Niektoré z problémov pri získavaní dát pomocou rozhodovacích stromov

Ľuboš Kostík, Tomáš Saloky

Abstrakt
Hlavnou témou, ktorej sa venuje tento článok je popis jednej z časti umelej inteligencie - strojového učenia. Táto oblasť umelej inteligencie má v súčasnosti veľké uplatnenie, a to nielen staršie typy algoritmov rozhodovacích stromov (ID3, ID5R, C4.5 ...), ale aj novšie oblasti umelej inteligencie. ako učenie neurónových sietí, genetické algoritmy ... Tento článok popisuje štruktúru rozhodovacích stromov a jednoduchý algoritmus pre tvorbu týchto stromov. Popisuje výhody a nevýhody rozhodovacích stromov a poukazuje na niektoré problémy pri ich tvorbe.

Kľúčové slová: strojové učenie, rozhodovacie stromy, umelá inteligencia, ID3, ID5R, TDIDT

Úvod

Pri implementovaniu umelej inteligencie do života musíme vychádzať z poznania, že v rámci psychických procesov existujú určité zákonitosti, ktoré dokonale poznáme a môžeme ich namodelovať alebo naprogramovať do počítača.

Z tohto poznania vychádzame pri implementovaní umelej inteligencie. Aplikácie techník umelej inteligencie sa aplikujú a transformujú v čoraz rozsiahlejších technických oblastiach. K týmto oblastiam patrí aj strojové učenie. Na schopnosť učiť sa môžeme z technického pohľadu pozerať z viacerých. Preto strojové učenie netvorí jediná metóda, ktorá napodobňuje spôsob učenia sa ľudí, ale je to veľká množina metód a ich aplikácií v rôznych situáciách.

1. Rozhodovacie stromy

Rozhodovacie stromy patria k jedným zo základných princípov symbolických metód strojového učenia. Tvoria silný nástroj používaným na klasifikáciu a predikciu. Ich atraktívnosť spôsobuje fakt, že v porovnaní s neurónovými sieťami, rozhodovacie stromy reprezentujú pravidlá. Pravidlá je možné jednoducho vyjadriť prirodzeným jazykom, alebo priamo v databázovom jazyku.

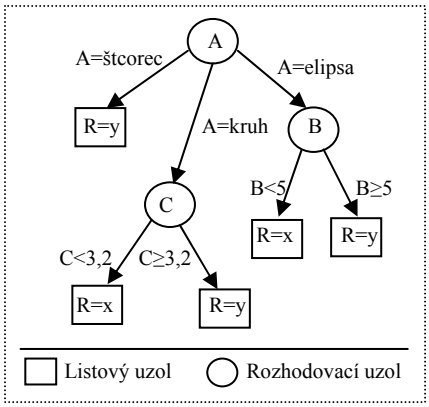
Základom rozhodovacích stromov je množina všetkých popisov konceptu s trénovacimi dátami, ktorý nazývame *priestor verzií* (version space).

Poznáme 3 typy algoritmov (Mitchell):
- Algoritmus generalizácie (*specific to general search*)
- Algoritmus špecializácie (*general to specific search*)
- Algoritmus eliminácie kandidátov (*candidate elimination algorithm*).

1.1 Induktívna tvorba rozhodovacích stromov

Metódy založené na spôsobe rozhodovacích stromov TDIDT (ID3, ID5R, C4.5, ...) sú v súčasnej dobe dobre rozpracované. Proces tvory rozhodovacieho stromu je z veľkej mieri induktívny, kde z existujúcich prípadov usudzujeme na vlastnosti príkladov, ktoré prídu až v budúcnosti. Hlavná heuristika riadiaca toto prehľadávanie sa najprv zoberá atribútmi, ktoré nesú najväčšie množstvo informácií. Tvorba rozhodovacieho stromu je procesom generalizácie.

