

R语言和其他计算机语言的混合编程

冯兴东

上海财经大学统计与管理学院

November 5, 2012

R的缺陷

R语言，Matlab等统计计算机语言提供了丰富的统计类功能函数。但是，由于其逐句解释编译的语言解读方式导致了程序运行耗时比较长。尤其在模拟计算过程中，R程序可能导致循环等越来越慢，影响大规模的数据模拟效率。

便利和效率

世界上无免费的午餐，在享受了R语言便利的同时，我们在大规模数据模拟中牺牲了效率。C语言，Fortran语言等编程相对繁琐，但是在科学计算中极具效率。

软件安装和计算机设置等

- 1 下载R安装包(比如安装了R-2.15.2)
- 2 下载Rtools安装包
- 3 设置环境变量“Path”
 - 1 右击“电脑”，选择“属性”
 - 2 选择“高级系统设置”
 - 3 选择“环境变量”
 - 4 创建或者编辑“Path”这个变量,加上这一路径:

32位系统 C:\Program Files\R\R-2.15.2\bin\i386\

64位系统 C:\Program Files\R\R-2.15.2\bin\x64\

R语言的接口程序 I

几个用于接口的重要R函数

- `dyn.load()`、`dyn.unload()`
- `.C()`、`.Call()`、`.Fortran()`

`dyn.load()`和`dyn.unload()`是分别用来装载和卸载编译过后的C或者Fortran程序，在Windows操作系统中通常是.dll文件。我们用C语言来做范例，需要在R程序中调用`.Call()`或者`.C()`函数来将参数传递给C程序。

R语言的接口程序 II

一个范例

```
dyn.load("orthogonal4");  
simulation<-function(loops,error,a,size,mu1,mu2,phi1,phi2,  
                    n,m,sigma1,sigma2,sigma.n,sigma.t,sigma.c)  
{.Call("Simulation",as.integer(loops),as.integer(error),  
as.double(a),as.integer(size),as.double(mu1),as.double(mu2),  
as.double(phi1),as.double(phi2),as.integer(n),as.integer(m),  
as.double(sigma1),as.double(sigma2),as.double(sigma.n),  
as.double(sigma.t),as.double(sigma.c),new.env());  
}  
  
.  
.  
.  
dyn.unload("orthogonal4");
```

C语言的接口程序 I

步骤

- 1 编写相应C程序
- 2 编译C程序
 - 1 在Windows中打开Dos窗口（Win 7系统中可以通过寻找并运行“cmd”命令打开该窗口）
 - 2 进入存储C语言或者Fortran源程序目录
 - 3 在Dos窗口中输入 R CMD SHLIB file.c或者file.f（其中file.c或者file.f是源程序文件名，在我们之前的范例中，我们输入R CMD SHLIB orthogonal4.c）
 - 4 根据编译提示信息判断是否编译成功

C语言的接口程序 II

编写C语言接口程序

以我们例子中的orthogonal4.c源程序来作为范例，在文件开始部分，需要包含一些R提供的头文件，这样我们就可以方便地调用一些R函数了！

```
/*Some useful header files from R*/  
#include <R.h>  
#include <Rinternals.h>  
#include <Rmath.h>  
/*Some Commonly used header files in computing*/  
#include <math.h>  
#include <stdio.h>  
#include <stdlib.h>
```


C语言的接口程序 III

R的数据结构

SEXP是R定义的数据结构，可以用来将R程序和C程序之间传递R的Object，包括整数、实数、向量以及list等等。需要注意的是，当R将一个矩阵传递给C程序的时候，其实是将矩阵转换成一个大的向量传递过来的（将列向量连接起来生成一个大的向量），因此在读数据的时候需要小心一些。比如说：*DMatrix*是一个传递过来的矩阵数据，我们同时需要传递给C程序有关其维数的信息(假设是一个 $n \times m$ 的矩阵)，然后将其做为一个大向量来读取数据:该矩阵的第*i*行第*j*列上的数据是 $DMatrix[i + (j - 1) * n]$ 。

读取数据所需函数:

REAL()、INTEGER()

C语言的接口程序 IV

在C程序中产生随机数

在做数据模拟中，产生随机数是必要一步。R拥有非常棒的随机数产生器，我们也可以在C程序中来调用这些函数。

产生随机数的常用函数：

GetRNGstate()、PutRNGstate()、rnorm()、rchisq()、rt()等等，更多随机数产生器函数可参看R的在线文档“Writing R Extensions”

C语言的接口程序 V

内存分配问题

当R和C混合编程的时候，R会来掌控内存的分配。如果在C程序中自己声明数组的时候，那么会出现数组被R释放或者占用的可能，从而导致程序的崩溃或者结果的差错。在这种情况下，我们需要使用R提供的保护措施来进行内存分配。

所需函数：

PROTECT()、allocVector()、UNPROTECT()

C语言的接口程序 VI

返回结果

在C程序中运行完成之后，将会返回结果到R里面进行进一步的运算或者画图等等。R提供了一种交互方式，即传递一个SEXP对象回R程序。
所需函数：

`SET_VECTOR_ELT()`、`SET_STRING_ELT()`、`setAttrib()`、`return()`

C语言的接口程序 VII

例子

```
SEXP Simulation(SEXP Loops, SEXP error, SEXP A, SEXP Size,  
SEXP Mu1, SEXP Mu2, SEXP Phi1, SEXP Phi2, SEXP dim1,  
SEXP dim2, SEXP Sigma1, SEXP Sigma2, SEXP Sigma_n,  
SEXP Sigma_t, SEXP Sigma_c, SEXP rho)  
{ SEXP stats,lists,names;  
  int n,m,loops;  
  SEXP theta[2];  
  n=INTEGER(dim1)[0];  
  m=INTEGER(dim2)[0];  
  loops=INTEGER(Loops)[0];  
  GetRNGstate();
```

C语言的接口程序 VIII

```
for(i=0;i<2;i++)PROTECT(theta[i]=allocVector(REALSXP,n));
for(j=0;j<n;j++)
{
REAL(theta[0])[j]=rnorm(REAL(Mu1)[j],REAL(Sigma1)[0]);
REAL(theta[1])[j]=rnorm(REAL(Mu2)[j],REAL(Sigma2)[0]);
}
PROTECT(lists=allocVector(VECSXP,1));
PROTECT(names=allocVector(STRSXP,1));
PROTECT(stats=allocVector(REALSXP,loops));
.
.
.
```

C语言的接口程序 IX

```
SET_VECTOR_ELT(lists,0,stats);  
SET_STRING_ELT(names,0,mkChar("stats"));  
setAttrib(lists,R_NamesSymbol,names);  
PutRNGstate();  
UNPROTECT(9);  
return(lists);  
}
```

R程序对于返回结果的处理

在C程序处理完毕之后，将会返回相应结果，那么R可以将其当做通常的R程序中的Object加以处理。接着上面的例子，我们可以这样处理：

R程序

```
temp=simulation(loops,error,a,l,mu1,mu2,phi1,  
               phi2,n,m,sigma1,sigma2,sigma.n,sigma.t,sigma.c)  
pvalues.null=1-pchisq(temp$stats,l)
```


Thank You!

谢谢!