

Estudo da ferramenta Keel fazendo uma abordagem de um algoritmo genético-fuzzy de aprendizado

Andre Brito Fonseca
Acadêmico de Ciência da Computação - UNICENTRO/Guarapuava(PR),
andrebritofonseca@gmail.com

Angelita Maria de Ré
Professora Doutora do Departamento de Ciência da Computação - UNICENTRO/Guarapuava(PR),
andre65@yahoo.com.br

Resumo: As soluções geradas a partir de sistemas inteligentes podem ser, muitas vezes, complexas e extensas. Dessa forma, muitos pesquisadores e estudantes, depois do término dos seus respectivos projetos, precisam comparar os resultados das suas pesquisas, utilizando diferentes abordagens. Atualmente, existem muitas ferramentas com Algoritmos Inteligentes já implementados e executando versões de pesquisas desenvolvidas. Este artigo tem como objetivo o estudo da ferramenta KEEL, abordando mais especificamente o algoritmo SLAVE.

Palavras-chave: sistemas inteligentes; KEEL; SLAVE; ferramenta.

Abstract: The solutions generated by intelligent systems can, sometimes, be very complex and extensive. Looking that way, researchers and students, after finishing their current projects, need to compare the results of their work with different approaches. Nowadays, there are many tools with Intelligent Algorithms already implemented and executing versions of researches developed. This paper aims the study of the tool KEEL, approaching specifically the SLAVE algorithm.

Key-words: smart systems; KEEL; SLAVE; tool.

Introdução

Ferramentas que auxiliam o desenvolvimento de sistemas inteligentes podem se tornar úteis quando o tempo para o desenvolvimento de um novo aplicativo é pequeno ou quando os resultados de uma implementação devem ser comparados com outros existentes na literatura. Sendo assim, diversas ferramentas são disponibilizadas para uso, muitas vezes, gratuito. Algumas ferramentas se encontram disponíveis na Internet, sendo de fácil acesso a grande parte das pessoas.

A ferramenta KEEL¹ (*Knowledge Extraction based on Evolutionary Learning*) oferece algoritmos para diversos propósitos. Dentre os principais, é necessário destacar algoritmos de mineração de dados, algoritmos evolucionários que utilizam diferentes abordagens (Pittsburgh, Michigan e IRL (*Iterative Rule Learning*)) e diferentes técnicas evolutivas combinadas com outros algoritmos, “permitindo, assim, uma análise completa de qualquer modelo de aprendizado” (ALCALÁ-FDEZ et al., 2008). Além disso, a ferramenta foi desenvolvida com o intuito de auxiliar a pesquisa e o ensino.

¹ Disponível em <<http://www.keel.es>>

Para que os algoritmos funcionem corretamente, é necessária uma base de dados, contendo diversos casos reais com seus respectivos resultados. Essa base é considerada como uma base de conhecimento, a qual será utilizada para o aprendizado do algoritmo. É necessário ressaltar que a base deve estar dentro dos padrões da ferramenta, ou seja, sua formatação deverá obedecer determinadas regras.

Um dos algoritmos incluídos na ferramenta é o SLAVE (*Structural Learning Algorithm in a Vague Environment*), "um algoritmo de aprendizado intuitivo que utiliza conceitos baseados na teoria dos conjuntos *fuzzy*" (GONZÁLEZ; PÉREZ, 1999). Sendo assim, o SLAVE é um sistema genético-*fuzzy*, que combina Algoritmo Genético com Lógica *Fuzzy*.

Este artigo está dividido em seções: na seção 1, é feita uma introdução contextualizando o objetivo deste estudo; a seção 2 apresenta o embasamento teórico, uma breve introdução sobre Algoritmos Evolucionários e Algoritmos Genéticos, a ferramenta KEEL e suas funções, o sistema genético-*fuzzy* SLAVE, o formato da base de dados e um exemplo de execução. Na seção 3 são apresentados os resultados deste estudo, e na seção 4 são feitos comentários sobre as conclusões e trabalhos futuros.

Algoritmos Evolucionários e Algoritmos Genéticos

Algoritmos Evolucionários são algoritmos de otimização baseados na evolução natural e processos genéticos e são técnicas que resolvem diversos problemas científicos complexos (EIBEN; SMITH, 2003). Existem diversos tipos de Algoritmos Evolucionários: Algoritmos Genéticos, Algoritmos Meméticos, Estratégias Evolutivas, *Simulated Annealing*, Colônia de Formigas, Programação Genética, entre outros. No presente artigo serão enfatizados os de Algoritmos Genéticos.

