Avaliação do Desempenho Paralelo de um Algoritmo de Mineração de Dados do Pacote Estatístico R

Gabriel Machado Lunardi[1], **Patrícia Pitthan Barcelos**[1], **Andrea S. Charão**[1]

[1]Laboratório de Sistemas de Computação
Universidade Federal de Santa Maria (UFSM) - Santa Maria, RS - Brazil

{glunardi,pitthan,andrea}@inf.ufsm.br

Resumo. *O presente trabalho apresenta a avaliação de desempenho paralelo de um algoritmo de mineração de dados do pacote estatístico R. Tal avaliação é importante, visto que vivemos em uma era em que a quantidade de dados é muito grande, necessitando assim, de técnicas e ferramentas de processamento com alta performance. Os resultados mostram que o algoritmo pode ser eficaz, mas não tão eficiente.*

1. Introdução

O framework R é uma linguagem de programação e ao mesmo tempo um ambiente de desenvolvimento livre, utilizado para análise estatística de dados [Venables and Smith 2010]. Esse ambiente possui extensões e funções que auxiliam na mineração de dados, como algoritmos de classificação, agrupamento, associação, dentre outros.

Há pouco tempo surgiu um pacote denominado Multicore, para processamento paralelo de código R em arquiteturas com múltiplos núcleos. Esse recurso pode ser combinado com outros pacotes em R para acelerar análises que consomem muito tempo. Um desses pacotes é o Caret (Classification And REgression Training) [Kuhn 2010], cujo desempenho sequencial foi analisado em um *benchmark*, variando-se o tamanho dos dados [Engelhardt 2010].

Neste contexto, o objetivo do presente trabalho é avaliar o desempenho paralelo de um algoritmo de mineração de dados, associando os pacotes Caret e Multicore. O trabalho justifica-se pela popularidade do ambiente R em análises estatísticas, e também pela importância de se obter alto desempenho na mineração de grandes quantidades de dados. Além disso, cabe salientar que o pacote Multicore é recente e que não existem estudos aprofundados sobre seu desempenho, sendo uma boa oportunidade ao presente trabalho. Ao longo do artigo, serão apresentados o ambiente R, suas extensões, o *benchmark* utilizado e por fim os resultados obtidos.

2. O Pacote R e suas Extensões

O ambiente R agrega um conjunto de recursos para manipulação, processamento e visualização de dados. Além dos recursos nativos, existem extensões que implementam os mais variados procedimentos de análise de dados. Dentre algumas extensões que se prestam à mineração de dados, tem-se os pacotes Caret (short for Classification And REgression Training) [Kuhn 2010] e Boruta [Kursa and Rudnicki 2010]. O primeiro é uma compilação de funções que visam facilitar a criação de modelos. O segundo é um algoritmo para procura de atributos importantes em sistemas de informação por aprendizagem

interativa. Além destes, também merece destaque o pacote Rattle [Togaware 2010], que reúne vários algoritmos de mineração de dados e uma interface gráfica que viabiliza ainda mais a utilização do R para esse fim.

No que diz respeito ao processamento paralelo, o pacote Multicore [Eugster 2009] provê recursos para a execução paralela de código R em arquiteturas *multicore*, em sistemas baseados em Unix. Uma de suas principais funções é mclapply, que permite aplicar uma dada função em paralelo sobre diferentes porções de uma lista de dados. O pacote gerencia os processos e *jobs* paralelos, dividindo a computação entre os núcleos disponíveis.

Caret é usado para simplificar problemas complexos de regressão e classificação. Uma de suas características positivas é a de permitir a execução de códigos R em paralelo, se o recurso Multicore estiver instalado. O pacote Caret procura não carregar todas as funções quando é iniciado. Estas vão sendo carregadas à medida que são requisitadas, garantindo o desempenho [Kuhn 2010].

3. Descrição do Benchmark

Ao adotar uma técnica de mineração de dados, é necessário analisar quais variáveis são relevantes para a mineração. A técnica responsável por esse trabalho é chamada de seleção de características *(feature selection)*. Uma análise com todas as variáveis de um determinado objeto de estudo é inviável, visto que a grande quantidade de dados degrada o desempenho dos algoritmos e os recursos computacionais, além do resultado final da análise. É para sanar e prevenir esses problemas que a seleção de características existe.

