Procedimentos de Validação Cruzada em Mineração de Dados para ambiente de Computação Paralela

Luciano Drosda M. dos Santos, Renê Mikami,
Ana Cristina B. Kochem Vendramin, Celso Antônio A. Kaestner

Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná (UTFPR) – Curitiba, PR – Brasil

{lucianodrosda,rene0907}@gmail.com,
{cristina,kaestner}@dainf.ct.utfpr.edu.br

Abstract. *This paper presents a performance evaluation of a Data Mining task in a distributed computational environment that uses a computer cluster. The tests are performed considering the Cross Validation procedure, which is employed to evaluate accuracy of the data mining algorithms. The results are obtained for a single machine, and for the cluster with different number of nodes, which are gradually increased. Obtained results prove that this is an interesting alternative for use in computer-intensive applications.*

Resumo. *Este artigo apresenta uma avaliação de desempenho de uma tarefa de Mineração de Dados em um ambiente de computação paralela formado por cluster de computadores. Os testes são feitos considerando a Validação Cruzada, mecanismo usado para avaliar a taxa de acerto dos algoritmos de mineração de dados. São obtidos os resultados para a execução em uma única máquina e empregando-se o cluster, que tem seu número de nós aumentado gradativamente. Os resultados obtidos comprovam que esta é uma alternativa interessante para a solução de problemas computacionalmente intensivos.*

1. Introdução

Apesar do desenvolvimento dos equipamentos de informática ocorrido nas últimas décadas, a necessidade de processar dados cada vez mais complexos e em maior quantidade faz com que a construção de computadores de alto desempenho continue sendo uma importante área de pesquisa em Computação.

Uma alternativa econômica e muito importante aos computadores de alto desempenho são os *clusters* de computadores. Um *cluster* é basicamente um sistema formado de múltiplos computadores, cada um dos quais de poder computacional limitado, conectados por meios de comunicação adequados, onde é possível executar um único programa usando todas as máquinas simultaneamente [Pitanga 2002]. Este procedimento faz com que o sistema como um todo tenha um poder computacional elevado e em muitos casos comparável ao de um computador de alto desempenho.

No caso de problemas paralelizáveis, que exigem muito processamento, o uso de *clusters* é uma solução alternativa a um custo razoavelmente baixo se comparado com os altos valores necessários para a aquisição de um computador da mesma classe de processamento [Pitanga 2002]. A utilização de *clusters* apresenta outras vantagens: (a) escalabilidade, pois é possível que novos componentes sejam adicionados à medida que cresce a carga de trabalho; (b) tolerância a falhas, pois o sistema pode continuar

funcionando mesmo caso alguma máquina falhe; (c) independência de fornecedores, pois, diferentemente dos sistemas proprietários dos supercomputadores, *clusters* podem utilizar máquinas com hardware aberto e software livre.

Este trabalho objetiva avaliar o potencial dos *clusters* em um procedimento específico de mineração de dados. As tarefas de mineração de dados são sabidamente computacionalmente intensivas, pois muitas vezes exigem o tratamento de problemas de otimização combinatória [Papadimitriou e Steiglitz 1988]. Com isso representam uma tarefa adequada para avaliar o desempenho de um ambiente computacional paralelo.

Em linhas gerais este trabalho consiste no desenvolvimento de um programa que realiza a distribuição de uma tarefa de mineração de dados em um *cluster*, na execução do algoritmo de mineração considerado em cada componente do *cluster*, e na obtenção dos resultados de desempenho. De forma a avaliar a vantagem ou não desta solução, são obtidos os resultados da execução da tarefa em um único computador e no *cluster* – que tem o número de nós gradativamente aumentado – e feitas as comparações necessárias.

Para possibilitar a execução de testes de desempenho neste trabalho, utilizou-se um *cluster* do tipo *Beowulf* [Alecrim 2005], usando software livre, com 11 computadores interligados em rede Gigabit Ethernet.

