

PROGRAMAÇÃO GENÉTICA E DATA MINING PARA DESCOBERTA DE CONHECIMENTO EM MEDICINA.

CELIA C. BOJARCZUK¹, HEITOR S. LOPES², ALEX A. FREITAS³

^{1,2}CEFET-PR / CPGEI, Av. Sete de Setembro, 3165, 80230-901. Curitiba – PR.

³PUC-PR, PPGIA-CCET, Rua Imaculada Conceição, 1155, 80215-901. Curitiba – PR.

celia@cpgei.cefetpr.br, hslopes@cpgei.cefetpr.br, alex@ppgia.pucpr.br

Resumo O objetivo deste trabalho é a descoberta de regras de classificação para diagnosticar certas patologias. As regras são capazes de discriminar 12 patologias diferentes cujo sintoma principal é a dor torácica. Para descobrir tais regras é utilizada a programação genética, bem como alguns conceitos de data mining, enfatizando particularmente a descoberta de conhecimento compreensível.

1 Introdução

Para classificar e diagnosticar alguma patologia, deve-se verificar quais atributos estão associados com aquela doença. Neste trabalho há 189 atributos e 12 doenças diferentes (classes) cuja característica principal é a dor torácica. Os atributos referem-se a características da dor torácica, outros sintomas informadas pelo paciente, sinais observados pelo médico, detalhes da história clínica e resultados de testes de laboratório. As doenças diagnosticadas são: angina estável, angina instável, infarto agudo do miocárdio, dissecação de aorta, tamponamento cardíaco, tromboembolismo pulmonar, pneumotórax, pericardite, úlcera péptica, espasmo/refluxo esofágico, doença de origem músculo-esquelética, doença de origem psicogênica. O objetivo é prever qual dessas doenças um paciente tem, dados os valores dos 189 atributos.

O mesmo problema, diagnóstico de dor torácica, foi abordado previamente por [Lopes 97] usando raciocínio por analogia e algoritmos genéticos paralelos. Naquele trabalho os resultados globais com relação à precisão do prognóstico foram satisfatórios, mas tiveram a desvantagem de não serem compreensíveis pelo usuário (o médico). O problema é que o algoritmo genético evoluiu um conjunto de pesos numéricos para os atributos, e todos os 189 atributos mais os pesos numéricos eram usados para a classificação (o qual foi efetuado por um método semelhante ao do agrupamento pelo vizinho mais próximo). Conseqüentemente, o resultado do sistema (o melhor indivíduo encontrado pelo AG) envolveu todos os atributos e muita informação de baixo nível (os pesos dos atributos), ficando muito difícil para o usuário entender os resultados.

Tem havido um interesse crescente na área de *data mining*, onde a meta é descobrir conhecimento que não só seja correto, mas que também seja compreensível para os usuários [Fayyad 96], [Freitas 98]. Assim, o usuário pode entender os resultados do sistema e combiná-los com seu conhecimento para tomar uma decisão bem informado, em vez de confiar cegamente num sistema com resultados incompreensíveis.

Este trabalho envolve o problema anterior de classificação no contexto de *data mining*. Neste contexto, o conhecimento descoberto é representado freqüentemente na forma de regras de previsão (classificação) SE-ENTÃO. A parte SE da regra contém uma combinação lógica dos atributos e a parte ENTÃO contém a patologia (classe) para um paciente cujos atributos clínicos satisfazem a parte SE da regra.

O uso de programação genética (PG) para descobrir regras de classificação compreensíveis, no espírito de *data mining*, é uma área relativamente pouco explorada. Acredita-se esta seja uma área promissora, devido à eficácia de PG em problemas com espaço de busca muito grande [Koza 92] e sua habilidade para executar uma busca aberta das combinações lógicas dos valores dos atributos previsores.