O campo de aplicações de Algoritmos Evolucionários é extenso. Podem-se destacar o aprendizado de controle em robótica (MUCIENTES et al., 2006), melhorias em sistemas de aprendizado eletrônico (ROMERO et al., 2004), escalonamento de processos (HOU; REN ASARI, 1994) e problemas que envolvem otimização combinatória (NEUMANN et al., 2004).

Quando Algoritmos Genéticos são utilizados, deve-se adaptar o problema à concepção Darwiniana (GOLDBERG, 1989). Dessa forma, indivíduos representam as soluções possíveis para o problema e operadores como cruzamento e mutação são aplicados de modo a manter a diversidade em um conjunto de indivíduos.

Cada indivíduo tem um *fitness*, que representa o quão apto ele é. Segundo Gonçalves (2004), a função de aptidão define o *fitness* do indivíduo e a seleção é a etapa na qual indivíduos serão escolhidos para a próxima geração. Alguns métodos de seleção são o torneio, *ranking*, roleta, roleta viciada, entre outros.

Existem diversos parâmetros em um Algoritmo Genético. Os principais são taxa de cruzamento (a quantidade de indivíduos que efetuarão o cruzamento), taxa de mutação (a quantidade de indivíduos irão realizar a mutação), quantidade de indivíduos e quantidade de gerações. Segundo Goldberg (1989), dependendo da escolha desses parâmetros, o resultado final da execução pode variar. Dessa forma, muitos testes devem ser realizados, de maneira que a melhor combinação dos parâmetros seja encontrada.

Para que um Algoritmo Evolucionário realize o aprendizado de máquina, é necessária a utilização de bases de casos, em que diversos casos em forma de regra estão armazenados. Estas regras possuem antecedentes e um conseqüente, que representam as condições a serem satisfeitas e o resultado, respectivamente (REZENDE, 2005). Para a implementação, existem diversas maneiras de tratar as regras, porém existem diferentes tipos de abordagens já implementadas, como a abordagem Pittsburgh, Michigan e IRL. A escolha da abordagem depende da aplicação e os resultados finais podem variar.

Na abordagem Pittsburgh (SMITH, 1980) indivíduos possuem um conjunto de regras e as operações são aplicadas nestas regras. Dessa forma, cada indivíduo representa uma solução para o problema. Por outro lado, na abordagem Michigan (MICHALEWICZ, 1997), um indivíduo representa uma e apenas uma regra. Sendo assim, a solução para um problema é um conjunto de indivíduos, podendo ser também a população inteira. Segundo Galea et al. (2005), na abordagem IRL, o melhor indivíduo de uma geração, ou seja, o indivíduo, que possui a maior quantidade de regras ativadas, é armazenado em um banco de dados. Assim, a cada geração, um indivíduo é armazenado no banco de dados e no final da execução, os indivíduos que apresentaram os melhores resultados no decorrer da execução são apresentados.

Segundo Alcalá-Fdez et al. (2008), Algoritmos Genéticos são importantes técnicas para aprendizado e extração de conhecimento. Além disso, é interessante utilizar Algoritmo Genético para a extração de dados (*Data Mining*), pois a técnica lança mão de métodos robustos que podem realizar buscas em todo o espaço de soluções, por exemplo, em regras que representam o conhecimento.

Algoritmos Evolucionários podem ser de complexa codificação (ALCALÁ-FDEZ et al., 2008). Sendo assim, pesquisadores e desenvolvedores se reúnem para construir diversos tipos de ferramentas, para auxiliar na execução do algoritmo de forma correta e intuitiva.

Como exemplos de ferramentas, serão elencados alguns exemplos, a saber: KNIME (BERTHOLD et al., 2006), um ambiente de relacionamento entre diversos algoritmos e conjunto de dados; Tanagra (RAKOTOMALALA, 2005), um software de extração de dados com propósitos educacionais e para pesquisas, além de contar com diversos esquemas de aprendizado de máquina e análise experimental; Weka (WITTEN; FRANK, 2005), uma das ferramentas mais completas de extração de dados e aprendizado de máquina, além de possuir diversos pacotes com um amplo campo de abrangência.

