Utilizando a Abordagem Pittsburgh para Combinação de Classificadores Simbólicos

Flávia Cristina Bernardini*, sob orientação de Maria Carolina Monard

Laboratório de Inteligência Computacional
Instituto de Ciências Matemáticas e Computação
Universidade de São Paulo
Av. do Trabalhador Sancarlense, 400 – Caixa Postal 668
CEP 13560-970 São Carlos, SP, Brasil
{fbernard,mcmonard}@icmc.usp.br

Resumo Em problemas nos quais é necessário extrair conhecimento compreensível por seres humanos, indica-se o uso de algoritmos de aprendizado de máquina supervisionado simbólico. Entretanto, a maioria dos algoritmos de aprendizado simbólico disponíveis não conseguem manipular grandes conjuntos de dados. Uma maneira de tentar solucionar esse problema é induzir vários classificadores e combiná-los em um *ensemble* de classificadores, ou evoluí-los em um único classificador simbólico. Neste trabalho exploramos essa segunda solução, utilizando um algoritmo genético proposto como parte do trabalho de doutorado relacionado. Os resultados experimentais mostram que é possível obter bons classificadores utilizando essa abordagem.

Palavras-chave: Aprendizado de Máquina Simbólico, Algoritmo Genético, Computação Evolutiva.

Nível: Doutorado, a ser defendido até setembro de 2006.

1 Introdução

Em problemas nos quais é necessário extrair conhecimento compreensível por seres humanos a partir de uma base de dados, é indicado o uso de algoritmos de aprendizado de máquina supervisionado e simbólico. A um algoritmo de aprendizado supervisionado é fornecido um conjunto de exemplos rotulados a partir do qual um classificador é induzido, cujo objetivo é rotular corretamente novos exemplos. Tal classificador, quando simbólico, pode ser diretamente interpretado pelo usuário/especialista do domínio. Entretanto, grandes conjuntos de dados não podem ser manipulados pela maioria dos algoritmos de aprendizado simbólico disponíveis. Diversas soluções têm sido propostas para lidar com esse problema. Uma delas é utilizar a abordagem de *ensembles* de classificadores. Nessa abordagem, podem ser utilizados algoritmos de aprendizado para induzir um conjunto de classificadores utilizando diversas amostras da base de dados.

* Trabalho realizado com auxílio da FAPESP, Brasil, Proc. N° 02/06914-5.

Após, é construído um *ensemble* que consiste do conjunto desses classificadores e cujas decisões são combinadas de alguma maneira para classificar novos exemplos. Uma outra solução é utilizar Algoritmos Genéticos — AGs — com a abordagem Pittsburgh, por exemplo, para evoluir esse conjunto de classificadores em um único classificador final. No trabalho de doutorado relacionado a este trabalho, intitulado "Combinação de Classificadores Utilizando Medidas de Avaliação de Regras e Algoritmos Genéticos", são utilizadas essas duas abordagens para extrair conhecimento de conjuntos de dados de grande porte.

O objetivo deste trabalho é descrever o AG proposto para combinação de classificadores simbólicos utilizando a abordagem Pittsburgh, bem como descrever alguns resultados experimentais obtidos com esse AG. Para realizar tais experimentos, foi utilizado um sistema computacional por nós proposto e implementado, o qual também é brevemente descrito neste trabalho.

O restante deste trabalho está organizado como segue: na Seção 2 são descritos alguns trabalhos relacionados à combinação de classificadores simbólicos utilizando AGs; na Seção 3 são apresentados brevemente alguns conceitos necessários para a compreensão deste trabalho bem como a notação utilizada; na Seção 4 é descrito o AG proposto; na Seção 5 são descritos os experimentos realizados utilizando conjuntos de dados naturais e os resultados obtidos; por fim, na Seção 6 encontram-se as conclusões deste trabalho.

2 Trabalhos Relacionados

Para tratar o problema de manipulação de grandes bases de dados em mineração de dados utilizando algoritmos de aprendizado simbólico, podem ser utilizadas diferentes abordagens. Como mencionado, uma das abordagens é a abordagem de *ensembles*, na qual vários classificadores são induzidos sobre diversas amostras da base de dados disponível utilizando-se diferentes (ou eventualmente o mesmo) algoritmos de aprendizado simbólico. Após, as decisões do conjunto de classificadores são combinadas para classificar novos exemplos. Em [1,2], propomos diversos métodos de construção de *ensembles* de classificadores simbólicos, os quais foram implementados e avaliados utilizando diversas bases de dados. A outra abordagem proposta está relacionada com o uso de algoritmos genéticos.