Ao contrário da maior parte dos algoritmos de *data mining*, PG pode descobrir regras expressas em lógica de primeira-ordem da forma $\langle At_i Op At_j \rangle$, onde At_i e At_j estão predizendo atributos do mesmo tipo de dados, $i \neq j$, e Op é um operador relacional (tipicamente “=”, “≤”, ou “>”). A lógica de primeira ordem tem poder mais expressivo do que a lógica proposicional, permitindo a descoberta de regras da forma $\langle At Op Val \rangle$, onde At_i é um atributo previsor, Op é um operador relacional e Val é um valor que pertence ao domínio do At . Para observar a vantagem da lógica de primeira ordem, note-se que, se os

atributos que são comparados têm valores reais, uma condição da primeira ordem da forma $At_i < At_j$ (por exemplo, $Renda < Despesas$) não pode ser expressa por uma representação de lógica proposicional.

Para resumir, o objetivo deste artigo é investigar a eficácia de PG na descoberta de regras compreensíveis e de alto nível, para o domínio de aplicação de diagnóstico de dor torácica.

Este artigo é organizado como se segue. A seção 2 apresenta um breve resumo de PG. A seção 3 descreve com alguns detalhes o sistema de PG proposto neste artigo. A seção 4 informa e analisa os resultados dos experimentos computacionais. A seção 5 discute trabalhos relacionados. Finalmente, a seção 6 conclui o artigo.

2 Um Breve Resumo de Programação Genética

Programação genética é um poderoso método de busca inspirado na seleção natural. A idéia básica é evoluir uma população de “programas” candidatos à solução de um problema específico. Um programa (um indivíduo da população) é usualmente representado na forma de uma árvore onde os nós internos são funções (operadores) e os nós de folha são símbolos terminais. O conjunto de funções e o conjunto de terminais devem conter símbolos apropriados para o problema designado. Por exemplo, o conjunto de funções pode conter operadores aritméticos, operadores lógicos, funções matemáticas, etc; considerando que o conjunto de terminais possa conter as variáveis (atributos) do problema designado.

Cada indivíduo da população é avaliado com relação à sua habilidade para resolver o problema designado. Esta avaliação é executada por uma função de *fitness*. Então o indivíduo sofre a ação de operadores genéticos como reprodução e *crossover*. O operador de reprodução seleciona indivíduos da população atual proporcionalmente aos seus valores de *fitness*, a fim de que façam parte na próxima geração de indivíduos. O operador de *crossover* substitui uma sub-árvore selecionada aleatoriamente de um indivíduo com uma sub-árvore escolhida aleatoriamente de outro indivíduo.

Após a reprodução e o *crossover* terem sido aplicados de acordo com determinadas probabilidades, a geração de indivíduos recentemente criada é avaliada pela função de *fitness*. Este processo é repetitivo, normalmente para um número fixo de gerações. O resultado da programação genética (a melhor solução encontrada) é o melhor indivíduo produzido ao longo de todas as gerações.

3 O Sistema Proposto de Programação Genética

Nesta seção é descrito nosso sistema de programação genética proposta (PG) para o problema de classificação discutido na introdução.

O conjunto de terminais consiste de 189 atributos binários. O conjunto de funções consiste de três funções lógicas, isto é E, OU e NÃO. Estas funções foram escolhidas devido ao objetivo de descobrir regras compreensíveis de alto-nível, como discutido anteriormente. Conseqüentemente, um indivíduo (programa) consiste de uma combinação dos três operadores lógicos aplicado a alguns atributos. Cada indivíduo codifica a parte SE de uma regra.

Por exemplo, o indivíduo hipotético: (E (OU vertigem febre) rx_ok) corresponderia a uma parte SE com as seguintes condições: SE ((vertigem OU febre) E rx_ok)).

Note-se que a parte ENTÃO da regra (a classe) não é codificada no indivíduo. A razão para isto é o fato de que em uma determinada execução da PG, todos os indivíduos representam regras que predizem a mesma classe, como se segue.

