VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

Ing. Pavel Králík

APLIKACE GENETICKÝCH ALGORITMŮ VE STROJOVÉM UČENÍ

GENETIC ALGORITHMS IN MACHINE LEARNING

Teze doktorské disertační práce

PhD Thesis

Obor:	26–15–9 Technická kybernetika
Školitelé:	doc. Ing. Milan Šrutka, CSc. doc. Ing. Ivan Bruha, CSc.
Oponenti:	prof. Ing. Dr. Miroslav Pokorný doc. Ing. Petr Berka, CSc. doc. Ing. Pavel Ošmera, CSc.
Datum obhajoby:	29. června 2001

ISBN 80–214–1913–X
ISSN 1213–4198

Obsah

1 Úvod **5**

2 Současný stav řešené problematiky **5**

3 Cíl disertační práce **6**

4 Zvolené metody zpracování **7**
4.1 Genetické algoritmy 7
4.2 Horolezecký algoritmus 7
4.3 Algoritmus CNx 7
4.4 Preprocesor KEX 8

5 Hlavní výsledky práce **9**
5.1 Reprezentace chromozomu 9
5.2 Reprezentace rozhodovacích pravidel 11
5.2.1 Bitová reprezentace 12
5.2.2 Reprezentace spojitého numerického atributu 13
5.3 Inicializace populace 14
5.4 Fitness funkce 14
5.4.1 Upravené Laplaceovo kritérium 14
5.4.2 Kombinované kritérium 15
5.5 Ukončovací podmínka 15
5.6 Ověření algoritmu 16
5.6.1 Testovací úlohy 16
5.6.2 Diskretizace a fuzzifikace testovacích databází 17
5.6.3 Výsledky testů 18
5.6.4 Zhodnocení dosažených výsledků 18

6 Závěr **21**

Literatura **24**

Seznam publikací autora **26**

Summary **27**

Autorovo curriculum vitae **28**

1 Úvod

Disertační práce se zabývá induktivními algoritmy strojového učení a možnostmi jejich propojení s genetickými algoritmy (GA). Snaží se najít způsob, jak využít genetické algoritmy *přímo* pro hledání množiny rozhodovacích pravidel.

Strojové učení [17, 18] patří k nejstarším a nejprozkoumanějším oblastem umělé inteligence. Cílem strojového učení je vytvořit z databáze popis konceptů (tříd). Samotný induktivní proces se skládá z prohledávání většinou obrovského prostoru možných popisů konceptu [3]. Existuje několik klasických algoritmů provádějících toto hledání, avšak v poslední době nastává potřeba nových robustních metod pro extrakci znalostí z velkých objemů dat. Jako jedna z takových slibných a efektivních metod se jeví genetické algoritmy.

Genetické algoritmy [12, 15] emulují biologickou evoluci. Jejich výkonnost je založena na velkém počtu iterací (generací) a bez dostatečně velkého výpočetního výkonu nedokáží poskytnout žádané výsledky v přijatelném čase. V posledních letech dochází k intenzivnímu výzkumu v této oblasti a aplikaci genetických algoritmů na čím dál větší množství technických i netechnických problémů.

2 Současný stav řešené problematiky

Lidské učení je jednou z nejdůležitějších charakteristik lidské inteligence. Podobně pak, *strojové učení* (*machine learning*) je jednou z nejdůležitějších oblastí *umělé inteligence*. Strojové učení je stará vědní disciplína. Byla jednou ze základních oblastí umělé inteligence při jejím vzniku v padesátých letech a stala se atraktivním tématem výzkumu díky nástupu expertních systémů a aplikací. Vzhledem k tomu je strojové učení jedním ze synonym pro automatizované získávání znalostí.

Učení z příkladů (*learning from examples*) je jednou z nejprozkoumanějších forem učení. Samotný algoritmus byl použit při řešení mnoha úloh, například v lékařské diagnostice, při předpovědi počasí, v rozpoznávání řeči, v chemii, v geologii atd. Nejobvyklejší použití je spojeno se získáváním znalostí v expertních systémech. Učení z příkladů využívá známé empirické pravidlo: pro experta (nebo učitele) je snazší vytvořit množinu dobrých příkladů (objektů) problému než poskytnout obecnou teorii jevu nebo popis konceptu, tzv. *znalostně-inženýrský paradox* (*knowledge-ingineering paradox*) [3].

Objekt (též příklad, fakt, vzorek, formulace, výraz, případ) [10] je jednotka nebo část daného problému. Například, jestliže se dítě učí rozeznávat různé značky automobilů, potom každé auto pro něj znamená jeden objekt. V současnosti existují dva hlavní druhy jazyka pro popis objektů: atributový

popis a strukturální (relační) popis.

Kromě popisného jazyka objektů musí existovat i jazyk pro popis konceptů (hypotéz). Nejvíce používanými nástroji pro popis tříd (konceptů) ve strojovém učení jsou: rozhodovací stromy, disjunkce komplexů a rozhodovací pravidla [3].

Ve chvíli, kdy jsou induktivním systémem vytvořeny hypotézy, lze je okamžitě použít pro klasifikaci. Klasifikátor využívá popis konceptu jako svoje rozhodovací schéma a klasifikuje neznámé objekty přivedené na vstup do jedné ze tříd (konceptů) dané úlohy.

Po skončení fáze učení, kdy systém vytvořil hypotézy z trénovací množiny, přichází *testování*. Existuje několik uznávaných kritérií pro měření *kvality učení*: klasifikační přesnost, srozumitelnost indukovaných popisů konceptu a výpočetní obtížnost [4].

Empirické učení lze charakterizovat jako heuristické prohledávání prostoru popisů konceptu. Potom je učení formulováno jako produkční systém s následujícími prvky: databáze (trénovací množina), znalostní báze (množina inferenčních pravidel), podpůrná znalostní báze (znalost specifická pro danou úlohu) a inferenční mechanismus [21].

3 Cíl disertační práce

Cílem práce je vytvořit genetický induktivní algoritmus pro vytváření množiny rozhodovacích pravidel z databází, tj. genetické algoritmy by měly být použity *přímo* pro hledání rozhodovacích pravidel.

Tento cíl lze rozdělit do několika dílčích úloh:

- Výklad teorie, algoritmů strojového učení, genetických algoritmů a hybridních metod mezi strojovým učením a GA.
- Návrh efektivního kódování chromozomu s možností tvorby pravidel obsahujících konjunkce selektorů, negace a interní disjunkce.
- Návrh fitness funkce pro řízení tvorby rozhodovacích pravidel.
- Návrh zpracování spojitých numerických atributů databází a řešení jejich diskretizace nebo fuzzifikace [17, 18].
- Ověření klasifikační přesnosti nově vzniklého algoritmu a jeho porovnání s jinými algoritmy.
- Vytvoření programátorské knihovny funkcí nové metody.