Utilizando AGs, tanto é possível evoluir regras de conhecimento quanto classificadores, com o objetivo de encontrar, respectivamente, a melhor regra ou a melhor hipótese que represente os dados disponíveis. Com essa finalidade, duas abordagens são propostas na literatura [3]: a abordagem Michigan, que evolui um conjunto de regras de conhecimento individuais em uma única regra de classificação; e a abordagem Pittsburgh, que evolui um conjunto de classificadores simbólicos em um único classificador simbólico. Diversos sistemas que utilizam a abordagem Pittsburgh têm sido propostos, tais como o sistema EDRL-MD [4]. Um problema central na abordagem Pittsburgh é a codificação dos indivíduos da população manipulada pelo AG, a qual pode gerar indivíduos sintaticamente longos, o que pode dificultar o uso dos AGs. Neste trabalho propomos uma codificação dos indivíduos que visa minimizar esses problemas.

3 Conceitos e Notação

No problema padrão de aprendizado supervisionado, ao algoritmo de aprendizado é dado um conjunto de exemplos de treinamento S com N exemplos $T_i, i = 1, ..., N$, escolhidos de um domínio $\mathcal{X}$ com uma distribuição D fixa, desconhecida e arbitrária, da forma $\{(\mathbf{x}_1,y_1),...,(\mathbf{x}_N,y_N)\}$ para alguma função desconhecida $y = f(\mathbf{x})$. Os $\mathbf{x}_i$ são tipicamente vetores da forma $(x_{i1},x_{i2},...,x_{iM})$, com valores discretos ou contínuos, onde x_{ij} refere-se ao valor do atributo j, $j = 1, ..., M$, denominado X_j, do exemplo T_i. Os valores y_i referem-se ao valor do atributo Y, freqüentemente denominado classe.

Em problemas de classificação, tratados neste trabalho, o atributo classe Y é discreto, ou seja, $y_i \in \{C_1, C_2, ..., C_{N_{Cl}}\}$. A partir do conjunto de treinamento S, um *classificador* $\mathbf{h}$ é induzido. No caso de aprendizado simbólico proposicional [5], $\mathbf{h}$ pode ser transformado em um conjunto de regras de classificação **if-then**, *i.e.* $\mathbf{h} = \{R_1, R_2, ..., R_{N_R}\}$.

Um *complexo* é uma disjunção de conjunções de testes de atributos da forma X_i *op Valor*, onde X_i é o nome do atributo, *op* é um operador pertencente ao conjunto $\{=, \neq, <, \leq, >, \geq\}$ e *Valor* é um valor válido para o atributo X_i. Uma *regra proposicional* R apresenta a forma **if** B **then** H ou, simbolicamente, $B \rightarrow H$, onde H é a *cabeça*, ou a conclusão da regra R, e B é o *corpo*, ou condição de R. H e B são ambos complexos sem atributos em comum. A *cobertura* de uma regra $R = B \rightarrow H$ é definida como segue: exemplos que satisfazem B (o corpo da regra) compõem o conjunto de cobertura de R; em outras palavras, esses exemplos são cobertos por R. Uma *regra de classificação* assume a forma **if** B **then** *classe* $= C_i$; ou seja, a cabeça H de uma regra de classificação é *classe* $= C_i$, com $C_i \in \{C_1, ..., C_{N_{C_l}}\}$.

Dada uma regra $R = B \rightarrow H$ e um conjunto de exemplos S, uma maneira de avaliar essa regra é utilizando medidas de avaliação de regras [6]. Para o cálculo dessas medidas, são necessárias algumas informações a respeito de cada regra: o número de exemplos em S para os quais H é verdade e B é verdade (hb); o número de exemplos em S para os quais H é falso e B é verdade ($\overline{h}b$); o número de exemplos em S para os quais H é falso e B é falso ($\overline{hb}$); e o número de exemplos em S para os quais H é verdade e B é falso ($h\overline{b}$). Algumas medidas de avaliação de regras frequentemente consideradas são acurácia ($Acc(R) = \frac{hb}{b}$) e acurácia de Laplace ($Lacc(R) = (bh + 1)/(bh + b\overline{h} + N_{Cl})$), dentre outras.