Para elucidar a prática da mineração de dados, foi utilizado um *benchmark* escrito em R chamado bench-caret.R [Engelhardt 2010]. Sua estrutura é a seguinte: inicialmente são carregados os pacotes Caret e RandomForest (conjunto de algoritmos utilizados para ordenação); em seguida são geradas amostras aleatórias de dados, com tamanhos crescentes; os dados gerados são analisados pelo trecho de código responsável pelo processamento e, na sequência, o resultado é impresso, constando o tamanho da amostra testada e o tempo decorrente.

4. Avaliação de Desempenho

Para avaliar o desempenho do algoritmo bench-caret.R, foi utilizado um computador com dois processadores Intel Xeon 2.00GHz de 4 núcleos (8 núcleos ao total), sistema operacional Linux 64 bits (Kernel 2.6.20), o ambiente estatístico R com os pacotes Multicore e Caret.

Os testes foram realizados com 1 a 8 núcleos, sendo que para cada caso foram executadas 3 rodadas do mesmo teste. Isso permitiu uma verificação nas flutuações de tempo para uma mesma quantidade de núcleos. Após a coleta dos dados de desempenho, pode-se calcular a média, o desvio padrão e o coeficiente de variação para cada núcleo, englobando as 3 rodadas de cada um. Não foram realizadas mais execuções porque o coeficiente de variação foi baixo (de 0 a 0.03) para todos os casos. Além disso, foi executada uma rodada sem o pacote Multicore para averiguar a influência da função nos tempos.

Como mostra a figura 1, o teste que levou mais tempo foi com a utilização de somente um núcleo. Houve uma grande diferença de tempo utilizando 1, 2, 3, 4 e 5 núcleos. A partir daí as diferenças passaram a ser pequenas. Cabe salientar que com 5 e

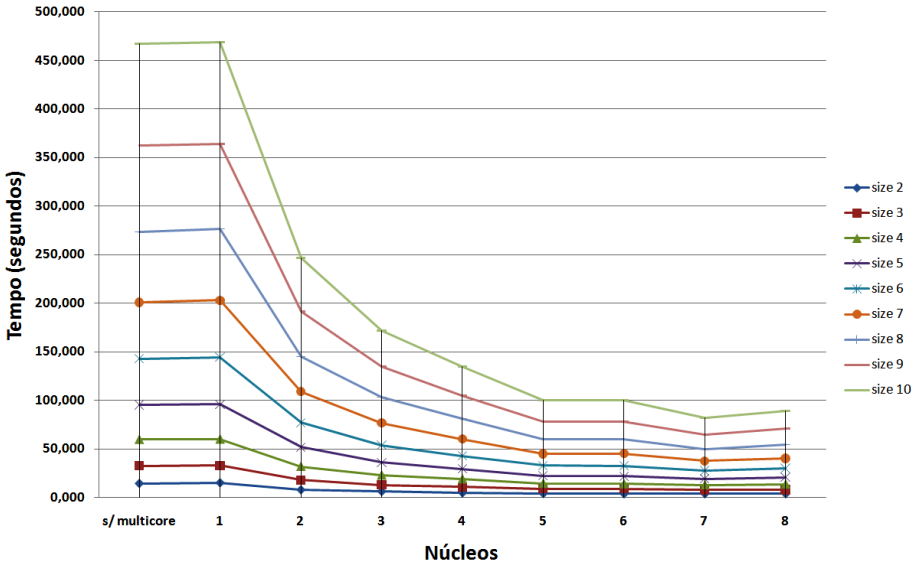

Figura 1. Tamanho da amostra em relação a quantidade de núcleos e ao tempo

6 núcleos os tempos foram praticamente iguais. Intuitivamente, pode-se pensar que com 8 núcleos o teste teria sido realizado mais rápido. No entanto, não foi o que ocorreu. O teste executou mais rapidamente com 7 núcleos. Para os tamanhos de amostra 4, 5, 6, 7, 8, 9 e 10, os 8 núcleos foram mais lentos que os 7. Ao rodar o teste sem o Multicore, verificou-se uma diminuição no tempo de execução em relação ao teste com 1 núcleo e com a função. Isso ocorre, pois esse tipo de função gera um *overhead*.

