Implementace GUHA rozhodovacích stromů

Martin Ralbovský, Petr Berka

Katedra informačního a znalostního inženýrství
Fakulta informatiky a statistiky, Vysoká škola ekonomická
martin.ralbovsky@gmail.com, berka@vse.cz

Abstrakt. Rozhodovací stromy jsou jedním z osvědčených způsobů řešení klasifikačních úloh. Článek navrhuje modifikaci klasických algoritmů konstrukce rozhodovacích stromů ve smyslu metody GUHA. Tímto smyslem je generování více rozhodovacích stromů jakožto GUHA verifikací. Představujeme proceduru implementující GUHA rozhodovacích stromů ETree a testujeme užitečnost procedury na příkladech.

Klíčová slova: Metoda GUHA, rozhodovací stromy, ETree, Ferda

1 Úvod

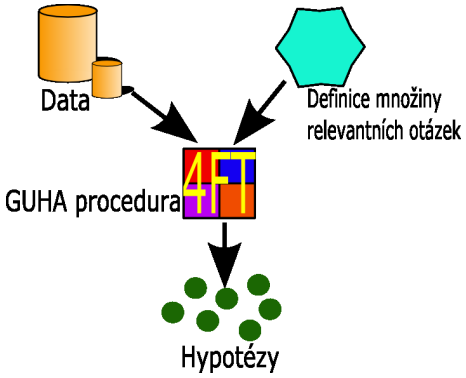
Náš přístup kombinuje dvě metody dobývání znalostí z dat. Metoda GUHA je původní česká metoda exploratční analýzy dat, jejíž hlavním cílem je nabídnout uživateli vše zajímavé v datech [2, 3]. Naproti tomu algoritmy tvorby rozhodovacích stromů [8] patří k nejznámějším algoritmům v oblasti symbolických metod strojového učení.

Základní myšlenkou v příspěvku je vytvoření více rozhodovacích stromů v důsledku modifikace klasických algoritmů pro indukci rozhodovacích stromů, tyto stromy lze poté považovat za GUHA verifikace/hypotézy a pro uživatele vybírat jen nejlepší z nich. Navržená procedura pojmenována ETree byla implementována v systému Ferda a byly provedeny počáteční experimenty ukazující konstrukci stromů lepších charakteristik, než byly stromy původní.

Práce je strukturována následovně: v sekci 2 a 3 seznamujeme čtenáře se základy metody GUHA a algoritmy pro tvorbu rozhodovacích stromů. Sekce 4 popisuje nově navrženou proceduru ETree a její implementaci v prostředí Ferda. Sekce 5 shrnuje provedené experimenty a konečně sekce 6 uzavírá práci a poskytuje náměty na další výzkum.

2 Metoda GUHA

Metoda GUHA je jednou z prvních metod exploratční analýzy dat se vznikem v polovině šedesátých let v Praze. Metoda poskytuje obecný rámec k získávání potenciálně zajímavých znalostí z dat, který má silné teoretické základy v logice (observační kalkuly) a statistice [2, 3]. Pro potřeby tohoto příspěvku se omezíme



Obr. 1. Metoda GUHA

pouze na vysvětlení základního principu metody GUHA, tak jak je zobrazeno na obrázku 1.

Metoda je realizovaná pomocí GUHA procedur. Vstupy tvoří data a jednoduché zadání potenciálně velké množiny relevantních otázek. Procedura automaticky generuje všechny relevantní otázky a verifikuje je oproti datům. Výstupem procedury jsou relevantní otázky, které jsou v datech pravdivé (hypotézy v terminologii metody).

Metoda GUHA se v současnosti nejvíce používá pro hledání *zobecněných asociačních pravidel*, k čemuž slouží procedura 4FT (*4ft-miner*) [12, 9]. Nejnovějšími systémy podporujícími metodu GUHA jsou systém *LISp-Miner*[14] a Ferda[6], oba vyvíjené na KIZI FIS VŠE.