Štruktúra rozhodovacieho stromu sa skladá z uzlov (obsahujú testy hodnôt atribútov) a „listov“ (obsahujú ohodnotenie). Pri klasifikácií prechádzame stromom zhora dole a vykonávame predpísané testy v jednotlivých uzloch. Výsledky jednotlivých testov nám potom určia ktorou cestou budeme pokračovať



Obr.1 Príklad rozhodovacieho stromu
Fig.1 Example decision trees

1.2 Vytváranie rozhodovacieho stromu

Najprv si nájdeme taký atribút ktorý v sebe nesie najväčšie množstvo informácie. Tento atribút sa stane koreňom stromu.

V ďalšom kroku si rozdelíme množinu príkladov na toľko podmnožín koľko je hodnôt koreňového atribútu. V každej podmnožine sú príklady s jedinou hodnotu tohto atribútu. Potom vyhľadáme v každej podmnožine ďalší najvýznamnejší atribút, a takto sa pokračuje pokiaľ nevyčerpáme všetky atribúty, alebo príklady.

Pre výber najvýznamnejšieho atribútu existuje viacero kritérií. Dobré kvantitatívne meranie vhodnosti atribútu poskytuje štatistická vlastnosť nazývaná informačný zisk, ktorá udáva mieru do akej atribút rozdeľuje tréningové príklady do ich cieľovej klasifikácie. Aby sme precízne stanovili informačný zisk, musíme zadefinovať entropiu, ktorá charakterizuje (ne)čistoty v ľubovoľnej skupine príkladov. Daná je množina H obsahujúca iba pozitívne a negatívne príklady nejakého cieľového konceptu. Potom entropia množiny H zodpovedajúcej tomuto jednoduchému príkladu binárnej klasifikácie je definovaná ako:

$$H_{aj} = -p_1 \cdot \log_2 p_1 - p_2 \cdot \log_2 p_2 \quad (1)$$

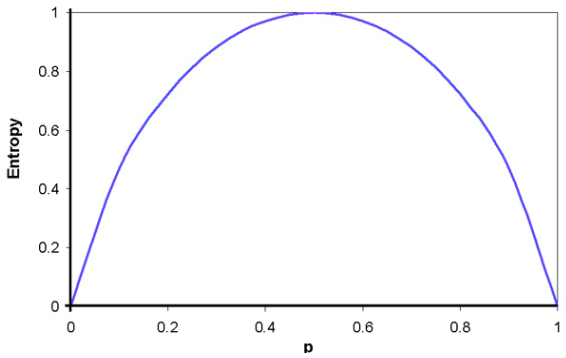
$$H(A) = \sum_{j=1}^{k} P(aj) \cdot \ (aj) \quad (2)$$

$$H(C) = -\sum_{j=1}^{k} P(cj) \cdot H(cj) \quad (3)$$

H(A) – je entropia A-tej podmnožiny, p_1 – je pomer pozitívnych príkladov, p_2 – je pomer negatívnych príkladov, *H(c)* – je entropia kvalifikačných tried, k – je počet podmnožín indukovaných daným atribútom, $P_{(aj)}$ – je pomer veľkosti A-tej podmnožiny k množine všetkých príkladov

Zo získaných entropií jednotlivých uzlov si následne odvodíme *informačný zisk* (4)

$$I(A) = H(C) - H(A) \quad (4)$$



Obr.2 Tvar krivky entropie

Fig.2 Waveform entropy

2. ID3 a ID5R algoritmus

Patria medzi najznámejšie systémy pre klasifikáciu objektov z tréningových príkladov. Začína s množinou príkladov {i_1 , i_2 , i_3 ,..., i_n } opísaný ich typických atribútov a_1 , a_2, a_3 , ... , a_n . Atribúty majú niektoré možné hodnoty U_{j1} U_{j2}, U_{j3}, ... , U_{jk} špecifikujúce samotný príklad (napr. veľkosť, farba, váha).

