénovací množiny v N-rozměrném prostoru (kde N je počet vstupních proměnných) na dvě podmnožiny (ve vizualizaci se to projeví jako hyperkvádry). Základem je postup, kde se využívá poznatků z teorie informace, konkrétně entropie (mírá neuspořádanosti systému), podle níž provádí výběr nejlepšího rozdělení, algoritmus se snaží nalézt takové rozdělení množiny M, aby entropie byla co nejnižší (=velká rozdílnost podmnožin množiny M). Dále se rozdělení rekurzivně provádí na obou takto vzniklých podmnožinách až do doby, než vzniklé množiny obsahují pouze prvky jedné třídy. Místo entropie se taktéž může používat gini index.
Takovýto postup má řadu problémů, např. přeučení, kdy se strom naučí s téměr stoprocentní jistou rozpoznávat prvky z dané vstupní množiny, ale pro jiné prvky je velmi nepřesný a dává chybné výsledky. Tento problém se řeší prořezáváním stromu, kde jsou odstraňovány některé listy ze stromu pomocí např. metody *error based* - používaný např. u algoritmu C4.5 a C5.0, kde se odřezávají větve podle toho, jak se zvětšuje chyba stromu při jejich ostranění (tzn. odstraňují se větve, jejichž absence se na výsledku projeví minimálně). Další metoda, *cost complexity*, zároveň zohledňuje počet uzlů každého stromu.

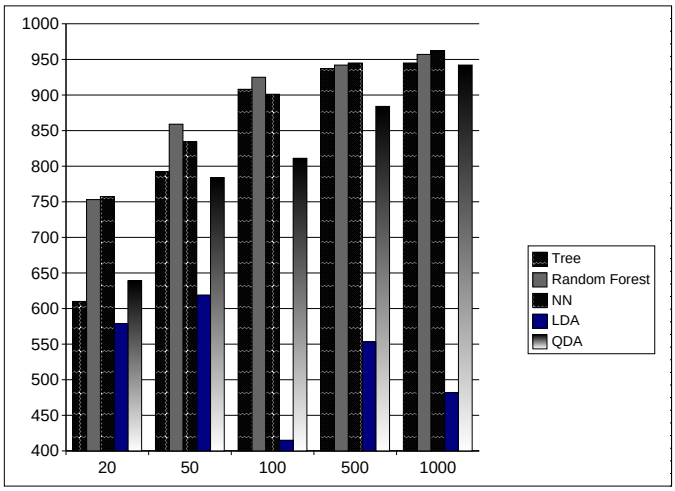
2.3 Rozhodovací lesy

V některých případech je použití jednoho rozhodovacího stromu nevýhodné, a proto přichází ke slovu jiná technika - a to rozhodovací lesy. Rozhodovací lesy nejsou nic jiného než množina rozhodovacích stromů, v nichž každý určí výsledek z daných vstupních dat, a následně je prováděn výběr nejdůvěryhodnějšího výsledku. Používá se např. většinového hlasování stromů, ale častější je jeho obměna, kdy každému stromu je přiřazena váha jeho hlasu.
Samostatnou kapitolou je vytvoření rozhodovacích stromů pro daný les, jelikož je nutno zajistit, aby stromy nebyly identické. Nejčastějšími technikami jsou:

- Bagging - v němž se provádí náhodný výběr prvků z trénovací množiny, kde pomocí každé takto vzniklé podmnožiny je vytvořen jeden rozhodovací strom.
- Randomizace - u ní se např. provádí náhodný výběr několika vstupních proměnných a na nich je prováděn nejlepší split (tuto metodu používá např. algoritmus *Random Forest*).
- Boosting - u něhož se provádí stanovení vah jednotlivých prvků - prvky špatně klasifikované dostávají větší váhu - příkladem použití je alg. *CART* a *C5.0*.

2.4 Porovnání některých nejpoužívanějších algoritmů

V tomto srovnání byla daným algoritmům dána trénovací množina od 20 do 1000 prvků a do grafu bylo promítnuto, v kolika případech byl později po natrénování algoritmus schopen provést správnou klasifikaci při ověřování. Velmi dobře si v testu vedl algoritmus NN, což bylo ovšem velkou měrou dáno charakterem testovacích dat, v nichž existovaly dvě třídy - první přibližně vyplňovala kruh a druhá tvořila jeho okolí. Naopak zde naprosto selhává algoritmus LDA, což je dáno tím, že tato data nejsou lineárně separabilní.



Obrázek 2: Porovnání algoritmů

3 Shrnutí

Problematika strojového učení a AI je v současné době velmi rozvíjející se oblastí, která neustále nabývá na významu a jenž nachází své uplatnění v širokém spektru problémů. Rozhodovací stromy představují poměrně jednoduchou, ale velmi mocnou metodu pro řešení těchto problémů.

Poděkování

Autoři by tímto chtěli poděkovat především našemu supervizorovi (Ing. Emilu Kotrči), fakultě FJFI ČVUT za organizaci Fyz. týdne a Janu Havlíkovi za zapůjčení fotoaparátu pro praktické studium stromů.

Reference

[1] Leo Breiman, Jerome Friedman, Charles J. Stone, R. A. Olshen, *Classification and Regression Trees*, Chapman & Hall/CRC, Boca Raton, 1998

[2] Devroye Luc, Györfi Laszlo, Lugosi Gabor, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, New York, 1996

[3] J. Ross Quinlan, *C4.5: programs for machine learning*, Morgan Kaufman Publishers, San Mateo, 1993