4 Zvolené metody zpracování

4.1 Genetické algoritmy

Genetické algoritmy (*genetic algorithms*) jsou součástí *evolučních algoritmů* [15], rychle se rozvíjející oblastí umělé inteligence. Jsou inspirované Darwinovou teorií evoluce. Evoluční programování vzniklo v šedesátých letech a následně vznikly genetické algoritmy popsané Johnem Hollandem [13] (1975).

Princip genetických algoritmů spočívá v kopírování a přesouvání bloků genů. Na začátku je vytvořena *populace* N_{GA} jedinců (chromozomů), ve které každý z jedinců obsahuje genetickou informaci konstantní délky, která představuje zakódované řešení problému. Kvalitu řešení pak představuje ohodnocení jednotlivce *fitness funkcí* (účelovou funkcí). Fitness funkce může být zjištěna v podstatě libovolnými prostředky. Nejčastěji bývá vypočtena analyticky, ale je možné ji zjišťovat na základě chování reálné soustavy [25].

4.2 Horolezecký algoritmus

Horolezecký algoritmus (*hill climbing*) není součástí klasických GA a jeho funkci by stejně tak mohly plnit i jiné algoritmy [15].

Horolezecký algoritmus (HC) je stochastický optimalizační mechanismus a z hlediska GA se jedná o *cílenou mutaci*, tedy mutaci s vazbou na hodnotu fitness funkce jednotlivých jedinců (chromozomů) [24].

Samotný algoritmus provádí lokální optimalizaci v okolí bodu a ve většině případů skončí v lokálním extrému. Tím velmi dobře doplňuje GA, které jsou vhodné pro globální optimalizaci, ale mají problémy v závěrečné fázi hledání řešení, tedy lokální optimalizaci. Horolezeckým algoritmem aplikovaným do GA lze podstatně zkrátit celkovou dobu optimalizace. Aplikuje se většinou po mutačním operátoru na vybraný chromozom.

4.3 Algoritmus CNx

Algoritmy rodiny CNx obsahují dvě hlavní procedury pro generování rozhodovacích pravidel z dané trénovací množiny: proceduru, která opakovaně kontroluje hledání pravidel (*top procedure*), a hledací proceduru (*search procedure*), která nalézá nejlepší komplex pro část trénovací množiny [6].

Algoritmus hledá *setříděný* nebo *nesetříděný* seznam [4] rozhodovacích pravidel tvaru

if $Cmplx$ then class is C_r

Komplex $Cmplx$ (konjunkce selektorů) je ohodnocován heuristickou funkcí (definovanou uživatelem), například Laplaceovo kritérium pro třídu C_r [5]

$$Lapl(C_r, Cmplx) = \frac{K_r(Cmplx) + 1}{K(Cmplx) + R} \tag{1}$$

$K_r(Cmplx)$ je počet trénovacích příkladů (objektů) třídy C_r pokrytých komplexem $Cmplx$ (*class-sensitive coverage*),

$K(Cmplx)$ celkový počet příkladů pokrytých komplexem (*overall coverage*),

R počet tříd zahrnutých do dané úlohy,

Čím vyšší je hodnota ohodnocovací funkce, tím je komplex lepší. Dané heuristiky preferují komplexy pokrývající velké množství příkladů jedné třídy a jen několik příkladů ostatních tříd.

4.4 Preprocesor KEX

Systém KEX (*Knowledge EXplorer*) [1] je vyvíjen na VŠE v Praze od poloviny 80-tých let a je určen zejména pro: systematickou analýzu mnoharozměrných kategoriálních dat a získávání báze znalostí v podobě pravidel bez přítomnosti experta.

Systém KEX může sloužit také jako off-line preprocesor [2], neboť obsahuje pro tento účel proceduru pro diskretizaci spojitých numerických atributů. Preprocesor KEX diskretizuje (kategorizuje) každý spojitý numerický atribut A_n odděleně. Základní ideou je vytvořit intervaly pro něž je *aposteriorní* distribuce tříd $P(C_r \mid interval)$ signifikantně rozdílná od apriorní distribuce tříd $P(C_r)$, $r = 1, 2, \ldots, R$ v celé trénovací množině T. To lze jednoduše udělat sloučením hodnot, pro které nejvíce objektů (příkladů) patří do stejné třídy. Výsledkem takovéto procedury je rozhodovací pravidlo tvaru

if $Cond$ then class is C_r

Jeden ze selektorů podmínky $Cond$ je $A_n \in interval$.

Diskretizace spojitých numerických atributů do přesných hranic mezi intervaly často nevyhovuje požadavkům dané úlohy. Vzniklé intervaly přiřadí trénovací objekty z různých tříd do jednoho intervalu, který je tím pádem nekonzistentní. Popisovaná situace se projevuje hlavně blízko hranic intervalů. Je lepší použít *fuzzy* intervaly [17, 18], které eliminují poruchy na hranicích intervalů. Fuzzifikační preprocesor KEX pracuje podobně jako diskretizační procedura. Jsou stanoveny intervaly (horní a dolní hranice intervalu), ale místo spojení „nejistých“ intervalů do jednoho nejslibnějšího jsou vytvořeny fuzzy hranice intervalu (lichoběžníkovou funkcí příslušnosti) mezi „jistými“ intervaly [7].

5 Hlavní výsledky práce

Algoritmus GA-CN4 je spojením induktivního systému CN4 a genetických algoritmů (GA). Metoda obsahuje dvě hlavní procedury pro generování rozhodovacích pravidel z trénovací množiny: proceduru, která opakovaně kontroluje hledání pravidel (*top procedure*), a hledací proceduru (*search procedure*), která pátrá po nejlepších komplexech za použití genetických algoritmů.

Genetický algoritmus hledá *přímo* podmínky rozhodovacích pravidel, a proto se tento přístup nazývá *opravdový* (*genuine*), narozdíl od *externího* přístupu, který používá GA *jen* pro optimalizaci jednoho (nebo více) parametru algoritmu klasického, tedy negenetického.

Autor navrhuje následující algoritmus GA-CN4 popsaný pseudokódem na obrázku 1 (T je trénovací množina).

Cílem optimalizačního procesu je najít podmínku $Cond$ rozhodovacího pravidla ve tvaru (C je majoritní třída pro podmínku $Cond$)

$$\text{if } Cond \text{ then class is } C$$

která po provedení klasifikace zajistí maximální velikost účelové funkce f (fitness funkce)

$$\max f(P_i), \quad i = 1, 2, \ldots, N_{GA}$$

P_i je prvek populace a N_{GA} je velikost populace.