3 Rozhodovací stromy

Podle [5] jsou rozhodovací stromy vůbec nejpoužívanější technikou dobývání znalostí z databází. Za svou popularitu vděčí zejména intuitivní interpretovatelnosti a pochopitelnému způsobu konstrukce rozhodovacích stromů. Algoritmus konstrukce (indukce) rozhodovacích stromů vysvětlíme podrobněji, neboť bude jeho pochopení důležité pro další text.

Při tvorbě rozhodovacího stromu se postupuje metodou *rozděl a panuj*. Trénovací data se postupně rozdělují na menší a menší podmnožiny (uzly stromu) tak, aby v těchto podmnožinách převládaly příklady jedné třídy. Tento postup bývá často nazýván "top down induction of decision trees" (TDIDT)[8]. Obecné schéma algoritmu pro tvorbu rozhodovacích stromů je v tabulce 1.

Uvedený algoritmus bude fungovat pouze pro kategoriální data (počet podmnožin - uzlů v bodě 2 odpovídá počtu hodnot daného atributu), která nejsou zatížena šumem. Uvedeným způsobem pracuje např. Quinlanův algoritmus ID3 [8]. Tato dvě omezení lze samozřejmě překonat odstraněním šumu či kategorizací reálných numerických dat ve fázi předzpracování dat.

Tabulka 1. Obecný algoritmus pro tvorbu rozhodovacích stromů

TDITD Algoritmus
1. zvol jeden atribut jako kořen dílčího stromu, 2. rozděl data v tomto uzlu na podmožiny podle hodnot zvoleného atributu a přidej uzel pro každou podmnožinu, 3. existuje-li uzel, pro který nepatří všechna data do téže třídy, pro tento uzel opakuj postup od bodu 1, jinak skonči.

Požadavek na konzistenci stromu s (trénovacími) daty (3. bod algoritmu) bývá v současných implementacích změkčen buď následným prořezáním, nebo požadavkem na zastavení růstu stromu, pokud v uzlu patří do požadované třídy dostatečný počet prvků, ať už v absolutním či relativním vyjádření. V druhém případě mluvíme o *čistotě uzlu*[1].

Klíčová otázka celého algoritmu je jak vybrat vhodný atribut pro větvení stromu (bod 1). Cíl je zřejmý: vybrat takový atribut, který od sebe nejlépe odliší příklady různých tříd atributu, podle kterého se má klasifikovat. Výpočet vychází z kontingenční tabulky zobrazené v tabulce 2.

Tabulka 2. Kontingenční tabulka pro výběr atributu pro větvení stromu

	Y_1	Y_2	.....	Y_S	Σ
X_1	a_{11}	a_{12}	.....	a_{1S}	r_1
X_2	a_{21}	a_{22}	.....	a_{2S}	r_2
:	:	:		:	:
:	:	:		:	:
X_R	a_{R1}	a_{R2}	.....	a_{2S}	r_R
Σ	s_1	s_2	.....	s_S	n

Y_i jsou v tabulce třídy klasifikačního atributu, X_i jednotlivé atributy a $a_{i,j}$ je četnost kombinace $(X = X_i) \wedge (Y = Y_j)$. Dále

$$r_i = \sum_{j=1}^{S} a_{ij}, s_j = \sum_{i=1}^{S} a_{ij}, n = \sum_{i=1}^{R} \sum_{j=1}^{S} a_{ij}.$$

Uveďme čtyři vzorce pro posouzení vhodnosti atributu. Kritérium entropie má podobu:

$$H(X) = \sum_{i=1}^{R} \frac{r_i}{n} \left(- \sum_{j=1}^{S} \frac{a_{ij}}{r_i} \log_2 \frac{a_{ij}}{r_i} \right),$$

přičemž preferujeme atribut s nejnižší hodnotou kritéria. Kritérium Gini indexu má podobu:

