myR – Uma Biblioteca C++ para Acesso ao R*

Pedro Ribeiro de Andrade Neto
Eduardo Sant'Ana da Silva
Thiago Eugênio Bezerra de Mello
Marcos Aurélio Carrero
Paulo Justiniano Ribeiro Júnior

Universidade Federal do Paraná
Departamento de Estatística
Caixa Postal 19081, CEP 81531-990. Jardim das Américas, Curitiba – PR.
{pedro,esantana,thiago,carrero}@inf.ufpr.br
paulojus@est.ufpr.br

1 Introdução

O programa estatístico R[4] é um aplicativo de código aberto e de domínio público. No R existem e são desenvolvidas inúmeras bibliotecas implementando uma diversidade de métodos estatísticos.

Em cada linguagem de programação existem várias bibliotecas que implementam diversas funções. Mas, em nenhuma delas, existe a quantidade de funções estatísticas disponíveis no R. Então, toda vez que se necessita de uma nova função estatística no seu programa em uma particular linguagem, ela tem que ser implementada na linguagem de programação que está sendo utilizada, mesmo que já tenha sido implementada no R. Uma solução seria o reaproveitamento do código. O único problema que temos então é: como ter acesso à funcionalidade do R através do meu programa?

A idéia central do trabalho é criar um mecanismo para acessar a funcionalidade do R a partir de um outro programa computacional, incorporando ao aplicativo a capacidade de efetuar análises estatísticas. Por exemplo, sistemas de informação geográfica (SIG's) são capazes de manipular diversas estruturas complexas de dados georeferenciados. Porém, possuem pouca, ou nenhuma, capacidade de fazer análises estatísticas nos dados. Portanto, esses sistemas podem se beneficiar dessa interação na medida que métodos estatísticos podem ser usados no processamento desses dados.

myR é uma biblioteca de programação escrita em C++ para o acesso a essas funcionalidades. Para utilizar o myR é necessário ter um conhecimento das funcionalidades do R, que podem ser encontradas em [3].

*Este trabalho é desenvolvido como parte das atividades do projeto SAUDAVEL (Sistema de Apoio Unificado para a Detecção e Acompanhamento em Vigilância EpidemioLógica). Mais detalhes em http://saudavel.dpi.inpe.br/. Este trabalho é financiado pelo CNPq.

Acesso	Requisitos	Desvantagem	Vantagem
Batch	R	Carregar o R, copiar e analisar os dados.	Implementação fácil e rápida.
R*serve*	libR + rede	Transporte e cópia dos dados.	Processamento distribuído em mais de uma máquina. Não precisa carregar o R.
Biblioteca	libR	Implementar a interface.	Não há duplicação dos dados na memória. R é carregado apenas uma vez.

Tabela 1: Comparando as três formas de utilizar o R.

2 Acessando o R

Existem três formas de acessar as funções do R, por *batch*[4], utilizando o R*serve*[5] e utilizando a biblioteca dinâmica libR[4]. Na Tabela 1 há uma descrição das vantagens, desvantagens e requisitos destas diferentes formas de se acessar o R.

2.1 Acesso por *batch*

É uma solução pouco eficiente. Escrevemos os comandos do R em um arquivo texto, e o invocamos através de uma chamada de sistema, passando o arquivo como argumento, e redirecionando a saída para outro arquivo texto. Esta solução é bastante rápida de ser executada, uma vez que seus passos são fáceis de serem implementados. As desvantagens dessa implementação são:

- Toda vez que o R tiver que ser chamado na aplicação ele tem que ser carregado, o que consome tempo, mesmo com operações simples como, por exemplo, calcular a média de alguns números;
- Os valores das variáveis a serem utilizadas no R têm que estar escritos em um arquivo texto, para ser então carregados pelo R. Mas esta operação, dependendo da quantidade dos valores, pode consumir mais tempo do que carregar o próprio R.

2.2 Acessar através do R*serve*

O Rserve é um servidor TCP/IP que provê uma interface para acessar o R remotamente, através de um cliente. Ele é executado como um servidor R, que fica aguardando requisições de conexão. Quando um usuário se conecta ao servidor, podem ser enviadas requisições de comandos e declarações de variáveis. O servidor Rserve necessita do R compilado dinamicamente (libR).

Existem dois programas clientes disponíveis no site do Rserve. Um desenvolvido em Java, e um protótipo de cliente em C. Esse programa cliente pode ser escrito em qualquer

linguagem de programação, desde que ela implemente funções cliente/servidor do protocolo TCP/IP.

O Rserve possui quatro funcionalidades para controle da aplicação e do servidor:

1. Definição de usuários autorizados a realizar requisições;
2. Especificação da porta a ser usada pelo servidor;
3. Possibilidade de uso do servidor remotamente ou localmente;
4. Opção para depurar a aplicação cliente.

A leitura e escrita de valores no R pode ser lenta, pois precisa passar todos os dados através da rede. Então o Rserve é indicado para uma grande quantidade de processamento, pois as aplicações podem ficar divididas, de modo que o cliente não precisa ter um computador de grande porte para ter acesso as funcionalidades do R.