5.1 Reprezentace chromozomu

Existuje množina trénovacích příkladů a je potřeba vytvořit množinu podmínek rozhodovacích pravidel. Příklady i podmínky si lze zjednodušeně představit jako vektory. Je definován vektor označující atributy příkladu (objektu) $\vec{x} = [x_1, x_2, \ldots, x_N]$, který se po doplnění o požadovanou třídu stane trénovacím příkladem (N je počet atributů). Uvažujme případ, že všechny atributy ve vektoru $\vec{x}$ jsou diskrétní, potom diskrétní atribut A_n obsahuje $J(n)$ hodnot $V_1, V_2, \ldots, V_{J(n)}$.

Celou problematiku si vysvětlíme na příkladu s počasím (*weather problem*). Úloha má dvě možné třídy (`class`) {`+, -`}. Dále obsahuje 4 diskrétní atributy: `windy` {`false, true`}, `humidity` {`normal, high`}, `temperature` {`cool, mild, hot`} a `outlook` {`rain, overcast, sunny`}:

```
class:          +,-
windy:          false,true
humidity:       normal,high
temperature:    cool,mild,hot
outlook:        rain,overcast,sunny
```

procedure GA-CN4(T)
Nechť $ListOfRules$ je prázdný seznam.
Until T je **nil do**
1. Nechť $Cond$ je nejlepší podmínka nalezená GA(T).
2. **If** $Cond$ není nulová **then**
Nechť $T^{'} \subseteq T$ jsou příklady pokryté $Cond$.
Nechť T se změní na $T \setminus T^{'}$.
Přidej pravidlo `If` $Cond$ `then class is` C na konec $ListOfRules$, kde C je majoritní třída v $T^{'}$.
else break (ukonči smyčku)
endif

enddo
Přidej pravidlo `If` *true* `then class is` *majoritní třída* na konec $ListOfRules$.
Return $ListOfRules$.

procedure GA(T)
Inicializuj novou populaci.
Until ukončovací podmínka je splněna **do**
1. Vyber objekty turnajovým výběrem.
2. Generuj potomky dvoubodovým křížením.
3. Proveď bitovou mutaci.
4. Aplikuj horolezecký algoritmus.
5. Zkontroluj, zda mají všechny objekty správné atributy (ve správném rozsahu), a ohodnoť objekty.

enddo
Zvol nejlepší objekt (pravidlo).
If tento objekt je statisticky signifikantní **then**
return tento objekt.
else return nil
endif

Obrázek 1: Algoritmus GA-CN4 v pseudokódu.

Úlohu lze přetransformovat z lexikálního vyjádření do numerického tím, že zaměníme jednotlivé hodnoty za čísla $\{0, 1, \ldots, J(n)-1\}$, kde $J(n)$ je počet hodnot n-tého diskrétního atributu. Vznikne nová úloha: třída (`class`) {`0, 1`} a 4 atributy `windy` {`0, 1`}, `humidity` {`0, 1`}, `temperature` {`0, 1, 2`} a `outlook` {`0, 1, 2`}:

```
class:          0,1
windy:          0,1
humidity:       0,1
temperature:    0,1,2
outlook:        0,1,2
```

Nyní není těžké přetransformovat jeden příklad z trénovací množiny

```
-,false,high,hot,sunny;
```

do numerického vyjádření:

```
1,0,1,2,2
```

Posloupnost čísel reprezentuje kromě příkladu i rozhodovací pravidlo následujícího významu:

```
if windy=false &&
   humidity=high &&
   temperature=hot &&
   outlook=sunny then class is -
```

Pro potřeby reprezentace chromozomu nás v tuto chvíli nezajímají třídy výsledných rozhodovacích pravidel. Do chromozomu jsou zakódovány jen atributy a výsledná třída pravidla je určena jako majoritní třída při klasifikaci trénovacích příkladů.

5.2 Reprezentace rozhodovacích pravidel

Autor práce navrhl pět reprezentací rozhodovacích pravidel pro nový algoritmus genetického strojového učení: bytovou, intervalovou, bitovou reprezentaci, reprezentaci maskou a reprezentaci spojitého intervalu. Pro složitost tématiky jsou v tezích disertační práce uvedeny pouze bitová reprezentace a reprezentace spojitého intervalu.

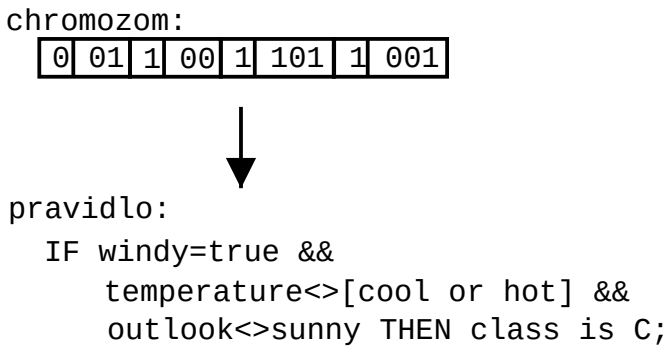
5.2.1 Bitová reprezentace

Reprezentace používá $J(n) + 1$ bitů pro každý atribut A_n. Délka binárního chromozomu k se spočte podle vztahu

$$k = N + \sum_{n=1}^{N} J(n) \tag{2}$$

N je počet atributů a $J(n)$ je počet hodnot n-tého atributu. Pro úlohu `weather` by byla délka chromozomu $k = 4 + 2 + 2 + 3 + 3 = 14$.

Každý atribut začíná jedním bitem (přepínačem) negace (1/0 — negace zapnuta/vypnuta) a pokračuje $J(n)$ bity, kde každý bit představuje jednu hodnotu atributu (1/0 — hodnota atributu je zahrnuta zapnuta/nezahrnuta do podmínky rozhodovacího pravidla). Příklad jednoho takového rozhodovacího pravidla je na obrázku 2.



Obrázek 2: Bitová reprezentace chromozomu.

V příkladu vidíme, že atribut `windy` není negovaný (0 na nejvyšší pozici), bitová kombinace `[01]` znamená, že z hodnot atributu `[false, true]` vstupuje do podmínky rozhodovacího pravidla jen `true` (1 — vstup do podmínky) atd.

Pravděpodobnost výběru nebo nevýběru jakékoli hodnoty atributu A_n do podmínky pravidla je 50%.

Shrňme si vlastnosti bitové reprezentace:

- Implementačně složitější reprezentace.
- Jednoduchá na pochopení.
- Hledání pravidel obsahujících interní disjunkci.

- Možnost negace hodnot atributů a negace interní disjunkce.
- Dobré výsledky při hledání pravidel.
- Pravděpodobnost výběru každé hodnoty atributů je stejná.