Cada rodada do algoritmo de PG resolve um problema de classificação de duas classes, onde a meta é determinar se o paciente tem ou não uma determinada doença. Então, para gerar regras de classificação para as 12 classes, é preciso executar a PG 12 vezes. Na primeira execução a PG encontraria uma regra que prediz a classe 1; na segunda, para classe 2 e assim por diante. Quando a PG está procurando regras que predizem uma determinada classe, todas as outras classes são agrupadas efetivamente em uma grande classe, que pode ser entendida como significando que o paciente não tem a doença prevista pela regra.

Recorde-se que cada paciente é descrito pelos 189 atributos binários. Então, o sistema simplesmente confere se o conjunto de 189 valores binários satisfaz a parte SE da regra especificada pelo indivíduo. Neste caso, o sistema prediz que o paciente tem a doença correspondente, caso contrário o sistema prediz que o paciente não tem aquela doença.

Como a parte ENTÃO da regra não precisa ser codificada no indivíduo, daqui em diante por questão de simplicidade, cada indivíduo será referido como sendo uma regra. Na próxima subseção será descrita a função de *fitness* que foi usada para avaliar a qualidade das regras associadas com os indivíduos da PG.

3.1 Função de *Fitness*

A função de *fitness* avalia a qualidade de cada regra (indivíduo). Este trabalho usa a função de *fitness* empregada por [Lopes 97]. Antes que seja definida a função de *fitness*, é necessário recordar alguns conceitos básicos da avaliação de regras de classificação. Quando se usa uma regra para classificar um caso desconhecido, dependendo da classe prevista pela regra e da verdadeira classe do paciente (a doença que ele tem), quatro tipos de resultados podem ser observados, como se segue:

- verdadeiro positivo (vp) - a regra prediz que o paciente tem uma determinada doença e o paciente de fato tem aquela doença;
- falso positivo (fp) - a regra prediz que o paciente tem uma determinada doença mas o paciente não a tem;
- verdadeiro negativo (vn) - a regra prediz que o paciente não tem uma determinada doença, e realmente o paciente não a tem;
- falso negativo (fn) - a regra prediz que o paciente não tem uma determinada doença mas o paciente a tem.

Onde *vp*, *fp*, *vn* e *fn* significam respectivamente o número de verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo, observando quando uma regra é usada para classificar um conjunto de casos (pacientes). A função de *fitness* utilizada combina dois índices geralmente usados no domínio médico: a sensibilidade (*S*) e a especificidade (*E*), definidas como se segue:

$$S = vp / (vp + fn) \quad (1)$$

$$E = vn / (vn + fp) \quad (2)$$

Finalmente, a função de *fitness* efetivamente usada é definida como o produto destes dois indicadores, isto é:

$$fitness = S * E \quad (3)$$

Assim, o objetivo da PG é maximizar *S* e *E* ao mesmo tempo. Este é um ponto importante, já que seria relativamente trivial maximizar o valor de um destes dois indicadores, às custas de reduzir significativamente o valor do outro. Além disto, a função de *fitness* definida tem a vantagem de ser simples e devolver um valor significativo, normalizado no intervalo [0..1]. Uma boa análise da função de *fitness* citada (independente de qualquer algoritmo),

como também outras medidas relacionadas com precisão preditiva, pode ser encontrada em [Hand 97].

4 Resultados Computacionais

O sistema proposto de PG foi aplicado para o problema de classificação previamente descrito. Em todas as experiências informadas neste artigo a população inicial foi gerada pelo conhecido método *ramped half-and-half*, que cria um número igual de árvores para valores de profundidade que variam de 2 à profundidade máxima [Koza 92]. Nas experiências realizadas, alguns parâmetros foram alterados nas diversas rodadas do programa e estão descritos a seguir, enquanto que os demais parâmetros usuais da PG foram mantidos conforme o sugerido por [Koza 92].

O conjunto de dados consiste de 138 exemplos (os pacientes) e 189 atributos. Como é usual na literatura de classificação, o conjunto de dados foi dividido aleatoriamente em dois conjuntos, sendo um de treinamento (com 90 exemplos) e um de teste (com 48 exemplos). Foi usado a mesma partição de treinamento/teste empregado por [Lopes 97] para tornar possível a comparação dos resultados.