[1] Čistota uzlu je poměr správně klasifikovaných prvků ku všem prvkům

$$Gini(X) = \sum_{i=1}^{R} \frac{r_i}{n} \left(1 - \sum_{j=1}^{S} \left(\frac{a_{ij}}{r_i} \right)^2 \right),$$

přičemž se preferujeme atribut s nejnižší hodnotou kritéria. Kritérium χ^2 má podobu:

$$\chi^2(X) = \sum_{i=1}^{R} \sum_{j=1}^{S} \frac{\left(a_{ij} - \frac{r_i \cdot s_j}{n} \right)^2}{\frac{r_i \cdot s_j}{n}},$$

přičemž preferujeme atribut s nejvyšší hodnotou kritéria. Konečně kritérium vzájemné informace má podobu:

$$MI(X) = - \sum_{i=1}^{R} \sum_{j=1}^{S} \frac{a_{ij}}{n} \log_2 \frac{\frac{a_{ij}}{n}}{\frac{r_i}{n} \cdot \frac{s_j}{n}},$$

přičemž opět preferujeme atribut s nejvyšší hodnotou kritéria.

Použití rozhodovacího stromu pro klasifikaci nových případů je velmi prosté. Počínaje kořenem stromu se postupně zjišťují hodnoty atributů. Konkrétní hodnota odpovídá určité větvi, která nás přivede k dalšímu atributu, až se dostaneme do listového uzlu, který odpovídá třídě, do které máme nový příklad zařadit.

4 Procedura ETree

4.1 Návrh procedury

Algoritmy pro tvorbu rozhodovacích stromů postupují gradientním způsobem, jednou vybrané větvení se již nemění a výsledkem algoritmu je jeden strom. Navrhovaný algoritmus pro tvorbu *exploračních stromů* představuje jednoduché rozšíření tohoto přístupu, které ale více vyhovuje základní myšlence metody GUHA: hledat v datech "vše" zajímavé. Algoritmus je ve velmi zjednodušené podobě zobrazen v tabulce 3.

Tabulka 3. Algoritmus ETree pro tvorbu exploračních stromů

Algoritmus ETree
1. forall atribut vhodý jako kořen dílčího stromu, 2. rozděl data v tomto uzlu na podmnožiny podle hodnot zvoleného atributu a přidej uzel pro každou podmnožinu, 3. existuje-li uzel, pro který není splněno kritérium zastavení, pro tento uzel opakuj postup od bodu 1, jinak skonči.

Navrhovaný algoritmus se od klasických algoritmů pro tvorbu rozhodovacích stromů (tabulka 1) liší:

1. v použití více atributů pro větvení v určitém místě stromů (bod 1 algoritmu)
2. v různých variantách kritéria pro zastavení růstu stromu (bod 3 algoritmu)

Jako kritérium vhodnosti atributu pro větvení bylo vybráno χ^2 (které je zároveň tradičním kvantifikátorem metody GUHA). Toto kritérium umožňuje nejen uspořádat atributy podle vhodnosti, ale i vyhodnotit sílu závislosti mezi uvažovaným atributem a třídou.

Co se týče kritéria pro zastavení růstu stromu týká, uvažujeme *minimální čistotu listu*, což je obecnější požadavek než náležení všech prvků listu do jedné třídy (bod 3 algoritmu TDITD) a dále také požadavek na *minimální frekvenci uzlu*. Tyto kritéria mohou platit samostatně nebo v disjunkci.

Procedura tedy generuje značné množství stromů. Horní odhad počtu stromů vyjadřuje následující vzoreček. Předpokládáme nepoužití žádného kritéria pro zastavení větvení kromě dosažení maximální hloubky.

$$NT = k \cdot \prod_{i=1}^{i_{max}} k^{v_i}$$

kde k je počet atributů použitých pro větvení, i je hloubka stromu a v_i je počet uzlů hloubky i. V reálných případech hodnoty v_i závisí na datech. Proto ho ve výpočtech nahrazujeme největším počtem kategorií u vstupních atributů, což opět odhad zvyšuje.