Dado um classificador $\mathbf{h}$, uma maneira de avaliar esse classificador como um todo consiste em coletar informações das decisões tomadas pelo classificador em um conjunto de teste S. Para cada par (C_i, C_j), sendo C_i e C_j pertencente ao conjunto $\{C_1, ..., C_{N_{Cl}}\}$, calcula-se o valor da função $M(C_i, C_j)$. Essa função $M(C_i, C_j)$ é definida pelo número de exemplos em S que são pertencentes a C_i e foram classificados por $\mathbf{h}$ como sendo pertencentes a C_j. A partir de tais valores, diversas medidas de avaliação de hipóteses podem ser utilizadas. Duas medidas frequentemente consideradas para avaliar a performance de uma hipótese $\mathbf{h}$ são a acurácia ($Acc(\mathbf{h})$) e a F_1 ($F_1(\mathbf{h})$). Para calcular a medida F_1, é necessário calcular

as medidas de Precisão ($Prec(\mathbf{h})$) e de Sensibilidade, ou *Recall*[1] ($Recall(\mathbf{h})$). Essas medidas estão definidas na Tabela 1.

Tabela 1. Medidas de Avaliação de Classificadores

$$Acc(\mathbf{h}) = \frac{\sum_{i=1}^{NCl} M(C_i,C_i)}{N}$$

$$Prec(\mathbf{h}) = \frac{\sum_{i=1}^{N_{Cl}} M(C_i,C_i)}{\sum_{i=1}^{N_{Cl}} M(C_i,C_i)+\sum_{j=1}^{N_{Cl}} M(C_i,C_j)}$$

$$Recall(\mathbf{h}) = \frac{\sum_{i=1}^{N_{Cl}} M(C_i,C_i)}{\sum_{i=1}^{N_{Cl}} M(C_i,C_i)+\sum_{j=1}^{N_{Cl}} M(C_j,C_i)}$$

$$F_1(\mathbf{h}) = \frac{2\times Prec(\mathbf{h})\times Recall(\mathbf{h})}{Prec(\mathbf{h})+Recall(\mathbf{h})}$$

Algoritmos Genéticos: Um AG necessita de uma estrutura de dados que codifica uma solução para um problema que se deseja resolver, ou seja, uma representação para um ponto no espaço de busca. Tal estrutura é denominada *genoma* e tal ponto no espaço é denominado *cromossomo*. A união de um cromossomo com sua aptidão é denominada *indivíduo*. Um parâmetro codificado no cromossomo é denominado *gene*. Dados um espaço de busca, uma função de aptidão, a qual determina a aptidão de cada indivíduo, ou seja, quão boa a solução por ele representada é para a resolução de um problema, e um conjunto de operadores genéticos (seleção, *crossover* e mutação), o algoritmo genético clássico segue os seguintes passos [7]:

1. Gerar aleatoriamente uma população de n cromossomos.
2. Para cada cromossomo x da população, calcular a função de aptidão $f(x)$ (população avaliada).
3. Repetir os seguintes passos até que n filhos tenham sido criados:
   (a) Selecionar um par de cromossomos da população corrente. A probabilidade de seleção do par de cromossomos é dada pela função de aptidão.
   (b) Aplicar a operação de *crossover* sobre os cromossomos selecionados, produzindo um único filho, resultante da união de uma parte de um dos cromossomos selecionados com outra parte do outro cromossomo.
   (c) Aplicar a operação de mutação em cada locus do filho criado e colocar o cromossomo resultante na nova população. A operação de mutação tipicamente substitui o valor de um gene por outro valor.
4. Substituir a população corrente pela nova população.
5. Voltar ao passo 2 se o(s) critério(s) de parada não for(em) satisfeito(s).

Podem existir diversos critérios de parada, tais como alcançar um número pré-fixado de gerações ou a função de aptidão permanecer constante. *Geração* é o nome dado a cada iteração do algoritmo descrito. O conjunto completo de iterações é denominado *execução* [7].

[1] A definição das medidas $Recall(\mathbf{h})$ e $Prec(\mathbf{h})$ para classes com mais de dois valores foi baseada na definição utilizada no KDD-Cup de 2005 — `http://www.acm.org/sigs/sigkdd/kdd2005/kddcup.html`.