KEEL

Segundo seus desenvolvedores, a ferramenta KEEL facilita a análise do comportamento de aprendizado em diferentes áreas, e é necessário fazer o pré-processamento de tarefas, facilitando, assim, o trabalho (ALCALÁ-FDEZ et al., 2008). Dessa forma, a ferramenta foi criada com dois propósitos: pesquisa e ensino. Segundo Alcalá-Fdez et al. (2008), a Keel é útil como uma ferramenta de pesquisa, porque auxilia a automação de experimentos, podendo comparar diferentes algoritmos perante uma mesma base de dados. É necessário ressaltar que aquela oferece mecanismos para mesclar diversos tipos de algoritmos, dados estatísticos e técnicas de Inteligência Artificial. Por outro lado, quando utilizada como uma ferramenta de ensino, um

número menor de algoritmos é disponibilizado e o seu tempo de execução é comparado à utilização da ferramenta com fins de pesquisa.

Os algoritmos presentes na ferramenta podem ser visualizados na seção de algoritmos, no site do aplicativo².

Na sequência serão descritos os principais componentes da ferramenta KEEL.

Inicialização

A interface é amigável e pode ser visualizada na figura 1. Na Keel, existem quatro opções: *Data Management*, *Experiments*, *Educational* e *Help*.

Na *Data Management*, há um conjunto de ferramentas que podem ser utilizadas para criar, exportar, importar e particionar bases de dados no formato que a ferramenta KEEL suporta (ALCALÁ-FDEZ et al., 2008). Na *Experiments* o principal objetivo é o projeto de experimentos, utilizando determinados conjuntos de dados, ou mesmo fornecendo diversas opções de algoritmos como validações, aprendizado por classificação, regressão, aprendizado não-supervisionado, entre outros. A *Educational* é semelhante à *Experiments*, porém fornece informações sobre o processo de aprendizado. A *Help* oferece diversos tópicos de ajuda, esclarecendo possíveis situações de dúvida.

Gerenciamento de Dados (Data Management)

Na seção de gerenciamento de dados, é possível criar, importar, exportar ou particionar uma base de dados. A tela *Data Management* pode ser visualizada na figura 2.

Pode-se adicionar um novo campo e é possível escolher dentre as opções de tipo (real, inteiro ou nominal). Além disso, a ferramenta permite inserir os valores (no caso de real ou inteiro, deve ser inserido os intervalos e se for nominal, devem ser valores únicos) e escolher entre *input* ou *output*, ou seja, se o valor que está sendo adicionado à base é um valor de entrada ou saída (Figura 3).

Figura 1 - Inicialização do KEEL

Fonte: Knowledge Extraction based on Evolutionary Learning: <<http://www.keel.es>>

As opções *Import Data* e *Export Data* servem para importar uma base já existente e exportar uma base criada, respectivamente. Existem diversos formatos para importar e exportar. Primeiramente, deve ser escolhida a base e após isso, são selecionados o formato de origem e destino, inclusive o diretório para onde será importado ou exportado, conforme mostra a figura 4. A conversão é uma restrição da ferramenta, ou seja, essa atividade executa um formato de arquivo específico.

Além dessas opções, é possível particionar a base de dados, preparar os dados e visualizar a base em gráficos. Para particionar a base, utiliza-se o *Partition*. Existem diversas escolhas de particionamento, como *k-fold cross validation* e *5x2 cross validation*. É possível,

² Disponível em <<http://www.keel.es>>

ainda, escolher o número de partições e propriedades da partição escolhida, conforme é apresentado na figura 5.

Experimentos

Na seção de experimentos, é necessário selecionar que tipo de algoritmo será utilizado, importar a base de dados e informar como a base de dados foi particionada. A tela de projetos de experimentos pode ser visualizada na figura 6.

Figura 2 - Gerenciamento de Dados

Fonte: <<http://www.keel.es>>

Figura 3 - Inserção de um valor na base

Fonte: <<http://www.keel.es>>

Após escolher o tipo de algoritmo e o tipo em que a base está particionada, é necessário importar a base. Para isso, verifica-se se a base está particionada, selecionando o diretório em que as partições se encontram; se a base não está particionada, seleciona-se o arquivo com extensão *.dat* criado no gerenciamento de dados, ou seleciona-se um arquivo com extensão *.zip*, representando um pacote de dados. Caso a base esteja particionada, além de selecionar o diretório, é necessário selecionar o tipo de partição. No entanto, caso a base não esteja particionada, o arquivo com extensão *.dat* é selecionado. Após este processo, um diagrama é inicializado com a base. Além disso, é necessário declarar um mnemônico para a base e um nome. A importação de um arquivo não particionado é apresentado na figura 7.