2. Mineração de Dados

Com o crescimento das bases de dados geradas de forma automática pelos sistemas computacionais, realizar análises manuais e intuitivas para extração de informações relevantes a um determinado problema tornou-se inviável. Entretanto, a descoberta de novas informações para identificar novas oportunidades é de extrema importância para a tomada de decisões estratégicas por empresas [Cabena *et al* 1997].

A *Mineração de Dados* é uma das etapas da área denominada Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases – KDD*), que desperta grande interesse junto às comunidades científicas e industriais [Goldschmidt e Passos 2005]. O processo de Mineração de Dados consiste, basicamente, em extrair informações interessantes, previamente desconhecidas e não-triviais de conjuntos de dados. Esses dados podem provir de sistemas de banco de dados, de arquivos texto, ou mesmo de outros sistemas, utilizando *data warehousing* [Mattison 1997].

2.1. Weka

Dentre as ferramentas desenvolvidas para a tarefa de mineração de dados destaca-se o ambiente Weka (*Waikato Environment for Knowledge Analysis*). Trata-se de uma coleção de algoritmos para aprendizado de máquina integrada em um ambiente computacional desenvolvido pela Universidade de Waikato [Weka 2008].

O ambiente Weka é escrito em Java e publicado sob licença GPL, o que facilita seu uso e divulgação. Além da possibilidade de uso via interface gráfica nativa e linha de comando, as classes integrantes do pacote Weka podem ser chamadas por qualquer outra aplicação Java, o que possibilita ao usuário escrever seus próprios programas, utilizando os algoritmos para mineração de dados. Estas facilidades foram consideradas na escolha deste ambiente para a realização deste trabalho, visto que se desejava realizar todo o desenvolvimento do projeto em Linguagem Java.

O Weka possui procedimentos para tratar as diversas etapas do processo de extração do conhecimento. Inicialmente, ele pode atuar na seleção, pré-processamento e transformação dos dados. Este processo consiste basicamente na escolha dos dados a serem analisados, na redução de valores ruidosos, no tratamento de inconsistências e em modificações nos dados que levam a um formato apropriado para a análise. Em seguida, ocorre a mineração de dados propriamente dita, onde se aplicam algoritmos específicos aos dados, obtendo-se os resultados da mineração. Finalmente, ocorre a visualização e interpretação dos dados, quando os resultados obtidos são analisados permitindo a obtenção do conhecimento que será útil ao problema em questão.

2.2. Validação Cruzada

Tipicamente em um processo de aprendizagem supervisionada em mineração de dados, após o pré-processamento e a formatação, os dados são fragmentados em dois subconjuntos, denominados base de treinamento e base de testes.

Numa primeira etapa um algoritmo de indução de conhecimento é aplicado à base de treinamento. Com isso se obtém um modelo "treinado", que de certa forma representa o conhecimento extraído. Numa segunda etapa o modelo obtido é aplicado ao fragmento da base de dados denominado base de testes. Como a base de testes também é previamente rotulada, se pode medir a taxa de acerto do modelo, comparando-se o resultado obtido com a rotulação disponível na base de testes [Cabena *et al* 1997].

A técnica de Validação Cruzada consiste em dividir a base de dados em ***x*** partes (*folds*). Destas, ***x-1*** partes são utilizadas para o treinamento e uma serve como base de testes. O processo é repetido ***x*** vezes, de forma que cada parte seja usada uma vez como conjunto de testes. Ao final, a correção total é calculada pela média dos resultados obtidos em cada etapa, obtendo-se assim uma estimativa da qualidade do modelo de conhecimento gerado e permitindo análises estatísticas.

3. Implementação

O pacote Weka permite várias formas de customização. Apesar da possibilidade da criação de scripts Shell que executem os procedimentos de Mineração de Dados via linha de comando, a opção considerada ideal foi escrever programas Java para executar as tarefas necessárias. Essa escolha, além de permitir a reutilização de classes nativas do Weka e melhor integração com os algoritmos de *Mineração de Dados*, possibilita a criação de várias *threads* e de um sistema de conexão cliente-servidor.