4 O Algoritmo Genético Proposto

O AG proposto no trabalho de doutorado, relacionado a este trabalho, segue os passos do AG clássico descrito na seção anterior, com algumas modificações, e utiliza a abordagem Pittsburgh. O que varia na nossa proposta são os componentes do AG. A vantagem do algoritmo genético por nós proposto, em relação a outros AGs descritos na literatura, está relacionado com a codificação dos indivíduos e com a população inicial do AG. Propomos utilizar, além das operações de crossover e mutação usuais, duas funções de avaliação e dois critérios de parada. A seguir, são descritos os componentes do AG proposto.

População Inicial, Inicialização do AG e Codificação dos Indivíduos: No AG proposto, para construir os classificadores (indivíduos) iniciais, são selecionadas regras (genes) de classificação de uma base de regras. Assim, cada gene de um indivíduo (classificador) é uma regra com um identificador único, e cada indivíduo é, portanto, uma seqüência (conjunto) de identificadores de regras. Para construir a base de regras mencionada, podem ser utilizadas regras construídas por especialistas do domínio, ou regras pertencentes a classificadores induzidos por algoritmos de aprendizado simbólico, tais como $\mathcal{CN}2$, $\mathcal{C}4.5$ e $\mathcal{C}4.5rules$. Na realidade, esses classificadores podem também ser utilizados como indivíduos iniciais do AG. Tais classificadores, se utilizados, podem facilitar o processo de busca do AG pela melhor solução, já que esses classificadores provavelmente são soluções melhores que as soluções construídas aleatoriamente selecionando regras da base de regras. Todas as regras presentes na base de regras devem estar em uma sintaxe padrão de regras, denominada $\mathcal{PBM}$ [8].

Operadores Genéticos: São dois os operadores genéticos utilizados no AG proposto: *crossover* e mutação. O operador de *crossover* utilizado é o *crossover* assimétrico. Em um *crossover* assimétrico, a cada indivíduo da população corrente é associado aleatoriamente um valor entre 0 e 1. Os indivíduos cujo valor associado é menor que a probabilidade de *crossover*, p_c, são selecionados para a operação de *crossover*. Para cada par de indivíduos "pais" são escolhidas aleatoriamente duas posições, uma em cada indivíduo. Os indivíduos pais são divididos em dois segmentos conforme as posições escolhidas. Por fim, são criados os indivíduos filhos por meio da composição dos segmentos dos indivíduos pai. A operação de mutação é utilizada de maneira usual: para aplicar a operação de mutação, seleciona-se aleatoriamente um gene (regra) de um indivíduo (classificador) e troca-se a regra selecionadada por outra selecionada aleatoriamente da base de regras disponível. A probabilidade do indivíduo ser selecionado é dado pela probabilidade de mutação p_m.

Funções de Avaliação: Para avaliar os indivíduos, deve-se avaliar o comportamento do conjunto de regras (classificador) que o indivíduo representa sobre um conjunto de exemplos de teste. Para avaliar os classificadores, neste trabalho são utilizadas as medidas de acurácia e F_1, definidas na Seção 3, denominadas medidas HQ (*Hypothesis Quality*), para avaliar o desempenho do classificador, ou seja, $HQ_{Acc}(\mathbf{h}) = Acc(\mathbf{h})$ e $HQ_{F1}(\mathbf{h}) = F1(\mathbf{h})$. Ainda, podem ser utilizadas diferentes maneiras para determinar a classificação de um novo exemplo, *i.e.*, dado um exemplo $\mathbf{x}$ e um classificador (indivíduo) $\mathbf{h} = \{R_1, ..., R_{N_R}\}$, podem

ser utilizados diversos métodos para classificar $\mathbf{x}$, dado que os classificadores são simbólicos. Neste trabalho, é utilizado um procedimento, denominado MR, no qual são utilizadas as duas medidas de avaliação de regras $Acc(R)$ e $Lacc(R)$, também definidas na Seção 3, ou seja, MR_{Acc} e MR_{Lacc}. Nesse procedimento MR, sao utilizadas todas as regras que cobrem $\mathbf{x}$ para encontrar sua classificação. Para determinar a classe de $\mathbf{x}$, para cada classe $C_v \in \{C_1, ..., C_{N_{Cl}}\}$ é somado o valor da medida utilizada para as regras que classificam o exemplo na classe em questão. A classe com maior valor total é a classificação de $\mathbf{x}$.