5.2.2 Reprezentace spojitého numerického atributu

Hledání rozhodovacích pravidel obsahujících spojité numerické atributy bývá většinou velmi výpočetně náročná a často se stává, že algoritmus nekonverguje. Z tohoto důvodu většina hybridních algoritmů strojového učení tuto vlastnost neobsahuje. Problém reprezentace spojitých hodnot se většinou obchází předzpracováním trénovacích dat preprocesorem, jehož jádro je tvořeno statistickým nebo symbolickým algoritmem. Autor popíše použití takového preprocesoru v jedné z dalších podkapitol.

Nyní přistupme k popisu reprezentace pro zpracování spojitého numerického atributu. Pro spojitý numerický atribut se vytvoří pole o velikosti `2 * sizeof(double)` (datový typ může být také `char` nebo `integer`, závisí na typu spojitého atributu), levá z těchto položek reprezentuje dolní mez $DMez_{A_n}$ n-tého atributu a pravá položka horní mez $HMez_{A_n}$. Matematická reprezentace takto vzniklého chromozomu je

$$DMez_{A_n} < A_n \leq HMez_{A_n} \tag{3}$$

A_n je n-tý atribut. Mohou nastat různé varianty:

- $DMez_{A_n} \geq HMez_{A_n}$, potom atribut jako celek není zahrnut do podmínky.
- $DMez_{A_n} < V_{A_n min}$, potom levá strana vztahu (3) není zahrnuta do podmínky (je zahrnuto jen $A_n \leq HMez_{A_n}$).
- $HMez_{A_n} > V_{A_n max}$, potom pravá strana vztahu (3) není zahrnuta do podmínky (je zahrnuto jen $DMez_{A_n} < A_n$).

$V_{A_n min}$ a $V_{A_n max}$ je minimální respektive maximální hodnota n-tého spojitého atributu. Pokud platí druhá a třetí možnost výčtu naráz, potom atribut jako celek není zahrnut do podmínky.

Pro kombinované úlohy (obsahují jak spojité tak diskrétní atributy) se vytváří kombinovaný chromozom ze spojitých a diskrétních reprezentací, pokud je tato možnost při nastavení parametrů algoritmu povolena.

5.3 Inicializace populace

Na začátku hledání je vytvořena populace N_{GA} chromozomů, které reprezentují jednotlivé podmínky rozhodovacích pravidel. Takto vytvořená populace může být inicializována několika způsoby:

- *Náhodná inicializace pro každé pravidlo* — populace je inicializována pseudonáhodně v každém cyklu (jeden cyklus najde jedno rozhodovací pravidlo) a každý atribut v chromozomu (většinou 8 bitů) získá hodnotu z rozsahu $0 \div 255$.
- *Náhodná inicializace jen pro první pravidlo* — populace je pseudonáhodně inicializována jen v prvním cyklu (hledání prvního pravidla) a v dalších cyklech (při hledání dalších pravidel) se populace již znovu neinicializuje, ale předává se z jednoho cyklu do dalšího nezměněna.
- *Inicializace ze znalostní báze* — kombinuje se pseudonáhodná inicializace se znalostní bází. Existuje-li například znalostní báze 10 rozhodovacích pravidel a populace 100 jedinců (chromozomů), potom atributy 10 jedinců jsou nastaveny na hodnoty atributů rozhodovacích pravidel a zbylých 90 jedinců je inicializováno pseudonáhodně.

5.4 Fitness funkce

Všechny chromozomy (podmínky rozhodovacích pravidel) v každé generaci je potřeba ohodnotit fitness funkcí, tedy stanovit jejich životaschopnost (kvalitu chromozomů). Podmínka $Cmplx$ (konjunkce selektorů) rozhodovacího pravidla ve tvaru:

$$\text{if } Cmplx \text{ then class is } C_r$$

je ohodnocována Laplaceovým kritériem (1) pro třídu C_r. To platí v proceduře řídící hledání rozhodovacích pravidel (*top procedure*).

Pro ohodnocování rozhodovacích pravidel vzniklých v hledací proceduře (*search procedure*) je nutno Laplaceovo kritérium modifikovat, protože se objevuje problém *složitosti* rozhodovacího pravidla.

5.4.1 Upravené Laplaceovo kritérium

Definujme si *rozměr podmínky* $size(Cmplx)$ jako počet všech atributů A_n obsažených v podmínce rozhodovacího pravidla

$$size(Cmplx) = \sum_{n=1}^{N} \kappa(A_n)$$

$Cmplx$ je komplex (konjunkce selektorů), N je počet atributů a $\kappa(A_n)$ je následující funkce:

$$\kappa(A_n) = \begin{cases} 0, & \text{atribut } A_n \text{ neobsažen v pravidle} \\ 1, & \text{atribut } A_n \text{ obsažen v pravidle} \end{cases} \quad (4)$$

Laplaceovo kritérium s vlivem rozměru podmínky pro třídu C_r vypadá následovně

$$LaplSize(C_r, Cmplx) = \frac{Lapl(C_r, Cmplx)}{(1 + size(Cmplx))^{\psi}} \quad (5)$$

K rozměru podmínky $size(Cmplx)$ je přičtena 1 kvůli ohodnocení prázdného předpokladu (`true`), který má rozměr 0.

5.4.2 Kombinované kritérium

Podobné účinky jako upravené Laplaceovo kritérium s vlivem rozměru podmínky rozhodovacího pravidla má *kombinované kritérium*. Autor tímto výrazem nazývá kombinaci různých kvalitativních a kvantitativních veličin. Definujme si vzorec, který je kombinací Laplaceova kritéria pro třídu C_r a kvality

$$CombEval(C_r, Cond) = w_1 Lapl(C_r, Cond) + w_2 Quality(Cond) \quad (6)$$

$w_1, w_2 \in \langle 0, 1 \rangle$ jsou uživatelem definované váhy, $Lapl(C_r, Cond)$ je Laplaceovo kritérium pro třídu C_r a $Quality(Cond)$ je kvalita rozhodovacího pravidla.

Kombinací Laplaceova kritéria a kvality se zvyšuje pravděpodobnost výběru nejlepšího možného pravidla, protože induktivní proces dostává vícekriteriální informaci, která mu podává věrnější obraz použitelnosti rozhodovacího pravidla.

5.5 Ukončovací podmínka

Protože algoritmus GA-CN4 obsahuje dvě hlavní procedury, tj. proceduru, která opakovaně kontroluje hledání pravidel (*top procedure*), a hledací proceduru (*search procedure*), která pátrá po nejlepším komplexu za použití genetických algoritmů, jsou ukončovací podmínky dvě:

1. *Ukončovací podmínka procedury kontrolující hledání pravidel* — top procedura opakovaně volá hledací proceduru tak dlouho, dokud hledací procedura dokáže najít relevantní podmínku pravidla. V opačném případě je procedura kontroly tvorby rozhodovacích pravidel ukončena.