Je zřejmé, že ne všechny vytvořené stromy budou dobře klasifikovat vstupní data. Kvalitu jednotlivých stromů můžeme hodnotit na základě tzv. *matice záměn* (confusion matrix), která ukazuje počet správně a chybně klasifikovaných příkladů.

Tabulka 4. Matice záměn

	Klasifikace systémem	
Správné zařazení	+	−
+	TP	FN
−	FP	TN

Tabulka 4 zobrazuje matici záměn, kde TP (správně pozitivní, true positive) je počet prvků, které systém správně zařadil do třídy +, FP (falešně pozitivní, false positive) je počet příkladů, které systém chybně zařadil do třídy + (patří do třídy −) atd. Kvalitu stromu můžeme hodnotit funkcí $f : (TD, FN, FP, TN) \rightarrow \langle 0, 1 \rangle$. Používaná je například *F-míra* [13].

$$F = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Matice záměn formálně odpovídá čtyřpolní kontingenční tabulce známé z dobývání zobecněných asociačních pravidel [10]. Pro hodnocení kvality stromu můžeme použít také libovolný vhodný *4ft-kvantifikátor* [10]. V této souvislosti je

vhodné poznamenat, že F-míra patří do třídy *dvojitě implikačních kvantifikátorů*, jak definováno v [10]. Důkaz tohoto tvrzení je jednoduchý a bude analogický důkazu použitém v příkladu 3 v [10].

4.2 Parametry ETree

Ve smyslu metody GUHA můžeme na výstupní stromy nahlížet jako na hypotézy GUHA[2].

V sekci 2 byly definovány vstupy pro GUHA procedury, z nichž nás bude zajímat definice množiny relevantních otázek pro proceduru ETree. Oproti jiným procedurám metody GUHA jsou to spíše parametry algoritmu. Logicky se dají rozdělit na čtyři skupiny: **parametry pro růst**, **pro větvení**, **pro zastavení větvení** a **pro uložení hypotézy**.

Parametry pro růst stromu byly vysvětleny v předcházejícím textu a jsou to:

- cílový atribut rozdělující data do tříd
- vstupní atributy pro větvení
- počet atributů použitých pro větvení

Přeskočme parametry pro větvení a vyjmenujme nejdříve **parametry pro zastavení větvení**, neboť první tři z nich byly také vysvětleny:

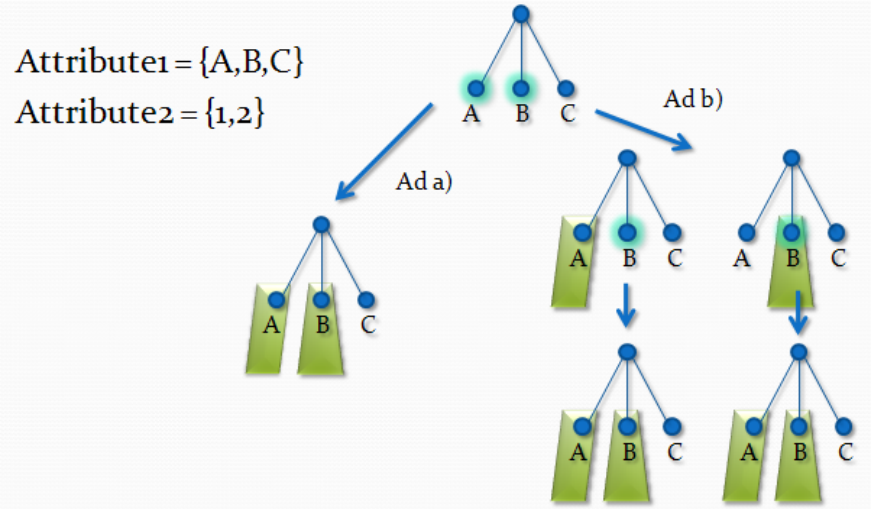
- minimální frekvence uzlu
- minimální čistota listu
- výběr kritéria pro zastavení větvení (minimální frekvence uzlu, minimální čistota listu, minimální frekvence uzlu *nebo* minimální čistota listu)
- maximální hloubka stromu

Význam posledního parametru je zřejmý, udává přípustnou hloubku stromu. Po dosažení této hloubky se již strom dále nevětví.