Critério de Parada: São dois os critérios de parada (*SC — Stop Criterion*) utilizados no AG proposto. O primeiro critério de parada, denominado SC_{Max}, é dado pelo número máximo de gerações que devem ser executadas. O segundo critério de parada, denominado SC_{Conv} é definido como segue: dado um número de gerações N_{Gen}, se a função de avaliação do melhor indivíduo não melhorar nas últimas N_{Gen} gerações, o algoritmo pára. O número de gerações de cada um dos critérios de parada são parâmetros do AG. Por *default*, N_{Gen} para o critério de parada SC_{Max} é 10, e N_{Gen} para SC_{Conv} é 5.

Pós-processamento do Indivíduo Resultante: Após o AG fornecer como saída o melhor indivíduo (classificador) simbólico por ele evoluído, o sistema proposto realiza um pós-processamento desse classificador. Isso é realizado da seguinte maneira: dado o conjunto de exemplos de treinamento S_{tr}, antes de iniciar a execução do AG, são retirados 10% dos exemplos desse conjunto S_{tr}, formando assim o conjunto de validação S_{val}, tal que a proporção de exemplos de S_{val} em cada classe é semelhante à do conjunto S_{tr}, sendo $S_{tr} = S_{val} \cup S'_{tr}$. Assim, o AG é executado utilizando o conjunto S'_{tr}. O conjunto S'_{tr} é utilizado a cada iteração do AG para o cálculo da função de avaliação utilizada. Ao final da execução, o melhor indivíduo (classificador) encontrado é pós-processado utilizando todo o conjunto de exemplos de treinamento original S_{tr}. O critério de pós-processamento utilizado consiste em retirar desse classificador todas as regras que não cobrem nenhum exemplo em S_{tr}.

Com o objetivo de avaliar nossa proposta, foi implementado um sistema computacional denominado *Genetic Algorithms for Evolving Rule Sets Environment* (GAERE) [9], integrado ao ambiente computacional DISCOVER. O ambiente DISCOVER tem como principal objetivo integrar e padronizar os diversos projetos desenvolvidos no Laboratório de Inteligência Computacional — LABIC — do ICMC-USP, relacionados com pré-processamento de conjuntos de dados, aquisição automática de conhecimento e avaliação de conhecimento [10]. Na maioria desses projetos, diversas tarefas tais como transformação de dados e formatos, execução de algoritmos, medições, entre outras, devem ser executadas diversas vezes. Assim, muitas ferramentas foram e estão sendo implementadas na linguagem de programação *Perl* no DISCOVER, como bibliotecas de classes, para automatizar parcial ou integralmente algumas dessas tarefas.

As vantagens do ambiente DISCOVER em relação a outros sistemas com objetivos semelhantes estão relacionadas à visão unificada que os formatos baseados em padrões proporcionam ao pesquisador (desenvolvedor) de novos componentes. Em [8] é proposta uma sintaxe padrão para representação de conhecimento de

diversos indutores simbólicos denominada $\mathcal{PBM}$. A biblioteca do DISCOVER que implementa a sintaxe padrão $\mathcal{PBM}$ converte os classificadores induzidos na linguagem de representação de conceitos dos principais algoritmos de AM simbólicos, tais como $\mathcal{CN}2$, $\mathcal{C}4.5$ e $\mathcal{C}4.5rules$, para a sintaxe padrão $\mathcal{PBM}$. Nessa sintaxe padrão, cada regra que compõe o classificador simbólico é reescrita na forma de uma regra de classificação. Já para a representação de dados foi proposta uma sintaxe padrão, denominada DSX — *Discover Dataset Sintax*, a qual permite a utilização da biblioteca de classes DOL [11], para converter os arquivos de dados para a sintaxe utilizada por diversos sistemas de AM simbólico.

5 Experimentos e Resultados

Com o objetivo de avaliar o AG proposto para evoluir classificadores simbólicos, foram realizados diversos experimentos utilizando 4 (quatro) conjuntos de dados disponíveis na UCI [12] — autos, heart, ionosphere e vehicle. Na Tabela 2, são mostradas algumas características desses conjuntos de dados: número de exemplos (# Ex.), número de atributos (contínuos, discretos) (# Atr.) e número de valores no atributo classe (# Classes). Deve ser observado que somente o conjunto de dados autos possui valores desconhecidos, somente o conjunto de dados ionosphere possui um exemplo duplicado, e a distribuição dos dados nas classes é uniforme.