Note-se que, a princípio, isto pode ser considerado um problema difícil de classificação, pois o número de exemplos pode ser considerado pequeno, levando-se em conta o grande número de atributos [Hand 97].

Para cada classe foi executado a PG três vezes e variando alguns de seus parâmetros. Consequentemente, a PG foi executada $3 \times 12 = 36$ vezes e cada execução foi independente das outras. Mais precisamente, a tabela 1 mostra os parâmetros usados em cada um dos três conjuntos de experimentos.

O resultado de cada execução é a melhor regra (indivíduo) encontrada para aquela execução e a melhor regra foi obtida com o melhor desempenho no conjunto de treinamento. Para cada classe foi selecionada a melhor regra das três regras rodadas nas três experiências da PG.

Portanto, o resultado final das experiências é um conjunto de 12 regras, uma para cada classe. A qualidade destas regras descobertas foi analisada de três maneiras, isto é, pela avaliação da precisão preditiva do conjunto de regras como um todo, a precisão preditiva das regras individuais e a compreensibilidade das regras. Estas análises são discutidas nas próximas três subseções, respectivamente.

Comparamos também a precisão preditiva do conjunto de regras descobertos pelo sistema de PG com a precisão preditiva do conjunto de regras descobertas pelo C5.0, um algoritmo de indução de regras “estado da arte” [Rulequest 99]. O resultado desta comparação é informado na próxima subseção.

4.1 Precisão Preditiva do Conjunto de Regras como um todo

Entre os vários critérios que podem ser usados para avaliar a precisão preditiva de um conjunto de regras descobertas, foi usada a taxa de acerto, que, apesar de suas desvantagens [Hand 97], ainda parece ser a mais usada na literatura de classificação. A taxa de acerto é simplesmente a relação do número de exemplos de teste corretamente classificados sobre o número total de exemplos de teste. O sistema alcançou uma taxa de acerto de 77.08%, isto é., o conjunto de regras descobertas classificou corretamente 37 dos 48 exemplos (conjunto de teste).

O algoritmo de indução de regras C5.0 foi aplicado para o mesmo conjunto de treinamento / teste usado nos experimentos com o sistema de PG. O algoritmo C5.0, com os parâmetros padrões, gerou uma árvore de decisão que é equivalente a 17 regras, as quais conseguiram uma taxa de acerto de 79.2%, isto é., classificou corretamente 38 dos 48 exemplos.

4.2 Precisão Preditiva de Regras Individuais

Informar a precisão preditiva do conjunto de regras como um todo é comum na literatura, entretanto, na prática o usuário analisa ou considera uma regra de cada vez. Assim sendo, faz sentido informar a taxa de acerto de cada regra individual. Recorde-se que isto é viável, neste caso, pois nosso sistema descobriu somente 12 regras (uma para cada classe). Tal análise da precisão preditiva das regras individuais é menos possível em casos onde o algoritmo de *data mining* descobre muitas regras.

A tabela 2 mostra os valores relativos à precisão preditiva de cada uma das 12 regras descobertas. Cada linha desta tabela refere-se à regra de melhor desempenho encontrada nas três execuções da PG para a classe correspondente. Todos os valores informados nesta tabela referem-se ao desempenho das regras descobertas no conjunto de teste.

Várias regras descobertas têm alta *S* e alta *E* o que conduz a um alto valor do produto dado pela equação 3. Em particular, seis regras (as que predizem

as classes 3, 4, 6, 7, 9, 10)¹ têm *S* e *E* (também o produto delas) maior que ou igual a 0,80. Porém, algumas regras não são tão boas. As regras que predizem classes 8 e 11, apesar do alto valor *E*, são possivelmente pouco úteis para o usuário, devido à baixa *S* delas. Isto é condizente com os resultados apresentados por [Lopes 97], pois as classes 8 e 11 realmente são as mais difíceis para prever.

4.3 Compreensibilidade das Regras Descobertas

Como mencionado na Introdução, neste trabalho o interesse não é só na precisão preditiva das regras descobertas, mas também na compreensibilidade das regras - no espírito de *data mining*.