Figura 4 - Importação de dados

Fonte: <<http://www.keel.es>>

Figura 5 - Particionamento da base de dados

Fonte: <<http://www.keel.es>>

Figura 6 - Experimentos

Fonte: <<http://www.keel.es>>

Figura 7 - Importação de um arquivo não particionado

Fonte: <<http://www.keel.es>>

Na sequência, novas opções aparecem. Assim, uma lista de sistemas classificadores é apresentada, possibilitando a seleção dos algoritmos necessários. No exemplo da figura 8, o algoritmo genético-fuzzy SLAVE é selecionado e conectado com uma base hipotética. Para conectar o algoritmo com a base ou com outros algoritmos, utiliza-se o recurso gráfico da seta para indicar o componente de origem e o de destino.

Figura 8 - Um diagrama, onde uma base é conectada com um algoritmo

Fonte: <<http://www.keel.es>>

É importante salientar que através da KEEL é possível acompanhar a execução do sistema. Isto é feito através de informações disponibilizadas na interface. Entretanto, dependendo do algoritmo, uma interface gráfica é apresentada com a execução. Porém, no caso do SLAVE, que foi desenvolvido sem uma interface gráfica, é necessário executar o comando `java -jar RunKeel.jar`, utilizando um terminal de linha de comando.

Algoritmo Genético-Fuzzy de Aprendizado - SLAVE

Este algoritmo utiliza uma abordagem iterativa para aprender regras *fuzzy* (GONZÁLEZ; PÉREZ, 2001). O aprendizado de regras *fuzzy* serve, principalmente, para capturar as decisões de um especialista que não são exatas. Essa característica é uma propriedade da lógica *fuzzy*: lidar com dados incertos. Quando Algoritmos Genéticos são acoplados à Lógica *Fuzzy*, a extração de dados de um conjunto de exemplos se torna um aprendizado, ou seja, os indivíduos do Algoritmo Genético são codificados de maneira que eles guardam informações sobre toda a base de exemplos (GONZÁLEZ; PÉREZ, 1999).

O algoritmo de aprendizado extrai um conjunto de regras *fuzzy* a partir de um conjunto de exemplos. A grande influência do Algoritmo Genético é na escolha da regra que melhor representa a resposta do especialista, ou seja, da pessoa que produziu a base de dados. A cada geração, a regra obtida é incorporada a um conjunto das melhores regras obtidas até o momento. Sendo assim, o SLAVE utiliza a abordagem IRL, obtendo regras diferenciadas a cada geração. É necessário ressaltar que a melhor regra da execução, depois de inserida no banco de dados, é penalizada. Dessa forma, um máximo global é evitado.

As regras que o sistema utiliza possuem diversos antecedentes e um consequente. Os antecedentes são as condições que devem ser satisfeitas e o consequente é o resultado do processamento dos antecedentes. Um indivíduo do Algoritmo Genético é uma representação binária de n elementos, onde n é o total de antecedentes somado a um, que representa o consequente (GONZÁLEZ; PÉREZ, 1999). A mutação modifica um gene de um indivíduo e o cruzamento ocorre sobre dois pontos, ou seja, dois pontos de corte são estabelecidos em dois elementos e dois filhos são gerados a partir dos pontos de corte. A condição de parada do Algoritmo Genético ocorre quando um número de gerações chegou a um valor estipulado, ou quando um máximo global é encontrado, ou seja, nenhum outro valor melhor é obtido na população. A função *fitness* é uma função de soma de acertos da regra. Dessa forma, a cada iteração, o indivíduo que mais acertou regras é armazenado.

Base de Dados

Uma base de dados é um conjunto de regras extraídas de um especialista que conhece determinado domínio. Dessa forma, o algoritmo de aprendizado faz uso de um sub-conjunto da base para realizar o aprendizado e testa-o, utilizando, para tanto, o complemento do conjunto de testes.