2. *Ukončovací podmínka hledací procedury* — hledací procedura má několik možností, jak ukončit svůj chod:

   - Neexistují žádné další trénovací příklady (kontrolní procedura při nalezení relevantního pravidla hledací procedurou odebere z trénovací množiny pokryté příklady. Hledací procedura oznámí kontrolní proceduře, že nelze najít další pravidlo a fáze učení končí.
   - Nelze najít nové statisticky signifikantní pravidlo, které pokrývá dostatečně velký počet příkladů.
   - Za posledních g iterací (generací) hledací procedury nebyl nalezen jedinec (chromozom) s fitness funkcí větší nebo rovnou nejlepší hodnotě fitness funkce během všech iterací hledací procedury

     $$f_{max} \geq \max f(P(i)), \quad i = t-g, t-g+1, \ldots, t \qquad (7)$$

     t je pořadové číslo iterace od inicializace GA, $\max f(P(i))$ je maximální hodnota fitness funkce v i-té iteraci a f_{max} je nejvyšší fitness funkce během hledání jednoho pravidla.

     Hledací procedura předá nejlepší podmínku rozhodovacího pravidla kontrolní proceduře a ukončí svoji činnost. Kontrolní procedura opět zavolá hledací proceduru, aby našla další relevantní pravidlo. Tímto způsobem algoritmus pokračuje, dokud nedojde k ukončení algoritmu.

5.6 Ověření algoritmu

Pro zjištění klasifikační přesnosti genetického induktivního algoritmu GACN4 bylo použito několik různých databází z UCI a pár medicínských příkladů z univerzit a jiných institucí.

5.6.1 Testovací úlohy

Podívejme se na jednotlivé úlohy, na kterých byly prováděny pokusy.

Japanese Credit — databáze byla získána z referenčního místa UCI [20] (stejně jako následující tři úlohy). Jedná se o úlohu japonské společnosti poskytující půjčky klientům. Cílem úlohy je rozhodnout zda půjčku udělit nebo ne.

Iris — jedná se o úlohu s květinami zvanými kosatec. Úloha byla vytvořena R. A. Fisherem v roce 1998 a jde o jednu z nejznámějších databází, na kterou se literatura často odkazuje. Cílem je rozhodnout na základě rozměrů listů a květů, o jaký druh kosatce se jedná.

Pima Indian Diabetes — databáze vytvořená Národním institutem pro diabetes a nemoci ledvin je v současné době ve správě UCI. Cílem predikce je určit, jestli pacient má nebo nemá cukrovku (diabetes). Pacienty jsou ženy starší 21 let z indiánského kmene Pima.

Australian Credit — úloha představuje podobný problém jako Japanese Credit. Popisných atributů je více, ale cíl je stejný — poskytnutí nebo neposkytnutí úvěru.

Thyreosis — diagnostická úloha nemoci štítné žlázy je získána z Institutu nukleární medicíny v Bernu ve Švýcarsku. Databáze byla používána na katedře matematiky university v Bernu a při testování algoritmu CN4. Kompletní úloha se ve skutečnosti skládá z pěti trénovacích podmnožin objektů. Každá podmnožina se označuje podle počtu atributů. Cílem je určit stav štítné žlázy: normální nebo hyperfunkční (úloha s třemi třídami zahrnuje i hypofunkční štítnou žlázu).

Onco — onkologická databáze byla použita pro testovací účely v Akademii věd ČR v Praze a pro původní algoritmus CN4. Celá úloha zahrnuje dvě podmnožiny označené podle počtu atributů. Cílem úlohy pro tři třídy je určit, jestli je vyšetřovaný pacient zdravý, má lymfatickou uzlinu nebo metastázy (pro 4 třídy je přidána ještě jedna třída).

5.6.2 Diskretizace a fuzzifikace testovacích databází

Jednotlivé databáze byly zpracovány preprocesorem KEX a jejich spojité atributy byly diskretizovány a fuzzifikovány. Tímto způsobem vznikly z každé úlohy (databáze) dvě nové. V dalším výkladu se označují: originální, diskretizovaná a fuzzifikovaná úloha (databáze).

Diskretizace a fuzzifikace má některé vedlejší efekty:

- Jestliže je pro spojitý numerický atribut vygenerován preprocesorem jen jeden interval, znamená to, že atribut je irelevantní a proto může být vypuštěn ze seznamu atributů.
- Po provedení diskretizace může transformovaná databáze obsahovat nekonzistentní objekty (příklady). Takové objekty jsou eliminovány a celkový počet objektů klesne.
- Maximální možná klasifikační přesnost založená na spojitých numerických atributech se může v důsledku diskretizace snížit. Kategorizací nebo fuzzikací se ztratí 4% maximální možné klasifikační přesnosti atributu A_{10} u databáze `Japanese Credit`.

5.6.3 Výsledky testů

Algoritmus byl testován na počítači s procesorem P200 MHz, 32 MB RAM a operačním systémem Linux (2.2.16). Program byl překompilován 32 bitovým překladačem *gcc*. Druhou konfigurací byl školní superpočítač *Jumbo* s operačním systémem Irix od firmy SGI a 64 bitový překladač *CC*.

Algoritmus byl srovnáván se třemi dalšími induktivními metodami: KEX [1] v induktivním režimu, CN4 [16] (setříděný seznam rozhodovacích pravidel) a C4.5 [23]. Byla použita originální, diskretizovaná a fuzzifikovaná data. Je potřeba zdůraznit:

- Originální data nebyla přímo použita v algoritmech GA-CN4 a KEX, protože metody nemají interní diskretizační procedury, respektive GA-CN4 ji obsahuje, ale je výpočetně velmi náročná.
- Ostatní algoritmy (CN4 a C4.5) používají svoji vlastní diskretizační proceduru při zpracování spojitých numerických atributů u originální databáze.
- C4.5 nemá mechanismus pro zpracování fragmentovaných objektů a proto nebyl použit na fuzzifikovaných datech.

Všechny algoritmy byly spuštěny se 4 krát různě nastavenými parametry, dbalo se však na to, aby algoritmus GA-CN4 byl nastaven „podobně“ jako algoritmus CN4.

Každá z 33 testovaných databází byla náhodně rozdělena na dvě podmnožiny (trénovací a testovací příklady v poměru 70% ku 30%), jednotlivé algoritmy vytvořily množinu rozhodovacích pravidel z trénovacích příkladů a následně provedly klasifikaci testovacích příkladů (objektů). Tento postup se opakoval celkem 15 krát.

Výsledky pokusů jsou zobrazeny v tabulce 1, která shrnuje průměrnou klasifikační přesnost [%] u testovací množiny pro 60 generování (15 krát rozdělení testovacích databází a 4 různá nastavení).