2.1 Spôsob vytváranie rozhodovacieho stromu ID3

Vytvorenie rozhodovacieho stromu algoritmom ID3 prebieha podľa nasledujúceho systému:

- Vyberieme jeden atribút ako koreň stromu
- Množina trenovacích príkladov sa rozdelí na podmnožiny podľa hodnôt tohto atribútu
- Postupne sa spracujú všetky tieto podmnožiny takto:
  a) Obsahuje podmnožina iba príklady z rovnakej triedy, vytvorí sa pre túto podmnožinu listový uzol a ohodnotí sa indexom príslušnej triedy.
  b) V opačnom prípade sa vyberie ďalší atribút ako atribút podstromu. Tento atribút sa potom spracuje podľa krokov 2 a 3.

ID3 požaduje, aby trénovacie príklady boli neprotirečivé, neredundantné, samozrejmá je požiadavka vzájomnej nezávislosti atribútov. Hlavné nevýhody ID3 sú najmä neinkrementálna indukcia a nevyužívanie informácie z predchádzajúcej indukcie.

2.2 Spôsob vytváranie rozhodovacieho stromu ID5R

Algoritmus ID5R v každom uzle uchováva dostatočnú informáciu na to, aby sa rozhodol, či daný atribút nahradí iným alebo nie. Táto informácia sa potom použije na reštrukturalizáciu podstromov. Algoritmus ID5R pracuje podobne ako ID3 s informačným ziskom a ide o inkrementálny algoritmus.

Tvorba stomu:

- Ak je strom prázdny, vytvor nový uzol, pridaj nový príklad do uzla a triedu v uzle nastav na triedu nového trénovacieho príkladu.
- Ak je strom v neexpandovanej forme a zároveň trénovacie príklady v uzle majú rovnakú triedu ako nový trénovací príklad, potom nový trénovací príklad pridáme k uzlu, inak bod 3.
- Ak je strom v neexpandovanej forme, potom expandujeme o jednu úroveň, testovací atribút zvolíme náhodne.
  a) Aktualizujeme počty príkladov pre všetky triedy, všetky hodnoty všetkých atribútov.
  b) Ak uzol obsahuje testovací atribút TA, ktorý nemá najnižšiu entropiu, potom
  c) Reštrukturalizujeme celý strom tak, aby sa atribút s minimálnou entropiou dostal do koreňa.
     - Rekurzívne reštrukturalizujeme všetky podstromy, okrem 3.d) .
     - Rekurzívne aktualizujeme rozhodovací strom pod aktuálnym uzlom pozdĺž vetvy s hodnotou atribútu, ktorá sa vyskytla v novom trénovacom príklade. Ak taká vetva neexistuje, potom sa vytvorí.
  d) Pozn. Ak by sme vypustili bod 3.3.2, potom by sme dostali algoritmus ID5, ktorý by nezaručoval minimálnosť stromu, ale bol by výpočtovo menej náročný.

V súčasnosti poznáme ďalší rad algoritmov založených na rozhodovacích stromov (napr. C4.5), ktoré dopĺňajú a modifikujú algoritmus ID3.

3. Problémy rozhodovacích stromov pri získavaní dát

Ako pri každom systéme , tak aj pri učení sa rozhodovacích stromov sa stretávame s rôznymi problémami.

- jeden z najbežnejších a najzávažnejších problémov je s určovaním hĺbky, do ktorej má rozhodovací strom narásť
- spracovanie spojitých atribútov
- výber vhodnej selekcie atribútu
- spracovanie tréningových dát s chýbajúcimi hodnotami atribútov
- spracovanie atribútov s rozdielnym ohodnotením a zvyšovanie efektívnosti výpočtu

Ako sme si už všimli vytvorenie rozhodovacieho stromu môže byť pomerne rozsiahly proces, pri ktorom môže natrafiť na viacero problémov. Tieto problémy môžu znížiť jeho zrozumiteľnosť.

V systémoch rozhodovacích stromov môže dôjsť k „preučeniu“ (overfitting) rozhodovacieho stromu, t.j. dosiahnutie neúmernej presnosti stromu (trénovacie dáta sú zaťažené šumom). Strom je možno zjednodušiť tak, že namiesto toho aby listovému uzlu odpovedali iba príklady jednej triedy, sa

uspokojíme s tým, že príklady jednej triedy budú v listovom uzle prevažovať.