Parametr pro větvení v původním návrhu procedury nebyl, vznikl až v průběhu implementace. Obrázek 2 ukazuje odlišné možnosti větvení uzlů a vniku nových stromů. Uzly A a B byly vybrány algoritmem jako vhodné pro větvení. Můžeme postupovat jako v klasických algoritmech větvením všech vhodných uzlů najednou (zde uzly A i B - situace a), nebo pro každý uzel vytvoříme nový strom a v tomto stromu větvíme pouze vybraný uzel (situace B). Jak ukazuje obrázek, nevýhoda druhého postupu je vznik více stromů na výstupu, z nichž některé budou identické. Zde jsme se rozhodli ponechat volbu na uživateli a vytvořili jsme parametr pro větvení nazvaný *větvení uzlů po jednom* (individual nodes branching).

Poslední skupinou jsou **parametry pro uložení hypotézy**:

[2] Rozhodovací stromy je možné srovnat s asociačními pravidly, jakožto nejpoužívanějšími druhy hypotéz metody GUHA. Jednu cestu v rozhodovacím stromu od kořene k listu lze přirovnat k asociačnímu pravidlu, kde levá strana pravidla je konjunkce atributů a jejich hodnot odpovídajícím uzlům v cestě a pravá strana klasifikovaná kategorie. Celý rozhodovací strom si bude potom tvořit disjunkci asociačních pravidel odpovídajících všem cestám. Takto může i procedura 4FT vrátit stejný výsledek jako ETree, ovšem s exponenciální časovou složitostí.



Obr. 2. Větvení uzlů po jednom

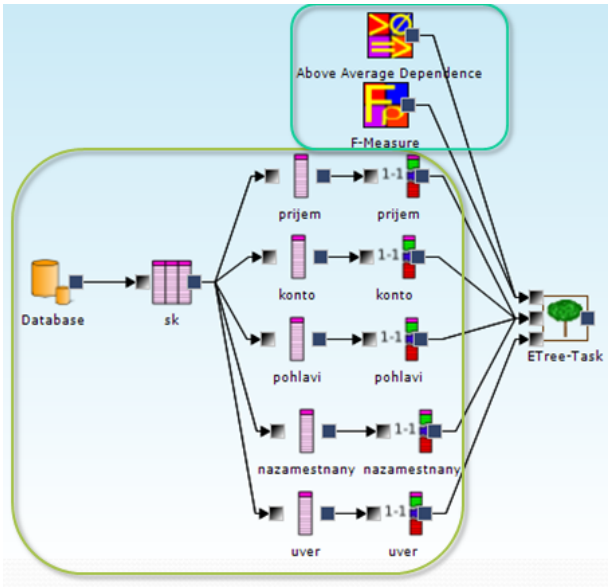
- minimální kvalita stromu vyjádřena pomocí matice záměn a konjunkcí 4ft-kvantifikátorů
- pouze maximální stromy
- maximální počet hypotéz

Funkci matici záměn jsme popsali v sekci 4.1. Booleovský parametr *pouze maximální stromy* určuje, jestli se na výstup mají dostat všechny stromy, nebo pouze stromy požadované délky (v případě nedosažení této délky to budou stromy o délce rovnající se počtu vstupních atributů pro větvení). Poslední parametr udává počet vygenerovaných hypotéz, po kterých se výpočet přeruší.

4.3 Implementace

Procedura ETree byla implementována v prostředí Ferda[6]. Ferda je vizuální a vysoce modulární systém pro dobývání znalostí a je zároveň nejmladší implementací metody GUHA. ETree je již devátá GUHA procedura v systému. Použili jsme stávající vrstvu předzpracování dat a engine pro práci s bitovými řetízky *mining processor* [7]. Vznikla krabička ETree (krabička je ve Ferdovi prvek vizuálního programování, více v [6]), jejíž zapojení ukazuje obrázek 3.