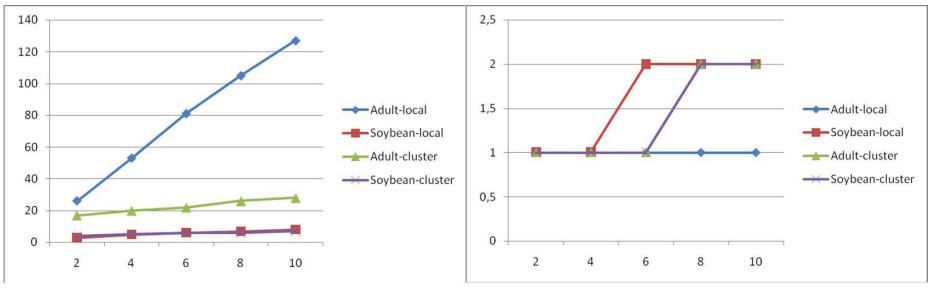
Assim foi criado um programa para o servidor, e outro para os clientes. O programa servidor, primeiramente, prepara os dados que serão processados pelos clientes. Em seguida, conecta-se com todos os clientes simultaneamente, criando uma *thread* para cada conexão e enviando as informações necessárias para cada cliente. Após os clientes enviarem os resultados de seu processamento, o servidor analisa as informações para gerar um relatório final do modelo de Validação Cruzada.

4. Testes de desempenho e resultados

Para avaliar o desempenho da aplicação em ambiente paralelo, foram comparados os resultados da *Validação Cruzada* para as bases de dados *Adult* (disponível em http://archive.ics.uci.edu/ml/datasets/Adult) e *Soybean*, disponibilizada no pacote do Weka, usando os algoritmos de classificação *J48* e NaiveBayes. A base de dados *Adult* é

composta por 14 atributos em 48842 instâncias. A base de dados *Soybean*, por sua vez, é composta por 36 atributos em 683 instâncias. Assim, enquanto a primeira possui muitas instâncias com poucos atributos, a segunda possui poucas instâncias, porém com um número maior de atributos em cada instância.

As comparações estão presentes para Validação Cruzada usando de 2 a 10 *folds*, cada qual processado em um nodo. A Figura 1 contém um resumo dos resultados obtidos, indicado-se a base de dados trabalhada, e se o processamento foi realizado em máquina local ou em *cluster*.



(a) **(b)**

Figura 1. Resultados aplicando o algoritmo *J48* (a), e o algoritmo *NaiveBayes* (b).
O eixo X indica o número de nós do *cluster* e o eixo Y o tempo de processamento em segundos.

5. Conclusões

Este trabalho apresentou um experimento que utiliza *clusters* de computadores para reduzir o tempo de execução em problemas de *Mineração de Dados*. Buscou-se avaliar a redução obtida na Validação Cruzada, empregada para quantificar o desempenho do modelo de classificação obtido. Os resultados obtidos comprovam que esta é uma alternativa a ser considerada no caso de problemas de alta complexidade computacional.

Agradecimentos

Este trabalho foi desenvolvido no âmbito de convênio entre a empresa Nova Soluções em Informática S/A e a Fundação de Apoio à Educação, Pesquisa e Desenvolvimento Científico e Tecnológico da Universidade Tecnológica Federal do Paraná.

Referências

Alecrim, E. (2005), "Cluster: principais conceitos", http://www.infowester.com/cluster.php, Acesso em 12/Nov/2008.

Cabena, P., Hadjinian, P., Stadler, R., Verhees, J e Zanasi, A. (1997), Discovering Data Mining: From Concept to Implementation, Prentice Hall.

Goldschmidt, R., Passos, E. (2005), Data Mining: um guia prático, Campus.

Mattison, R. (1997), Data Warehousing and Data Mining for Telecommunications, Artech Computer Science Library.

Papadimitriou, C.H.; Steiglitz, K. (1998) Combinatorial Optimization: Algorithms and Complexity, Dover Publishing.

Pitanga, M. (2002), Construindo Supercomputadores com Linux, Brasport.

Weka (2008), "Weka 3 - Data Mining with Open Source Machine Learning Software in Java", http://www.cs.waikato.ac.nz/ml/weka/, Acesso em 12/Nov/2008.