A compreensibilidade das regras é significativamente mais difícil de se medir de um modo objetivo, em comparação com a precisão preditiva, pois envolve um aspecto subjetivo considerável. Em grande parte da literatura, porém, a compreensibilidade das regras é associada com a complexidade sintática. Em geral, quanto menor o número de regras e menor o número de condições, mais compreensível será o conjunto de regras como um todo.

Em relação ao número de regras descobertas, o sistema de PG (por definição) prevê o melhor resultado possível que é exatamente uma regra para cada classe. Esta minimização do número de regras descobertas economiza potencialmente o precioso tempo do usuário humano, que é requerido para analisar somente a melhor regra encontrada para cada classe. Em contraste, alguns algoritmos de *data mining* podem facilmente sobrecarregar o usuário com um grande número de regras descobertas, que dificilmente podem ser consideradas como “conhecimento compreensível”.

A pergunta restante é: o quão pequenas são as regras descobertas pela PG? Algumas das regras (particularmente as que predizem classes 1, 6 e 10) foram relativamente extensas. Porém, a PG encontrou algumas regras pequenas, muito compreensíveis. Em particular, as duas menores regras descobertas pelo sistema foram as seguintes:

- SE (ecg_bvc) E (eco_st) ENTÃO classe 5
- SE (rx_aep) ENTÃO classe 7

¹ Os números de classe referem-se às doenças listadas na Introdução, na ordem como foram citadas.

Na primeira regra, os atributos na parte SE denotam respectivamente “ecg: baixa voltagem dos complexos” e “sinais de tamponamento no ecocardiograma”. A classe 5 representa a doença “tamponamento cardíaco”. Na segunda regra, o atributo na parte SE denota “ar no espaço pleural observado na radiografia do torax” e a classe 7 representa a doença “pneumotórax”.

É importante notar que não foi usada nenhuma medida de parsimônia na função de *fitness*. Conseqüentemente, as duas regras simples acima só foram descobertas devido ao seu alto valor de precisão preditiva - equação (3). De fato, a precisão preditiva das duas regras acima é bastante alta, como pode ser visto na tabela 2.

5 Trabalhos Relacionados

O sistema PADO [Teller 95] é uma variação sofisticada de PG que aprende um algoritmo inteiro de classificação para um tipo de sinal arbitrário (imagens, sons, etc.). A maioria do conjunto de funções usado pelo sistema PADO envolve funções matemáticas e primitivas de baixo nível, de forma que o sistema não descobre regras de classificação de alto-nível.

Marchesi, Stelle e Lopes [Marchesi 97] também usaram PG para resolver um problema de classificação. O trabalho deles também usou características de baixo nível do sinal no conjunto de terminais, e funções matemáticas no conjunto de funções, de forma que eles também não descobriram regras de classificação de alto-nível. Uma característica comum dos dois projetos citados, bem como de outros projetos na classificação de sinais de baixo nível, é que eles não descobrem regras SE-ENTÃO de alto-nível. E o “conhecimento descoberto” é avaliado somente com respeito à precisão preditiva, e não com respeito à compreensibilidade.

Sherrah, Bogner e Bouzerdoum [Sherrah 97] propõem um sistema de PG que executa seleção e construção de atributos (usando funções não lineares). O sistema também executa um tipo de seleção de algoritmos, no sentido de que cada indivíduo tem um de três possíveis algoritmos de classificação associado a ele. Porém, nenhum dos três algoritmos enfoca a descoberta de regras de classificação compreensíveis.

Revisamos agora alguns projetos de PG um pouco mais relacionados ao objetivo de *data mining*, isto é, de descobrir conhecimento compreensível. O sistema PGDT [Nikolaev 97] evolui programas de comprimento variável na forma de árvores de decisão. No espírito do paradigma de árvores de decisão, o

conjunto de funções consiste em prever atributos. Isto difere do sistema aqui proposto onde o conjunto de funções consiste dos operadores lógicos E, OU, NÃO.