A base de dados na ferramenta KEEL deve obedecer a algumas restrições: cada atributo deve ser separado por vírgula e espaço; cada regra deve ocupar uma e somente uma linha; o consequente da regra deve estar disponível e a extensão do arquivo de texto deve ser *.dat*.

Um exemplo da base de caso *iris* utilizada pelo SLAVE pode ser visualizada na figura 9. A base de dados disponibiliza informações sobre três diferentes classes de plantas usando 150 exemplos (GONZÁLEZ; PÉREZ, 1999). As variáveis, ou seja, os antecedentes são números reais e o consequente é uma expressão, relatando a classe da planta.

A partir da construção correta da base de dados, um algoritmo pode ser executado utilizando esta base.

Figura 9 - Um sub-conjunto da base de dados *iris*

Fonte: González e Pérez (1999)

Um exemplo de execução do SLAVE

Com uma base codificada corretamente, importada à seção *Experiments*, o algoritmo SLAVE é conectado com a base de dados e a execução do sistema pode ser efetivada. A figura 10 apresenta a execução do sistema SLAVE, utilizando a base de dados *iris*.

É possível observar que, de 150 casos avaliados, 147 foram reconhecidos com sucesso (*Success: 147.0*). A seguir, são apresentados os erros (*mistakes*) e a quantidade de regras que não foram classificadas. Também, a descrição da quantidade de regras, neste caso, quatro regras: a regra número zero acertou cinquenta casos e não obteve nenhum erro, a regra número um acertou 47 casos e a regra número dois acertou cinquenta casos e errou três. A regra número 3 não acertou e não errou, sendo eliminada da iteração seguinte. São apresentadas também as taxas de sucesso no treinamento (0.98, ou seja, 98%), a taxa de sucesso quando os testes foram aplicados, o número de regras, o número de variáveis por regra, ou seja, o número de antecedentes, a quantidade de variáveis utilizadas, a quantidade de iterações e as condições.

Vale ressaltar que estas execuções do SLAVE tiveram como intuito auxiliar na compreensão do algoritmo.

Figura 10 - Resultados finais da execução do SLAVE utilizando a base *iris*

Fonte: González et al. (1999)

Resultados

A ferramenta KEEL é de grande auxílio para a execução de diversos sistemas. Além de fornecer uma lista contendo diversos tipos de algoritmos, como Redes Neurais Artificiais, Algoritmos Genéticos, Sistemas Genético-Fuzzy, *Support Vector Machines* e Aprendizado baseado em Regras, a maneira para fazer com que um algoritmo reconheça uma base de dados é padronizada. Dessa forma, ao utilizar a ferramenta, não há necessidade de formatar uma base de acordo com as restrições de um algoritmo, mas sim de acordo com as restrições da ferramenta.

Quanto à execução do algoritmo SLAVE para a base de dados *iris*, aquele demonstrou um bom desempenho, pois o sistema acertou 98% dos casos de teste, o que pode ser considerado satisfatório. Vale ressaltar que os 2% de casos com erro podem ser atribuídos a diversos fatores, como ruídos nos dados, a escolha inadequada de alguns parâmetros e o tamanho da base.

Conclusão

No presente trabalho, foi apresentada uma ferramenta que possui diversos algoritmos de aprendizado. Além de executar em diferentes bases de casos, a ferramenta consegue lidar com extração de dados, ou seja, explorar diferentes bases de conhecimento, de maneira a apresentar um resultado satisfatório.

Ferramentas como a KEEL são de grande ajuda para diversos pesquisadores e estudantes. Muitas vezes, são necessários diversos testes, com diferentes abordagens e técnicas para que um problema possa ser resolvido. Uma saída para descobrir qual abordagem pode ser destacada é utilizar a ferramenta e analisar os resultados. Além disso, a ferramenta estudada mostrou-se de fácil compreensão, abrangente e apresenta inúmeras funcionalidades. Isto foi possível evidenciar por meio do gerenciamento correto dos dados e a partir das possibilidades de análise de projetos propostos.

Os Algoritmos Genéticos, além de resolverem problemas de otimização, são de grande importância para o aprendizado de máquina. Utilizando regras e uma base de dados, é uma poderosa técnica de aprendizado e tomada de decisões.

Neste contexto, o algoritmo SLAVE demonstrou, por meio do experimento realizado, que é possível utilizar um Algoritmo Genético-Fuzzy no aprendizado de máquina e no auxílio à tomada de decisão.