Na obrázku je řešena jednoduchá klasifikační úloha, zjišťujeme jestli dát klientovi úvěr v závislosti na jeho příjmu, výši jeho konta, pohlaví a zaměstnanosti. Zelený čtverec zobrazuje stávající implementované krabičky, které slouží ke konstrukci atributů. Za ně je zapojena krabička ETree. Modrý čtverec jsou 4ft-kvantifikátory určené pro měření kvality pravidla. Výhodou tohoto řešení je znovupoužitelnost pro uživatele - ETree se dá zapojit do krabiček pro konstrukci atributů stejným způsobem, jako každá jiná GUHA procedura implementovaná v systému Ferda.



Obr. 3. Zapojení ETree v prostředí Ferda

5 Experimenty

Pro otestování procedury bylo provedeno několik experimentů. Experiment s datovým zdrojem *Barbora* popisuje část 5.1. Dva experimenty s datovým zdrojem *Forest cover type* popisují části 5.2 a 5.3.

5.1 Barbora

Jako první testovací data jsme vybrali data klientů hypotetické banky Barbora [11]. Databáze neobsahuje reálná data, pouze záznamy předgenerované na základě znalostí domény.

První experiment byl proveden nad tabulkou *loans*, která obsahuje informace o klientech a jeho statutu (celkem zhruba 6100 záznamů). Experiment konstruoval stromy, které zjišťovaly statut klienta (rozdělení do 4 tříd) z informací o okresu, platu klienta a výši a trvání jeho půjčky. Parametry zadání procedury byly následující:

- počet atributů pro větvení = 4
- minimální čistota listu = 0,8
- minimální frekvence uzlu = 61 (1% dat)
- větvení uzlů po jednom = true
- minimální kvalita stromu: F-míra s proměnlivým prahem (viz. dále)

Tabulka 5. Experiment Barbora

Hloubka stromu	**Práh F-míry**	**Verifikací**	**Hypotéz**	**Nejlepší hypotéza** (F-míra)
1	0,5	5	2	0,75
2	0,7	17	7	0,88
3	0,85	193	26	0,88
4	0,85	7910	222	0,90

Cílem experimentu bylo zjistit, jestli může procedura vygenerovat více stromů podobné kvality (při zvyšující se hloubce stromu). Byly použity zvětšující se prahy F-míry a výsledky jsou zobrazeny v následující tabulce.

Uspokojivé výsledky experimentu potvrdily naši domněnku, že lze zkonstruovat více stromů podobné kvality. Například pro hloubku 4 bylo sestrojeno celkem 222 rozhodovacích stromů, které se liší v hodnotě F-míry pouze pěti procenty. Data *Barbora* jsou však uměle vygenerována (částečně nedokonale), což zřejmě způsobilo tak dobré výsledky. Dalším krokem bylo otestovat proceduru na reálném datovém zdroji.

5.2 Forest cover type - vzorek

Tímto zdrojem byly data *Forest cover type* z datového repositáře UCI KDD [4][3]. Data Forest cover type obsahují informace o pokrytí lesa pro buňky 30x30 metrů získané z US Forest Service Region 2 Resource Information System.

V druhém experimentu jsme konstruovali stromy, které určovaly výsledný typ pokrytí lesa z informací o chráněném území, nadmořské výšce, sklonu svahu, horizontální a vertikální vzdálenosti k vodnímu zdroji a horizontální vzdálenosti k výskytu požáru. Vstupních atributů bylo celkem 8 a klasifikační atribut měl 7 různých hodnot. Cílem bylo prověřit domněnku, že použití více atributů pro větvení dává kvalitnější stromy. Experiment byl proveden na vzorku původních dat o velikosti 10 000 záznamů z původních asi 600 000 záznamů. Vzorek jsme zkonstruovali vybráním každého zhruba 60. vzorku z původních dat. Parametry zadání procedury byly následující:

- počet atributů pro větvení = 1,3,5
- minimální čistota listu = 0,8
- minimální frekvence uzlu = 100 (1% dat)
- větvení uzlů po jednom = false
- minimální kvalita stromu: F-míra s prahem 0.5

Tabulka 6 ukazuje výsledky experimentu. Nárůst kvality stromů je výrazný zejména u delších stromů z 1 atributu pro větvení na 3 atributy pro větvení, kde například pro stromy délky 4 je to asi 30% F-míry[4].