Zjednodušenie (redukciu stromu) je možno urobiť dvoma spôsobmi:

- algoritmus sa modifikuje doplnením nejakého kritéria, ktoré indikuje či má uzol ďalej expandovať. Týmto spôsobom sa redukovaný strom vytvorí priamo.
- Vytvorí sa úplný strom a následne sa prevedie jeho prerezávanie (*post-prunning*). Postupuje sa zdola nahor a u každého podstromu sa podľa nejakého kritéria rozhoduje, či sa má podstrom nahradiť listovým uzlom. Tento spôsob sa používa častejšie.

3.1 Výhody rozhodovacích stromov

- sú schopné generovať pochopiteľné pravidlá.
- dosahujú klasifikáciu bez potreby prílišného počítania.
- sú schopné pracovať s kontinuálnymi aj kategorickými premennými.

3.2 Nevýhody rozhodovacích stromov

- sú menej vhodné pre výpočet úloh, kde cieľom je predikcia hodnôt kontinuálneho atribútu.
- sú náchylné k chybám v klasifikácii problémov s mnohými triedami a relatívne malým počtom tréningových príkladov.
- môžu byť výpočtovo náročné na trénovanie. Proces rastu rozhodujúceho stromu je výpočtovo náročný.

Záver

Strojové učenie a algoritmy rozhodovacích stromov nám stále viacej slúžia k získavaniu znalosti ktoré môžeme ďalej spracovávať v rôznych oblastiach. Hlavným prínosom je najmä ich schopnosť pracovať s údajmi, ktoré nie sú úplné alebo sa v nich vyskytujú chyby. Cieľom nebolo popísať podrobný postup pri tvorbe rozhodovacieho stromu ale poukázať na výhody a nevýhody rozhodovacích stromov. Využitie rozhodovacích stromov je široké, od ekonomických oblastí kde potrebujeme prehľadávať veľké množstvá dát po oblasti čisto technické (rozpoznávanie obrazov, navigácia, riadenie ...).

Literatúra

[1] MACHOVÁ, K.: Strojové učenie princípy a algoritmy. Košice 2002. ISBN 80-89066-51-8

[2] MAŘÍK, V., ŠTEPÁNKOVÁ, O., LAŽANSKÝ, J. a kol.: Umělá inteligence 1, Academia Praha 1993. ISBN 80-200-0496-3

[3] MAŘÍK, V., ŠTEPÁNKOVÁ, O., LAŽANSKÝ, J. a kol. : Umělá inteligence 2, Academia Praha 2003. ISBN 80-200-0504-8

[4] SALOKY, T., Some Problems of AI Impelmentation (pp 165-194), In: Tauer, I., Hrubina, K., Eds*: Optimal control of processes based on the use of informatics methods*, Informatech Košice, 2005. ISBN 80-88941-30-X

[5] SWERE E., MULVANEY D. J.: Robot Navigation Using Decision Trees. Electronic systems and control division research 2003

[6] SMUC T., DM Tutorial - Decision trees, [cit.:4.11.2004] http://dms.irb.hr/tutorial/tut_dtrees.php

[7] The ID3 Algorithm, [cit.:10.10.2005]. http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm

Abstract

The main goal of this paper is to describe an artificial intelligence – machine learning. These areas have a big remark today, and not only the old types of algorithms decision tress (TDIDT, ID3, ID5R...), but also a newest areas of artificial intelligence as learning of neural networks, genetic algorithms... This article describes the structure of decision trees and the basic algorithm for construction of these trees. It describes strengths and weakness of decision tree methods and names appropriate problems for decision tree learning.

Ing. Ľuboš Kostík

Technická univerzita Košice
Strojnícka fakulta
Katedra automatizácie a riadenia
Park Komenského 9
04200 Košice
Tel.: 0556022598
e-mail: lubos.kostik@tuke.sk

prof. Ing. Tomáš Saloky, CSc.

Technická univerzita Košice
Strojnícka fakulta
Katedra automatizácie a riadenia
Park Komenského 9
04200 Košice
e-mail: tomas.saloky@tuke.sk