O Rserve também pode ser útil para aplicações pequenas, e de baixo custo computacional, pois a perda com a transferência de dados pode compensar o ganho no tempo de implementação.

2.3 Acesso pela biblioteca libR

Esta é uma solução que acessa o libR diretamente. O código da biblioteca tem que ser estudado, para então descobrir os pontos de acesso ao R. Ele, assim como o Rserve, requer o R compilado dinamicamente. Mas, se a camada proposta por uma biblioteca deste tipo for implementada, ela terá as seguintes vantagens:

- O R é carregado apenas uma vez durante toda a execução do programa;
- O programa tem acesso direto às variáveis do R, e pode fazer com que o R também tenha acesso direto as suas variáveis.

Visto que a única desvantagem de acesso ao R por meio do libR é a implementação da interface de acesso, nosso trabalho se baseia no desenvolvimento dessa interface. Um relatório de como instalar e utilizar é descrito em [1].

3 Funções da biblioteca

Esta seção descreve as funções contidas na biblioteca para acesso ao R, e alguns comentários sobre suas funcionalidades.

3.1 Inicializando o R

Para iniciar o R dentro do programa, basta declarar uma variável do tipo myR. Ela já inicializará todas as variáveis necessárias para o funcionamento do R. Pode-se também iniciar o R utilizando argumentos, como se estivesse rodando o R de um terminal.

3.2 Executando uma operação

Existe uma função chamada `Execute`, que executa uma função do R. Essa função recebe dois parâmetros (1) `command`, que contém a função a se executada, e (2) `result`, que guarda o resultado da operação.

Esta função devolve um código de erro, indicando se a operação foi realizada com sucesso, se houve um erro na análise do comando, ou se houve um erro ao tentar executá-lo.

3.3 Declarando variáveis no R

Declarar variáveis utilizando operações, passando uma cadeia de caracteres para o R, pode ser bastante demorado e pode influenciar no desempenho do sistema. Para evitar isso, existe uma função para declarar variáveis no R, então os dois passam a compartilhar a mesma área de memória para objetos declarados dessa maneira. Esta função chama-se `DeclareVar`. Ela recebe como parâmetros (1) o nome que a variável terá no R, e (2) a variável a ser declarada. Esta função também diz para o R que esta variável está em uso, e não pode ser removida da memória.

3.4 Removendo variáveis

Após utilizar uma variável, pode ser necessário desalocar a memória alocada para ela. O R tem um controle próprio sobre a liberação da memória das variáveis utilizadas. Então basta avisr ao R que esta variável não está mais sendo utilizada. A função implementada tem o nome de `DeleteVar`.

4 Estudo de caso

Nesta seção mostraremos um exemplo de como utilizar a biblioteca dentro de um código C++. Na Figura 1 temos um exemplo comentado. O programa declara um vetor, e então aloca na memória para ele. A estrutura de dados utilizada é o SEXP, uma estrutura própria do R. Ele atribui valores esse vetor, e então o declara no R, com essa variável tendo o nome de x. Então é calculada a sua média, com a função *mean*. Finalmente o resultado é mostrado na tela.

5 Conclusões e trabalhos futuros

O myR se mostrou estável e robusto como uma interface para o R. O desenvolvimento subsequente consiste em utilizar o myR em um programa de grande porte, o Terralib/Terraview[2], um *software* livre de geoprocessamento. Isto possibilitará acessar a partir do aplicativo, os métodos estatísticos implementados no R, dotando o aplicativo de capacidade de realizar análises estatísticas, de forma prática e eficiente.

Figura 1: Executando o R no seu código.

```
myR R;
SEXP vetor, resp;

vetor = NEW_INTEGER(20); // alocamos memoria para o vetor

for(int i = 0; i != 20; i++)
    INTEGER(vetor)[i] = i; // atribuimos valores ao vetor

R.DeclareVar("x", vetor); // declaramos vetor no R com nome de x
R.Execute("mean(x)", resp); // calculamos a media do vetor

R.printSEXP(resp); // mostramos o resultado da operacao na tela
R.DeleteVar(vetor); // liberamos a memoria no R
```

Referências

[1] Andrade Neto, P. R. et al. *my*R – *Uma Biblioteca C++ para Acesso ao* R. http://www.inf.ufpr.br/~pedro/myR

[2] G. Câmara et al. *Terralib: Technology in support of gis innovation.* Em II Workshop Brasileiro de Geoinformática, GeoInfo2000, São Paulo, Brasil, 2000.

[3] R Development Core Team. *An Introduction to* R. R Foundation for Statistical Computing, 2003. Disponível em http://www.r-project.org/.

[4] R Development Core Team. R*: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2003. ISBN 3-900051-00-3.

[5] Urbanek, S. *Rserve - A Fast Way to Provide* R *Functionality to Applications*, Proc. of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003), 2003. Disponível em http://stats.math.uni-augsburg.de/Rserve/.