[3] Jeden z nejuznávanějších repozitářů dat pro benchmarking metod strojového učení.

[4] Cílem experimentu nebylo zkonstruovat co nejkvalitnější strom, nýbrž zjistit růst kvality stromů. Proto jsme použili pouze základní kategorizaci vstupních atributů a nesnažili jsme se primárně o nějaké její vylepšování.

Tabulka 6. Experiment Forest cover type - vzorek

Hloubka stromu	**Verifikací**	**Hypotéz**	**Nejlepší hypotéza** (F-míra)
Počet atributů pro větvení = 1			
1	2	0	0,304
2	3	0	0,304
3	4	0	0,385
4	5	1	0,511
Počet atributů pro větvení = 3			
1	4	0	0,304
2	13	0	0,305
3	40	1	0,512
4	121	3	0,816
Počet atributů pro větvení = 5			
1	6	0	0,304
2	9	0	0,385
3	156	5	0,515
4	781	103	0,816

5.3 Forest cover type

Třetí experiment byl zjednodušením druhého experimentu na celou datovou matici (přibližně 600 000 záznamů), vynechali jsme 5 atributů pro větvení. Zadání bylo následující:

- počet atributů pro větvení = 1,3
- minimální čistota listu = 0,8
- minimální frekvence uzlu = 6000 (1% dat)
- větvení uzlů po jednom = false
- minimální kvalita stromu: F-míra s prahem 0,5

Tabulka 7. Experiment Forest cover type

Hloubka stromu	**Verifikací**	**Hypotéz**	**Nejlepší hypotéza** (F-míra)
Počet atributů pro větvení = 1			
1	2	0	0,304
2	3	0	0,305
3	4	0	0,387
4	5	1	0,807
Počet atributů pro větvení = 3			
1	2	0	0,305
2	16	0	0,305
3	40	2	0,522
4	121	86	0,818

Tabulka 7 ukazuje výsledky experimentu. Experiment ukázal podobné výsledky jako experiment předchozí s výjimkou stromů délky 4, kdy byl ve všech

datech nalezen strom s F-mírou 0,807 a zlepšení přidáním atributu bylo pouze 1%.

Shrnutím provedených experimentů je konstatování, že má význam tvořit stromy pomocí algoritmu ETree. V některých datech se konstrukce stromů z více atributů projeví ve větší kvalitě techto stromů, ve všech zkoumaných datech získáváme velké množství vysoce kvalitních stromů.

6 Závěr

6.1 Shrnutí vykonané práce

Práce obohacuje tradiční algoritmy na konstrukci rozhodovacích stromů o aspekty metody GUHA pro explorační analýzu dat. Vzniká nový algoritmus ETree, který konstruuje více stromů na základě výběru více atributů pro větvení uzlů a větvení pouze jednotlivých uzlů. Algoritmus chápeme v názvosloví metody GUHA také jako novou GUHA proceduru.

Byla provedena implementace procedury ETree v prostředí Ferda, kde je ji možno zapojit do stávajících modulů po předzpracování dat stejným způsobem jako ostatní GUHA procedury. Byly dále provedeny experimenty, které ukazují použitelnost našeho přístupu. Konstrukce více stromů z více atributů se pozitivně projeví na kvalitě nejlepšího stromu i na množství vysoce kvalitních stromů.