Tabela 2. Sumário das características dos conjuntos de dados utilizados

Conj. Dados	# Ex.	# Atr. (cont.,disc.)	# Classes
autos	205	25 (15,10)	2
heart	270	13 (5,8)	2
vehicle	846	18 (18,0)	4
ionosphere	351	33 (33,0)	2

Diversos experimentos foram realizados utilizando o sistema GAERE, variando a função de avaliação e o critério de parada. Inicialmente, foram induzidos 3 classificadores utilizando os algoritmos de aprendizado simbólico $\mathcal{CN}2$, $\mathcal{C}4.5$ e $\mathcal{C}4.5rules$, denominados $\mathbf{h}_{\mathcal{CN}2}$, $\mathbf{h}_{\mathcal{C}4.5}$ e $\mathbf{h}_{\mathcal{C}4.5rules}$, respectivamente. Após, as regras que compõem esses três classificadores foram utilizados para compor a base de regras do AG.

Para inicializar a população inicial do AG com 15 classificadores (parâmetro *default*), nos experimentos realizados foram utilizados como indivíduos iniciais os 3 classificadores $\mathbf{h}_{\mathcal{CN}2}$, $\mathbf{h}_{\mathcal{C}4.5}$ e $\mathbf{h}_{\mathcal{C}4.5rules}$, mais 12 classificadores construídos, cada um composto de N_R regras distintas selecionadas aleatoriamente da base de regras, onde N_R é a média do número de regras presentes em $\mathbf{h}_{\mathcal{CN}2}$, $\mathbf{h}_{\mathcal{C}4.5}$ e $\mathbf{h}_{\mathcal{C}4.5rules}$.

Na Tabela 3, são mostrados os resultados obtidos nos experimentos. Nessa tabela, na primeira linha encontram-se os erros majoritários (EM) de cada conjunto de dados. Nas linhas 2, 3 e 4, encontram-se as taxas de erro estimadas dos classificadores induzidos utilizando os algoritmos $\mathcal{CN}2$, $\mathcal{C}4.5$ e $\mathcal{C}4.5rules$ respectivamente. Nas quatro linhas seguintes encontram-se as taxas de erro estimadas

para a evolução dos classificadores utilizando o AG proposto com os parâmetros *default*, variando a função de avaliação e utilizando o critério de parada SC_{Max}. Finalmente, nas últimas quatro linhas, encontram-se as taxas de erro estimadas para a evolução dos classificadores utilizando o AG proposto com os parâmetros *default*, variando a função de avaliação e utilizando o critério de parada SC_{Conv}. Ainda nessa tabela, $MR_{Acc}HQ_{Acc}SC_{Max}$ indica que foram utilizados na função de avaliação do AG o método de classificação MR_{Acc}, a medida de desempenho do classificador HQ_{Acc} e o critério de parada é o SC_{Max}. Semelhantemente estão indicadas as outras variações do AG utilizadas. Os números entre parênteses indicam o erro padrão da taxa de erro estimada.

Tabela 3. Resultados obtidos com o AG proposto variando a função de avaliação

	autos	heart	ionosphere	vehicle
EM	44,88%	44,44%	35,90%	74,23%
$\mathcal{CN}2$	10,29% (1,87%)	22,96% (2,46%)	9,09% (1,86%)	39,83% (1,57%)
$\mathcal{C}4.5$	9,31% (1,71%)	29,26% (1,78%)	10,56% (1,76%)	27,31% (1,37%)
$\mathcal{C}4.5rules$	8,71% (1,73%)	20,37% (1,27%)	9,96% (1,48%)	25,77% (0,85%)
$MR_{Acc}HQ_{Acc}SC_{Max}$	8,17% (4,60%) o	13,33% (2,54%) +	6,56% (1,27%) o	20,87% (0,66%) +
$MR_{Acc}HQ_{F1}SC_{Max}$	4,36% (1,95%) +	10,74% (2,10%) +	6,57% (1,21%) o	24,42% (1,51%) o
$MR_{Lacc}HQ_{Acc}SC_{Max}$	5,33% (1,83%) o	11,11% (1,56%) +	6,83% (1,21%) o	21,27% (0,85%) +
$MR_{Lacc}HQ_{F1}SC_{Max}$	3,86% (1,39%) +	12,96% (1,77%) +	7,11% (1,28%) o	24,93% (1,53%) o
$MR_{Acc}HQ_{Acc}SC_{Conv}$	5,81% (2,02%) o	13,70% (1,57%) +	6,23% (1,95%) o	23,06% (1,55%) +
$MR_{Acc}HQ_{F1}SC_{Conv}$	6,24% (2,76%) o	14,89% (3,18%) +	6,24% (1,71%) o	27,53% (1,79%) o
$MRLaccHQAccSC_{Conv}$	8,71% (2,84%) o	12,22% (1,75%) +	6,82% (1,70%) o	27,08% (2,46%) o
$MRLaccHQF1SC_{Conv}$	7,83% (1,98%) o	12,22% (2,07%) +	7,38% (1,74%) o	26,59% (1,40%) o