Maximální klasifikační přesnost v originálním souboru je vždy 100%, takže ztráta například 30% při diskretizaci dat `onco8` znamená, že maximální možná přesnost je jen 70%. Uvedená hodnota koresponduje s výsledky klasifikace v tabulce 1.

5.6.4 Zhodnocení dosažených výsledků

Při testování byly zjištěny následující poznatky:

- Chování klasických induktivních algoritmů (CN4 a C4.5) v součinnosti s jejich interními procedurami pro zpracování spojitých numerických

databáze	GA-CN4	KEX	CN4	C4.5
Japanese (originální)	–	–	97	83
Japanese (diskretizovaný)	97	97	95	76
Japanese (fuzzifikovaný)	96	97	94	–
Iris (originální)	–	–	99	98
Iris (diskretizovaný)	98	95	98	96
Iris (fuzzifikovaný)	96	96	97	–
Indian (originální)	–	–	96	90
Indian (diskretizovaný)	86	85	84	75
Indian (fuzzifikovaný)	87	86	88	–
Australian (originální)	–	–	100	91
Australian (diskretizovaný)	92	87	92	89
Australian (fuzzifikovaný)	90	89	89	–
Thyr21 (originální)	–	–	97	92
Thyr21 (diskretizovaný)	97	90	97	93
Thyr21 (fuzzifikovaný)	92	91	95	–
Thyr5 (originální)	–	–	91	91
Thyr5 (diskretizovaný)	90	89	89	90
Thyr5 (fuzzifikovaný)	91	89	90	–
Thyr7 (originální)	–	–	96	89
Thyr7 (diskretizovaný)	95	95	93	85
Thyr7 (fuzzifikovaný)	96	96	90	–
Thyr14 (originální)	–	–	93	75
Thyr14 (diskretizovaný)	84	73	85	72
Thyr14 (fuzzifikovaný)	86	69	78	–
Thyr15 (originální)	–	–	96	83
Thyr15 (diskretizovaný)	86	86	81	81
Thyr15 (fuzzifikovaný)	86	87	82	–
Onco7 (originální)	–	–	98	76
Onco7 (diskretizovaný)	77	69	77	73
Onco7 (fuzzifikovaný)	67	61	64	–
Onco8 (originální)	–	–	93	75
Onco8 (diskretizovaný)	66	57	66	69
Onco8 (fuzzifikovaný)	62	60	57	–

Tabulka 1: Klasifikační přesnost [%] testovacích úloh.

atributů je na hladině významnosti 0,05 lepší než chod s předzpracovanými diskretizovanými daty. Hypotéza byla testována t-testem [14]. Tento jev má dvě příčiny:

  - interní procedury byly vyvinuty přímo pro potřeby těchto induktivních algoritmů a
  - interní zpracování (on-line) diskretizace reflektuje aktuální distribuci hodnot atributů, které se mění s vytvořením každého nového pravidla.

  Na druhou stranu je možno vypozorovat, že rozdíly v přesnosti dosažené algoritmy na originálních a diskretizovaných datech byly v relaci s rozdíly maximální možné přesnosti, tj. s ohodnocením diskretizace.

- Původní předpoklad byl, že fuzzifikace by mohla mít lepší výsledky než diskretizace. Výsledky ale ukazují, že žádná z procedur není v tomto směru lepší než druhá.

Primárním cílem práce ovšem byl návrh nového induktivního algoritmu, jehož hledací procedura by zahrnovala úlohově nezávislé genetické algoritmy, které by pracovaly v součinnosti s diskretizačním/fuzzifikačním preprocesorem zpracovávajícím spojité numerické atributy vstupní databáze. Jako výsledek této snahy lze uvést:

- Použitím t-testu pro párové hodnoty[1] na hladině významnosti 0,05 bylo zjištěno, že genetický induktivní algoritmus dosahuje lepších výsledků než ostatní testované induktivní algoritmy. To lze vysvětlit faktem, že klasické algoritmy zkoumají malý počet hypotéz při jednom hledacím cyklu, zatímco genetické algoritmy paralelně zpracovávají robustní populaci možných řešení.

  Výpočty t-testu pro párové hodnoty na hladině významnosti 0,01 ukázaly, že algoritmus GA-CN4 je lepší než metody KEX a CN4, při srovnání s algoritmem C4.5 nelze tuto hypotézu dokázat ani vyvrátit.

  Výsledky t-testu pro dva výběry ukázaly, že jak na hladině významnosti 0,01 tak 0,05 nelze dokázat, že by byl GA-CN4 lepší nebo horší než ostatní algoritmy. Zatímco t-test pro párové hodnoty bere v úvahu fakt, že jenom algoritmus CN4 zpracovával všechny databáze a ostatní algoritmy jen některé, t-test pro dva výběry pracuje se statistickými hodnotami souborů jako celkem. Při tomto druhu posuzování jsou ve výhodě ty algoritmy, které zpracovávaly originální data, protože jejich

[1] V úvahu se berou jen hodnoty se stejnými pořadovými čísly měření, například při srovnání GA-CN4 a C4.5 jsou to jen diskretizované databáze.

maximální klasifikační přesnost je vyšší než diskretizovaných a fuzzifikovaných dat.

- Velkou nevýhodou všech genetických algoritmů je časová a výpočetní náročnost. Genetický algoritmus GA-CN4 je několikanásobně pomalejší než klasické algoritmy. Genetický algoritmus pracuje v rozmezí několika sekund pro jednodušší úlohy až několika minut pro složitější úlohy.

- Algoritmus GA-CN4 je rychlejší ve srovnání s jemu podobnými algoritmy, například systém REGAL [11] vyžaduje systém CM5 se 64 procesory a jeho chod trvá hodiny až dny.

6 Závěr

Předkládaná práce se zabývá problematikou strojového učení a genetických algoritmů a možnostmi jejich uplatnění při vytváření množiny rozhodovacích pravidel. Cílem disertační práce je spojit jmenované metody do nového stochastického induktivního algoritmu, který by vycházel z některé metody klasického strojového učení. Výsledný algoritmus GA-CN4 používá postupy metody CN4 [16], kterou nově definuje a od základů mění.

V teoretické části práce jsou rozebrány jednotlivé teoretické aspekty oboru strojového učení, jsou zmíněny druhy učení a zvýšená pozornost je věnována učení z příkladů. Jsou popsány druhy reprezentace objektů a znalostí. Je nastíněna problematika získávání znalostí a jejich testování. Pozornost je věnována jednotlivým algoritmům strojového učení, které se v současnosti používají se zaměřením na analýzu metody CN4. Následuje popis induktivního systému KEX [1] umožňujícího diskretizaci nebo fuzzifikaci spojitých numerických atributů. Teoretická část práce je ukončena popisem genetických algoritmů a metod genetického induktivního učení, tj. hybridních metod strojového učení a genetických algoritmů.