6.2 Směry dalšího výzkumu

ETree nabízí zajímavé možnosti dalšího výzkumu. Jednou z možností rozšíření je vybírat atributy i podle jiného kritéria než χ^2(další kritéria byly uvedeny v sekci 4.1). Z hlediska návrhu by bylo nejlepší vytvořit pro výběr atributů novou krabičku a zapojovat ji do krabičky ETree.

Aby byl nástroj plně použitelný je dále třeba implementovat klasifikátor samotných dat, který ještě není hotový. Prozatím jsme se zajímali pouze o konstrukci stromů a ne jejich použití při klasifikaci nových dat.

Odhad počtu relevantních otázek (počtu vygenerovaných stromů) je zajímavý statistický problém. Prozatím počítáme relevantní otázky pomocí nejhorších případů, což je zejména pro rozsáhlejší úlohy nepřesné. Průměrné či jinak významné případy by mohli pomoci, problém je však zajímavý zejména tím, že již před prvním větvením jakéhokoli z atributů by měl být odhad k dispozici.

Jeden z plánů rozvoje Ferdy je implementace fuzzy metody GUHA, zejména implementace fuzzy bitových řetízků a kvantifikátorů. Tyto změny by se samozřejmě týkaly i procedury ETree, zejména změny bitových řetízků na fuzzy. Jakým způsobem to ovlivní algoritmus procedury zatím ještě není rozmyšleno.

Náš přístup je velmi podobný metodě rozhodovacích lesů, např. [1], ve které se uplatňuje *elektivní klasifikace*, tzn. že stromy z rozhodovacího lesa hlasují o výsledné klasifikaci. Bylo by zajímavé zkusit experimentovat s elektivní klasifikací i u stromů vytvořených pomocí ETree.

Reference

1. Breiman L. Random Forrests. Machine Learning 45 (2001), 5-32.
2. Hájek P., Havel I., Chytil M.: The GUHA method of automatic hypotheses determination. Computing 1, 1966, p. 293 – 308
3. Hájek P., Havránek, T.: Mechanising Hypothesis Formation - Mathematical Foundations for a General Theory. Springer-Verlag: Berlin - Heidelberg - New York, 1978.
4. Hettich S., Bay S.D.: The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science, 1999
5. KDNuggets Polls, Data mining/analytic techniques you use frequently. www.kdnuggets.com/polls/2005/data_mining_techniques.htm
6. Kováč M., Kuchař T., Kuzmin A., Ralbovský M.: Ferda, New Visual Environment for Data Mining. Znalosti 2006, Conference on Data Mining, Hradec Králové 2006, p. 118 – 129 (in Czech)
7. Kuchař T.: Experimental GUHA Procedures, Master Thesis, Faculty of Mathematics and Physics, Charles University, Prague 2006 (in Czech)
8. Quinlan J. R.: Induction of Decision Trees. Machine Learning 1(1), 1986, 81 - 106
9. Ralbovsý M., Kuchař T.: Using Disjunctions in Association Mining. In: P. Perner (Ed.), Advances in Data Mining - Theoretical Aspects and Applications, LNAI 4597, Springer Verlag, Heidelberg 2007
10. Rauch J.: Logic of Association Rules. In: Applied Inteligence, Vol. 22, Issue 1, p. 9 – 28
11. Rauch J.: Mining for Association Rules in Financial Data. In: Seminar on Data Mining for Decision Support in Marketing. Porto: LIACC, 2001
12. Rauch J., Šimůnek, M.: An Alternative Approach to Mining Association Rules. Lin T Y, Ohsuga S, Liau C J, and Tsumoto S (eds): Foundations of Data Mining and Knowledge Discovery, Springer-Verlag, 2005 p. 219 – 239
13. Van Rijsbergen C.J.: Information Retrieval. Butterworth-Heinemann Newton, MA USA 1979
14. Šimůnek M.: Academic KDD Project LISp-Miner. In Abraham A. et al (eds): Advances in Soft Computing - Intelligent Systems Design and Applications, Springer Verlag 2003