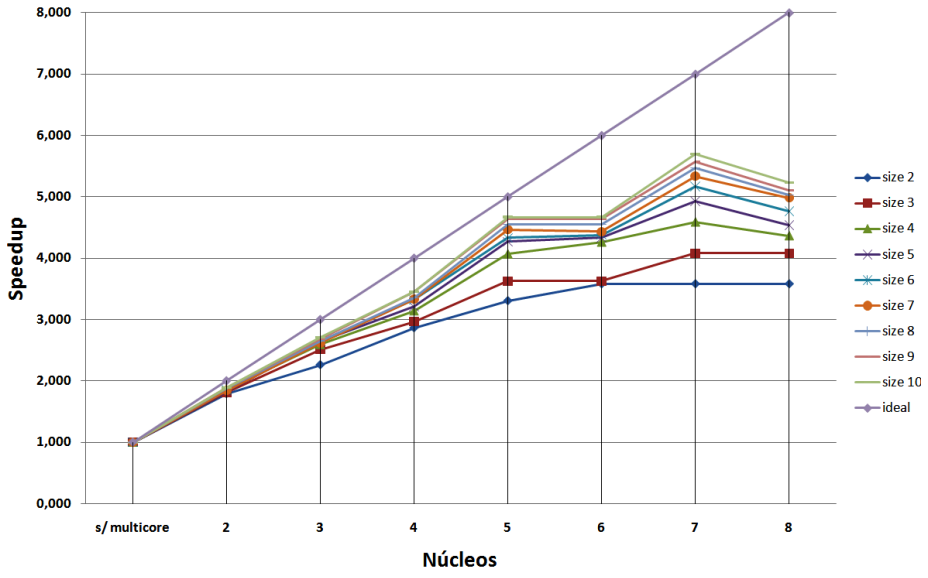

Figura 2. Speedup medido

A figura 2 traz o resultado de aceleração (*speed up*). Nela tem-se uma representação do que seria o desempenho ideal, mas como pode-se constatar, não é o que ocorre. Nota-se, no entanto, que quanto maior o tamanho de dados, maior a aceleração.

5. Conclusão

A combinação dos pacotes Caret e Multicore mostrou-se eficaz e apresentou resultados satisfatórios com, relativamente, grandes amostras de dados. O *benchmark* utilizado revelou que o recurso de processamento paralelo melhora o desempenho das análises na maioria dos casos. A aceleração, no entanto, não se mantém até os 8 núcleos disponíveis.

Pretende-se dar continuidade à esta pesquisa, principalmente para averiguar a causa do menor desempenho com 8 cores. Isso só será possível com uma análise mais profunda sobre os pacotes Caret, Multicore e do próprio algoritmo de mineração.

Referências

Engelhardt, A. (2010). Benchmarking feature selection with Boruta and Caret. Disponível em: <http://www.r-bloggers.com/benchmarking-feature-selection-with-boruta-and-caret/>. Acesso em: 3 Dezembro 2010.

Eugster, M. J. A. (2009). The multicore package - parallel computing with R tutorial, statistical computing, 2009. Disponível em: <http://www.informatik.uni-ulm.de/ni/staff/HKestler/Reisensburg2009/PDF/multicore.pdf>. Acesso em: 14 Dezembro 2010.

Kuhn, M. (2010). The Caret package. Disponível em: <http://cran.r-project.org/web/packages/caret/vignettes/caretTrain.pdf>. Acesso em: 14 Dezembro 2010.

Kursa, B. M. and Rudnicki, W. R. (2010). Feature selection with the boruta package. *Journal of Statistical Software*, 36. Disponível em: <http://www.jstatsoft.org/v36/i11/paper>. Acesso em: 13 Dezembro 2010.

Togaware (2010). Rattle: Gnome cross platform gui for data mining using R. Disponível em: <http://rattle.togaware.com/>. Acesso em: 21 Dezembro 2010.

Venables, W. N. and Smith, D. M. (2010). An introduction to R. Disponível em: <http://cran.r-project.org/doc/manuals/R-intro.pdf>. Acesso em: 21 Dezembro 2010.