V praktické části práce autor popisuje nově vytvořený originální systém GA-CN4. Genetické algoritmy jsou v této hybridní metodě použity *přímo* v proceduře pro hledání pravidel. Metoda je schopna zpracovávat diskrétní i spojité atributy nebo v symbióze s off-line preprocesorem KEX spojité numerické atributy diskretizovat/fuzzifikovat.

U metody je použita reprezentace znalostí rozhodovacími pravidly. Autor navrhl pět druhů kódování pravidel do chromozomů, matematicky je definoval a popsal jejich výhody a nevýhody. Každé z kódování prakticky implementoval a ověřil. Nejzdařilejší z těchto kódování mají některé unikátní vlastnosti. Kromě toho, že umožňují tvorbu komplexů (konjunkce selektorů), mohou tvořit také negace selektorů a interní disjunkce selektorů. Původní algoritmus CN4 tuto vlastnost nemá a většina klasických metod strojového

učení také ne. Autor v průběhu pokusů s jednotlivými reprezentacemi zjistil, že se liší v rychlosti tvorby rozhodovacích pravidel. Intervalová a bitová reprezentace rozhodovacích pravidel podávají nejlepší a plně srovnatelné výsledky. Bytová reprezentace je brána spíše jako ilustrativní pro uvedení do problematiky návrhu reprezentace. Reprezentace maskou má dobré výsledky, často srovnatelné s nejlepšími dvěma druhy kódování, ale je poměrně pracné ji dobře nastavit. Autor navrhl reprezentaci spojitých numerických atributů. Zpracování spojitých atributů se v hybridních metodách se prakticky nevyskytuje a v klasických algoritmech je obstarávají speciální procedury založené na statistice. Tento druh reprezentace je výpočetně velmi náročný, a proto je do nového algoritmu zařazen i klasický preprocesor. Propojením metody s preprocesorem KEX získal algoritmus schopnost zpracovávat spojité numerické atributy diskretizací nebo v posledních letech velice populární fuzzifikací. Fuzzifikací se sice ztrácí část informace, ale zato je velmi blízká lidskému uvažování.

Autor navrhl modifikovanou fitness funkci pro ohodnocování rozhodovacích pravidel vytvořených genetickými algoritmy na základě Laplaceova kritéria, matematicky ji popsal a doložil její přednosti při indukci optimální množiny rozhodovacích pravidel. Zdůraznil nutnost zahrnout do fitness funkce penalizaci rozměru pravidla (počet selektorů). Navrhl i variantu multikriteriální fitness funkce zahrnující Laplaceovo kritérium a kvalitu pravidla. Oba druhy fitness funkce dávají lepší výsledky než klasické Laplaceovo kritérium.

Autor navrhl a popsal několik druhů inicializace populace chromozomů, mimo jiné i heuristickou inicializaci ze znalostní báze.

V rámci testování vlastností algoritmu autor sestavil knihovnu funkcí nové metody v jazyce *C++*, kterou spojil s programem CN4 vyvíjeným na McMaster University v Hamiltonu (Kanada).

Autor porovnal vytvořený algoritmus se třemi klasickými metodami strojového učení a t-testem dokázal, že klasifikační přesnost algoritmu GA-CN4 na datech zpracovaných preprocesorem je lepší než chování zbývajících tří metod.

Použití systému GA-CN4 je variabilní, avšak oblastí, kde by mohl být asi nejvíce používán, je získávání znalostí pro expertní systémy a testování těchto znalostí. Není nutné zdůrazňovat, že při současném rychlém rozvoji vědních disciplín, používajících expertní systémy, je snadno pochopitelná znalost vyjádřená rozhodovacími pravidly velice důležitá. Algoritmus je schopen pracovat s úlohami obsahujícími stovky příkladů (například 768), desítky atributů (například 21), fuzzifikované hodnoty, diskrétní i spojité atributy a najít množinu rozhodovacích pravidel s velmi dobrou klasifikační přesností.

V předkládané práci byly naplněny hlavní stanovené cíle. Práce představuje další krok ke zkvalitnění procesu získávání znalostí z příkladů. Dosažené výsledky byly prezentovány jak na konferencích zabývajících se genetickými

a evolučními algorimy (MENDEL) tak klasickými algoritmy strojového učení (ACAI, AIA a SOCO). Výsledky budou v dohledné době publikovány v časopise Intelligent Data Analysis, Kanada.

Literatura

[1] P. Berka, Systém KEX pro strojové učení, *Dobývání znalostí z databází DZD'2000*, Praha, 2000.

[2] P. Berka, I. Bruha, Empirical Comparison of Various Discretization Procedures, *Int. J. of Pattern Recognition and Artificial Intelligence*, Vol. 12, No. 7, 1998, p. 1017-1032.

[3] I. Bruha, Machine Learning: Empirical Methods, *SOFSEM'91*, Jasná pod Chopkom, Slovensko, 1991.

[4] I. Bruha, Quality of Decision Rules: Definitions and Classification Schemes for Multiple Rules, *Machine Learning and Statistics*, The Interface, 1997.

[5] I. Bruha, F. Franek, Comparison of Various Routines for Unknown Attribute Value Processing: The Covering Paradigm, *International Journal of Pattern Recognition and Artifical Intelligence*, Vol. 10, No. 8, 1996, p. 939-955.

[6] I. Bruha, S. Kočková, A Support for Decision Making: Cost-Sensitive Learning System, *Artificial Intelligence in Medicine 6*, 1994, p. 67-82.

[7] I. Bruha, P. Králík, P. Berka, Genetic Learner: Discretization and Fuzzification of Numerical Attributes, *Intelligent Data Analysis*, Canada, 2001, (in press).

[8] P. Clark, R. Boswell, Rule Induction with CN2: Some Recent Improvements, *EWSL'91*, Porto, 1991, p. 151-163.

[9] P. Clark, T. Niblett, The CN2 Induction Algorithm, *Machine Learning 3*, 1989, p. 261-283.

[10] J. G. Carbonell, R. S. Michalski, T. M. Mitchel, *An Overview of Machine Learning*, In [19].

[11] A. Giordana, L. Saitta, REGAL: An Integrated System for Learning Relations using Genetic Algorithms, *Proc. 2nd Int'l Workshop Multistrategy Learning*, 1993, p. 234-249.

[12] D. E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, 1989.

[13] J. H. Holland, *Adaption in Natural and Artifical Systems*, Cambridge, MA: MIT Press, 1975.

[14] Z. Karpíšek, M. Drdla, *Statistické metody*, PC DIR, Brno, 1997.