Para estimar as taxas de erro mostradas na Tabela 3, foi utilizada a técnica de 10-*fold cross-validation*. Para os classificadores iniciais induzidos por $\mathcal{CN}2$, $\mathcal{C}4.5$ e $\mathcal{C}4.5rules$, foi utilizada a maneira usual de execução do 10-*fold cross-validation* [5]. Para calcular a estimativa do erro dos classificadores finais evoluídos pelo AG, também foi utilizado o 10-*fold cross-validation*. Entretanto, essa técnica foi adaptada para o problema de estimar a taxa de erro de classificadores evoluídos pelo AG proposto. Uma descrição detalhada dessa adaptação pode ser encontrada em [9].

Para verificar se o resultado do classificador final evoluído é significativamente melhor (ou pior) que o classificador inicial que apresenta menor taxa de erro (o classificador induzido pelo $\mathcal{CN}2$ para o conjunto de dados Ionosphere e os classificadores induzidos pelo $\mathcal{C}4.5rules$ para os outros conjuntos de dados), foi utilizado o teste t pareado com 95% de confiança. Para cada resultado com o AG na Tabela 3, o símbolo "+" indica que o classificador final é significativamente melhor que o melhor dos classificadores iniciais; e o símbolo "o" indica que não há diferença significativa entre as taxas de erro.

Como pode ser observado na Tabela 3, em todos os casos o classificador evoluído foi significativamente melhor (13 casos) ou não houve diferença significativa em relação ao classificador inicial que apresenta a menor taxa de erro. Para o conjunto de dados heart, o classificador evoluído foi sempre significativamente melhor, com uma taxa de erro muito menor que a taxa de erro do melhor classificador (indivíduo) inicial. Quanto aos critérios de parada SC_{Max} e SC_{Conv}, em alguns casos o primeiro critério apresenta melhores resultados que o segundo, o que não é esperado. Como trabalhos futuros, pretendemos estudar

com maiores detalhes o critério de convergência utilizado — critério SC_{Conv}. Ainda assim, considerando que a população inicial consiste somente de três classificadores induzidos pelos algoritmos $\mathcal{CN}2$, $\mathcal{C}4.5$ e $\mathcal{C}4.5rules$, e as regras desses classificadores são as que compõem mais outros 12 classificadores iniciais, os resultados são considerados bastante animadores.

Quanto a complexidade sintática dos classificadores evoluídos com o AG, foi observado que somente os classificadores evoluídos com o conjunto de dados Heart apresentam menor complexidade sintática que os classificadores induzidos com os algoritmos $\mathcal{CN}2$, $\mathcal{C}4.5$ e $\mathcal{C}4.5rules$. Nos experimentos realizados, quanto maior a proporção entre o número de atributos contínuos e o número de atributos discretos presentes no conjunto de dados — Tabela 2 —, maior é a complexidade sintática do classificador evoluído com esse conjunto de dados em relação aos classificadores iniciais, independentemente da função de avaliação utilizada pelo AG. Como trabalhos futuros, pretendemos estudar com maiores detalhes a veracidade desse fato bem como pretendemos propor outras funções de avaliação que considerem a complexidade sintática dos indivíduos em evolução na execução do AG.