Ngan, Wong e Leung [Ngan 98] usaram PG para descobrir regras compreensíveis. A abordagem deles tem a interessante característica de que diferentes regras descobertas podem prever atributos diferentes, contribuindo para uma maior autonomia do sistema. Este é um tipo de generalização da tarefa de classificação, onde todas as regras descobertas preveem o mesmo atributo. Porém, as regras descobertas são avaliadas dentro da estrutura proposta de suporte e confiança para regras de associação [Agrawal 93]. Esta estrutura, ao contrário da classificação, não avalia a precisão preditiva das regras descobertas em um conjunto de teste não visto. Além disso, a abordagem deles está baseada no uso de uma gramática para restringir o espaço de busca e assegurar a correção sintática das regras. Infelizmente, a gramática é inteiramente dependente do domínio de aplicação, i.e. uma gramática deve ser escrita para cada domínio de aplicação.

Martin, Moal e Vrain [Martin 98] usaram PG para descobrir regras de classificação compreensíveis. Assim como a presente abordagem, o sistema deles descobre regras expressas em lógica de primeira ordem. Porém, a característica principal do trabalho deles era integrar PG com sistemas de bancos de dados relacionais que contêm relações múltiplas (tabelas), de tal modo que os indivíduos da PG são associados com consultas em SQL. Para alcançar esta meta eles propuseram um modelo baseado em restrições (levando em conta características de bancos de dados relacionais) para a representação individual e operadores genéticos, os quais não são necessários no presente caso. Outro trabalho cuja característica principal é a integração de PG com sistemas de banco de dados relacionais e usa os indivíduos de PG para consulta em SQL, é discutido em [Freitas 97]. Este trabalho envolve a tarefa de classificação e indução generalizada de regras, mas assume que os dados que estão sendo “minerados” estão contidos em uma única relação.

6 Conclusões e Trabalhos Futuros

Apesar do pequeno número de exemplos disponível no domínio de aplicação explorado neste trabalho (levando em conta o grande número de atributos), os resultados das experiências podem ser consideradas muito promissoras. As regras descobertas tiveram um

bom desempenho relativo à precisão preditiva e considerando o conjunto de regras como um todo e cada regra individual.

Além disto, o que é mais importante do ponto de vista de *data mining*, o sistema descobriu algumas regras bastante compreensíveis, como discutido na seção 4.3. Deve ser enfatizado que este resultado foi obtido mesmo sem incluir na função de *fitness* um termo que penalize regras longas. A compreensibilidade das regras descobertas poderia, a princípio, ser melhorada com uma modificação adequada da função de *fitness*. O quanto a precisão preditiva seria reduzida com tal modificação é uma pergunta cuja resposta requer uma pesquisa adicional. De fato, ainda é um problema em aberto a questão do equilíbrio entre a precisão preditiva e a simplicidade dos programas gerados pela programação genética [Zhang 95].

Outra possibilidade para pesquisa futura seria executar uma seleção de atributos numa fase de pré-processamento para reduzir o número de atributos (e o espaço de busca correspondente) dado à PG. Seleção de atributos é uma área de pesquisa muito ativa em *data mining* [Liu 98].

Finalmente, um trabalho futuro também deveria incluir a aplicação do sistema de PG proposta neste artigo a outros conjuntos de dados, para validação dos resultados reportados neste artigo.

Referências Bibliográficas

[Agrawal 93] Agrawal, R., Imielinski, T. e Swami, A. Mining association rules between sets of items in large databases. Proc. 1993 Int. Conf. Management of Data (SIGMOD-93), 207-216. 1993.

[Fayyad 96] Fayyad, U.M., Piatetsky-Shapiro, G. e Smyth, P. From data mining to knowledge discovery: an overview. In: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (Eds.) Advances in Knowledge Discovery & Data Mining, 1-34. Cambridge: AAAI/MIT, 1996.

[Freitas 97] Freitas, A.A. A genetic programming framework for two data mining tasks: classification and generalized rule induction. Genetic Programming 1997: Proc. 2nd Annual Conf., 96-101. 1997.