E por último, destaca-se a relevância de estudos como o proposto, na medida em que esse permite aprofundar os estudos referentes a novas tecnologias e, por consequência, a ferramentas computacionais de forma mais específica.

Como trabalhos futuros, recomendam-se o desenvolvimento de um estudo comparativo entre os algoritmos da ferramenta KEEL e, também, a realização de testes e comparações de Sistemas Inteligentes e ferramentas como a estudada.

Referências

ALCALÁ-FDEZ, J. et al. KEEL: a software tool to assess evolutionary algorithms to data mining problem. *Soft Computing*, 13:3 (2009) 307-318. Springer-Verlag. 2008. DOI 10.1007/500500-008-0323-y disponível em <<http://www.keel.es>>.

BERTHOLD, M. R.; CEBRON, N.; DILL, F.; GABRIEL, T.R.; GEORGE, F.; MEINL, T.; OHL, P. *KNIME: the konstanz information miner*. proceedings of the 4th ISC'2006 - The Industrial Simulation Conference. Workshop on multi-agent systems and simulations. June 5-7, 2006 - UNIVERSITY OF PALERMO: Palermo, Italy 2006.

EIBEN, A. E. ; SMITH, J.E. Introduction to evolutionary computing, *Natural computing series*, 1. ed. Berlin: Springer, 2003.

GALEA, M.; SHEN, O.; SINGH, V. Encouraging complementari Fuzzy rules within iterative rule learning. *Repository online CADAIR*. Aberystwyth University. Disponível em: <<http://cadair.aber.ac.uk/dspace>>, 2005.

GOLDBERG, D. E. *Genetic algorithms in search, optimization, and machine learning*. Boston: Addison-Wesley, 1989.

GONÇALVES, R. A. Computação Evolutiva. In: ESCOLA REGIONAL DE INFORMÁTICA - SBC, 12., 2004, Guarapuava. *Anais...* Guarapuava: EDUNI, 2004.

GONZÁLEZ, A.; PÉREZ, R. Selection of relevant features in a fuzzy genetic learning algorithm. *IEEE transactions on systems, man and cybernetics – Part B, Cybernetics*, V. 31, N. 3, 2001.

_____. SLAVE: a genetic learning system based on an iterative approach. *IEEE Transactions on Fuzzy Systems*, V. 7, N. 2, 1999.

HOU, E.S.H.; REN ANSARI, H. A genetic algorithm for multiprocessor scheduling. *IEEE trans. On parallel and distributed systems*, V. 5, N. 2, p. 113-120, 1994.

MICHALEWICZ, Z. *Genetic algorithms + data structures = evolution programs*. London: IE-Springer-Verlag, 1997.

MUCIENTES, M.; MORENO, L.; BUGARIN, A.; SENEN, B. Evolutionary learning of a fuzzy controller for wallfollowing behavior in mobile robotics. *Soft Computing*, p. 881-889, 2006.

NEUMANN, L. G.; ROSA, T. F.; JUNIOR, V. B.; SOUZA, J. S. S.; FERRÃO, M. F.; FURTADO, J. C. Otimização combinatória empregando algoritmo genético aplicada na análise multivariada de medicamentos manipulados. ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, ENEGEP, 24, 2004, Florianópolis, *Anais...* [s.l.,s.d.]

RAKOTOMALALA, R. Tanagra: un logiciel gratuit pour l'enseignement et la recherche. *Proceedings of the 5th Journées d'Extraction et Gestion des Connaissances*, p. 697-702, 2005.

REZENDE, S. O. *Sistemas inteligentes*. São Paulo; Manole Ltda., 2005.

ROMERO, C.; VENTURA, S.; DE BRA, P. Knowledge discovery with genetic programming for providing feedback to courseware author, user modeling and user-adapted interaction. *Journal of Personalization Research*, V. 14, p. 425-465, 2004.

SMITH, S. F. *A learning system based on genetic adaptive algorithms*. University of Pittsburgh, Pittsburgh, PA, USA. 1980 214 f. Tese (Doutorado em Ciência da Computação).

WITTEN, I. H.; FRANK, E. *Data mining: practical machine learning tools and techniques*. 2. ed. San Francisco: Morgan Kaufmann, p. 525, 2005.