[15] V. Kvasnička, J. Pospíchal, P. Tiňo, *Evolučné algoritmy*, STU, Bratislava, 2000.

[16] S. Kočková, I. Bruha, CN4: An Extension of Covering Algorithm CN2, AS CR, Technical report, 1993.

[17] V. Mařík, O. Štěpánková, J. Lažanský, *Umělá inteligence*, 1. díl, Academia, Praha, 1993.

[18] V. Mařík, O. Štěpánková, J. Lažanský, *Umělá inteligence*, 2. díl, Academia, Praha, 1997.

[19] R. S. Michalski, J. G. Carbonell, T. M. Mitchel (eds.), *Machine Learning: An Artificial Inteligence Approach*, Tioga Publ., 1983.

[20] P. M. Murphy, D. W. Aha, UCI Repository of Machine Learning Databases, Irvine, University of California, Dept. of Information and Computer Science, `http://www.ics.uci.edu/~mlearn/`.

[21] N. J. Nillson, *Principles of Artificial Intelligence*, Tioga Publ., 1980.

[22] J. R. Quinlan, Induction of Decision Trees, *Machine Learning 1*, 1986, p. 81-106.

[23] J. R. Quinlan, *C4.5 Programs for Machine Learning*, Morgan Kaufmann, 1992.

[24] I. Šimoník, Použití genetických algoritmů pro fuzzy nelineární regresi, Diplomová práce, VUT FS, Brno, 1996.

[25] I. Šimoník, P. Ošmera, M. Pokorný, Z. Drobisz, Using the Genetic Algorithms For Fuzzy Nonlinear Regression, *2nd International Conference on Genetic Algorithms Mendel'96*, VUT FS, Brno, 1996, p. 160-163.

Seznam publikací autora

[1] I. Bruha, P. Králík, Genetic Learner GA-CN4: Some New Enhancements, (in progress).

[2] I. Bruha, P. Králík, P. Berka, Genetic Learner: Discretization and Fuzzification of Numerical Attributes, *Intelligent Data Analysis*, Canada, 2001, (in press).

[3] I. Bruha, P. Králík, Embedding Genetic Algorithm in Attribute-Based Rule-Inducing Learning, In *Intelligent Industrial Automation AIA'99 and Soft Computing SOCO'99* eds. R. Parenti, F. Masulli, Genova, Italy, 1999, p. 631-635.

[4] P. Králík, I. Bruha, Discretizing Numerical Attributes in A Genetic Attribute-Based Learning Algorithm, In *Workshop W06 Advanced Course on Artificial Intelligence ACAI'99*, Chania, Crete, Grece, 1999, p. 48-53.

[5] P. Králík, I. Bruha, Genetic Learning System, *5th International Conference on Soft Computing Mendel'99*, VUT FS, Brno, 1999, p. 55-58.

[6] P. Králík, Aplikace genetických algoritmů ve strojovém učení, Teze disertační práce, VUT FS, Brno, 1999.

[7] P. Králík, I. Bruha, Genetic Learner: Attribute-Based Rule-Inducing Approach, *4th International Conference on Soft Computing Mendel'98*, VUT FS, Brno, 1998, p. 45-51.

[8] P. Králík, Neural Network for Stock Price Prediction, *3rd International Conference on Soft Computing Mendel'97*, VUT FS, Brno, 1997, p. 338-341.

[9] P. Králík, HTML dokumenty s WinTextem602, *Win 5*, 1997.

[10] P. Králík, Predikce časových řad pomocí neuronové sítě, Diplomová práce, VUT FS, Brno, 1996.

K disertační práci se vztahují publikace [1, 2, 3, 4, 5, 6, 7].

Summary

A commonly used productive tool for data mining is machine learning [17, 18] (ML). Given a database, a machine learning algorithm induces concept (class) descriptions, which are involved in a given problem area. The induction itself consists in searching usually a huge space of possible concept descriptions. There exist several paradigms how to control this search, e.g. various statistical methods, logical/symbolic algorithms, neural nets, and the like.

It is worth mentioning that there is no optimal algorithm, which would be able to process correctly any database. One of the promising and efficient paradigms is a genetic algorithm [12] (GA). They emulate to a certain extent biological evolution. They are utilized for optimization processes, similarly to simulated annealing and neural nets.

There have been done many research projects of incorporating genetic algorithms into the field of machine learning. This work describes an efficient application of a GA in the attribute-based rule-inducing learning algorithm. Actually, a domain-independent GA has been integrated into the covering learning algorithm CN4 [16], a large extension of the well-known algorithm CN2 [8, 9]; the induction procedure of CN4 (beam search methodology) has been removed and the GA has been implanted into this shell.

Genetic algorithms are capable of processing symbolic attributes in a simple, natural manner. The processing of numerical (continuous) attributes by genetic algorithms is not so straightforward. One feasible strategy is to discretize numerical attributes before a generic algorithm is called. There exist quite a few discretization preprocessors in data mining and machine learning. This work describes a newer preprocessor for discretization (categorization) of numerical attributes. It was originally implemented for the KEX [1] (Knowledge EXplorer) learning algorithm as its preprocessing (off-line) procedure.

The genuine discretization procedures generate sharp bounds (thresholds) between intervals. It may result in capturing training objects from various classes (concepts) into one interval, which will not be pure; this in particular happens near the interval borders. One feasible way how to eliminate such an impurity around the interval borders is to fuzzify them.

This work firstly introduces the methodology of the new learning algorithm, the genetic learner. Then the discretization/fuzzification preprocessor is presented. Finally, the work compares the entire system (preprocessor and genetic learner) with well-known covering as well as TDIDT [22] learning algorithms.

Autorovo curriculum vitae

Osobní data

Jméno a příjmení:	Ing. Pavel Králík
Trvalý pobyt:	Žitná 23, 62100, Brno
Telefon:	05/4114 5413, mobil 0606/327 546
E-mail:	kralik@cis.vutbr.cz
Narozen:	27. února 1973 v Brně
Rodinný stav:	svobodný

Vzdělání

1996 - 1999	FS VUT v Brně, Technická 2, Brno technická kybernetika, interní PhD. studium Aplikace genetických algoritmů ve strojovém učení
1991 - 1996	FS VUT v Brně, Technická 2, Brno inženýrská informatika Predikce časových řad pomocí neuronové sítě

Pracovní historie

1999 - současnost	rektorát VUT v Brně, Antonínská 1, Brno, civilní služba — knihovnický systém *i*Dokis

Kurzy a školení

1998	Studijní pobyt na McMaster University, Hamilton, Kanada.
1996	Rakousko-český konverzační kurs.

Jazykové znalosti

angličtina: aktivní znalost
ruština: pasivní znalost

Zájmy a záliby

Lyžování, cyklistika, plavání, turistika, horolezectví.