6 Conclusões e Trabalhos Futuros

Neste trabalho foi utilizado um algoritmo genético, proposto como parte do trabalho de doutorado, para evoluir classificadores simbólicos iniciais em um único classificador final. Para compor os indivíduos (classificadores) iniciais bem como para executar operações genéticas, é utilizada uma base de regras, a qual pode ser composta por regras induzidas por algoritmos de aprendizado simbólico ou por regras criadas pelo próprio usuário. As vantagens do AG por nós proposto, em relação a outros AGs encontrados na literatura utilizando a abordagem Pittsburgh, estão na codificação dos indivíduos, pois todas as regras que compõem os indivíduos são escritas em uma sintaxe padrão de regras — a sintaxe $\mathcal{PBM}$ —, e na liberdade do usuário inicializar a base de regras com regras tanto induzidas por algoritmos de aprendizado simbólico quanto por regras por ele criadas. Porém, uma comparação experimental com outros AGs que também utilizam a abordagem Pittsburgh não foi possivel, por não ter acesso aos detalhes de implementação nem as implementações desses outros AGs.

Para o AG utilizado, foram conduzidos diversos experimentos, utilizando 4 (quatro) conjuntos de dados da UCI e diferentes configurações do AG. Com o objetivo de comparar as características do classificador evoluído com o melhor classificador construído usando algoritmos de aprendizado que comportam todo o conjunto de dados, foram escolhidas bases de dados de médio porte. Para compor a base de regras foram utilizadas somente regras que compõem classificadores induzidos com algoritmos de aprendizado simbólico. Os resultados obtidos foram animadores, tendo em vista que o classificador final evoluído obteve menor taxa de erro em relação ao classificador inicial com menor taxa de erro. Atualmente estamos investigando outros critérios de convergência para o algoritmo genético proposto, bem como outras funções de avaliação. Estamos também realizando

os experimentos finais com bases de dados de maior porte, bem como utilizando regras de conhecimento de diversas fontes para compor a base de regras, a fim de incrementar a diversidade dos indivíduos (classificadores) do AG proposto.

Agradecimentos: Gostaríamos de agradecer os comentários dos revisores anônimos utilizados para melhorar este trabalho.

Referências

1. Bernardini, F.C., Monard, M.C., Prati, R.C.: Constructing ensembles of symbolic classifiers. In: Int. Conf. on Hybrid Intelligent Systems — HIS 2005. Volume 1., California: IEEE Computer Society (2005) 315–320
2. Bernardini, F.C., Monard, M.C.: Uma proposta para a construção de ensembles simbólicos que explicam suas decisões. In: Conf. Latinoamericana de Informatica — CLEI 2005. Volume 1. (2005) 151–162
3. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer Verlag (2002)
4. Kwedlo, W., Kretowski, M.: An evolutionary algorithm for cost-sensitive decision rule learning. In: 12th Eur. Conf. on Machine Learning — ECML'2001. LNAI. Volume 2167., Springer Verlag (2001) 288–299
5. Monard, M.C., Baranauskas, J.A.: Indução de Regras e Árvores de Decisão. In: Sistemas Inteligentes: Fundamentos e Aplicações. (2003) 115–140
6. Lavrac, N., Flach, P., Zupan, B.: Rule evaluation measures: a unifying view. In: Proc. 9th Int. Workshop on Inductive Logic Programming. LNAI. Volume 1634., Springer Verlag (1999) 74–185
7. Mitchell, M.: An Introduction to Genetic Algorithms. The MIT Press (1997)
8. Prati, R.C., Baranauskas, J.A., Monard, M.C.: Padronização da sintaxe e informações sobre regras induzidas a partir de algoritmos de aprendizado de máquina simbólico. Revista Eletrônica de Iniciação Científica **2**(3) (2002) `http://www.sbc.org.br/reic/edicoes/2002e3`.
9. Bernardini, F.C., Monard, M.C.: Descrição da arquitetura e do projeto do sistema computacional GAERE para realizar evolução genética de classificadores simbólicos. Technical Report 275, ICMC/USP (2006) `http://www.icmc.usp.br/~biblio/download/RT_275.pdf`.
10. Prati, R.C.: O framework de integração do sistema Discover (2003) Dissertação de Mestrado, ICMC/USP. `http://www.teses.usp.br/teses/disponiveis/55/55134/tde-20082003-152116/`.
11. Batista, G.E.A.P.A., Monard, M.C.: Descrição da arquitetura e do projeto do ambiente computacional Discover Learning Environment - DLE. Technical Report 187, ICMC/USP (2003) `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_187.PDF`.
12. Blake, C., Keogh, E., Merz, C.J.: UCI repository of machine learning databases (1998) `http://www.ics.uci.edu/~mlearn/MLRepository.html`.