[Freitas 98] Freitas, A.A. e Lavington, S.H. Mining Very Large Databases with Parallel Processing. Londo: Kluwer, 1998.

[Hand 97] Hand, D. Construction and Assessment of Classification Rules. New-York: John-Wiley & Sons, 1997.

[Koza 92] Koza, J.R. Genetic Programming: on the Programming of Computers by Means of Natural Selection. Cambridge: MIT Press, 1992.

[Liu 98] Liu, H. e Motoda, H. Feature Selection for Knowledge Discovery and Data Mining. London: Kluwer, 1998.

[Lopes 97] Lopes, H.S., Coutinho, M.S., Heinisch, R., Barreto, J.M. e Lima, W.C. A Knowledge-based System for Decision Support in Chest Pain Diagnosis. Medical & Biological Engineering & Computing, v. 35, suppl., part I, 514, 1997.

[Marchesi 97] Marchesi, B., Stelle, A.L. e Lopes, H.S. Detection of epileptic events using genetic programming. Proc. 19th Int. Conf. IEEE/EMBS, 1198-1201, 1997.

[Martin 98] Martin, L., Moal, F. e Vrain, C. A relational data mining tool based on genetic programming. Principles of Data Mining & Knowledge Discovery: Proc. 2nd European Symp. LNAI 1510, 130-138. 1998.

[Ngan 98] Ngan, P.S., Wong, M.L. e Leung, K.S. Using grammar based genetic programming for data mining of medical knowledge. Genetic Programming 1998: Proc. 3rd Annual Conf., 254-259. 1998.

[Nikolaev 97] Nikolaev, N.I. e Slavov, V. Inductive genetic programming with decision trees. Proc. 1997 European Conf. on Machine Learning (ECML-97). Lecture Notes in Computer Science 1224, Springer-Verlag, 1997.

[Rulequest 99] Rulequest Research. "Data mining and knowledge discovery tools". <<http://www.rulequest.com/>>, may 1999.

[Sherrah 97] Sherrah, J.R., Bogner, R.E. e Bouzerdoum, A. The evolutionary pre-processor: automatic feature extraction. Genetic Programming 1997: Proc. 2nd Annual Conf., 304-312. 1997.

[Teller 95] Teller, A. & Veloso, M.. Program evolution for data mining. Int. J. Expert Systems, 3rd Quarter, 216-236, 1995.

[Zhang 95] Zhang, B.-T. e Mühlenbein, H. Balancing accuracy and parsimony in genetic programming. Evolutionary Computation, v.3, 17-38, 1995.

Tabela 1: Conjunto de parâmetros usados nos experimentos.

exp#	Tamanho da população	Número de gerações	Tamanho máximo da árvore	Profundidade máxima da árvore	Probabilidade de <i>crossover</i>	Probabilidade de reprodução
1	5.000	50	65	12	0,85	0,15
2	3.500	50	65	15	0,75	0,25
3	2.500	30	60	12	0,70	0,30

Tabela 2: Desempenho do sistema nas regras individuais.

classe#	<i>vp</i>	<i>fp</i>	<i>fn</i>	<i>vn</i>	<i>S</i>	<i>E</i>	<i>S x E</i>
1	2	0	2	44	0,50	1,00	0,50
2	1	0	1	46	0,50	1,00	0,50
3	6	0	0	42	1,00	1,00	1,00
4	6	1	0	41	1,00	0,97	0,97
5	3	1	1	43	0,75	0,97	0,73
6	4	0	1	43	0,80	1,00	0,80
7	3	0	0	45	1,00	1,00	1,00
8	0	1	2	45	0,00	0,97	0,00
9	4	2	0	42	1,00	0,95	0,95
10	4	2	0	42	1,00	0,95	0,95
11	1	0	2	45	0,33	1,00	0,33
12	3	0	2	43	0,60	1,00	0,60
média	3,08	0,58	0,91	43,41	0,706	0,984	0,694