C Fortran 90程式

在此提供本文所使用之Fortran 90程式。

```
!使用Linked list(鏈結串列)前，宣告Linked list的模式爲合
module typedef

   TYPE :: PermuteList
      INTEGER :: permute(25)
      TYPE(PermuteList), POINTER :: next !用來指向下一個節點
   END TYPE PermuteList

end module typedef


!一般副程式(subroutine)只能傳入參數，
!利用module讓指標可以當作參數傳入副程式InsertToPermuteList中
module pointerr
contains
!將執行副程式permutation後不重複的perm記錄在linked list中
subroutine InsertToPermuteList(num,perm,p1Head,p1Move)
USE typedef
INTEGER i,countt,num
INTEGER perm(num)
TYPE(PermuteList), POINTER :: p1Move,p1Head

!每次都從鏈結串列的最開頭開始判斷是否有重複
p1Move=>p1Head
do WHILE(ASSOCIATED(p1Move%next))
   countt=0
   do i=1,num
      if (perm(i)==p1Move%permute(i)) then
         countt=countt+1
      end if
   end do
   if (countt /= num) then
      p1Move=>p1Move%next
   else
      return
   end if
end do

do i=1,num
   p1Move%permute(i)=perm(i)
end do
ALLOCATE(p1Move%next)
end subroutine InsertToPermuteList


!利用遞迴將輸入的數值做排列(在此會有重複排列的情況發生)
recursive subroutine permutation(ch,num,perm,a,level,p1Head,p1Move)
USE typedef
```

```
TYPE(PermuteList), POINTER :: p1Head,p1Move
INTEGER i,level,num
INTEGER a(num),ch(num),perm(num)

level=level+1
do i=1,num
   if (a(i)==0) then
      a(i)=1
      perm(level)=ch(i)
      if (level==num) then
           call InsertToPermuteList(num,perm,p1Head,p1Move)
      end if
      call permutation(ch,num,perm,a,level,p1Head,p1Move)
      a(i)=0
   end if
end do
level=level-1
end subroutine permutation
end module

!此副程式主要在計算兩種方法與貝氏法的相對誤差及CPU的平均時間，
!並將結果輸出在result.txt中
subroutine permute(ii,jj,a,b,n,ch,mm,kk,expec,theta,u)
USE typedef
USE pointerr
TYPE(PermuteList), POINTER :: p1Head,p1Move,previous
INTEGER i,level,ii,jj,n,mm,kk,perm_num
REAL time_Gibbs,time_quasi,avetime_Gibbs,avetime_quasi
INTEGER aa(n),ch(n),perm(n)
REAL a(ii),b(ii,jj),expec(ii),theta(ii),u(ii)
real rel_Gibbs(ii),rel_quasi(ii),averel_Gibbs(ii),averel_quasi(ii)

WRITE(*,*) "輸入數字爲 ",ch
aa=0
level=0
ALLOCATE(p1Head)
ALLOCATE(p1Move)

NULLIFY(p1Head%next)
call permutation(ch,n,perm,aa,level,p1Head,p1Move)
!將最後一個沒用到的節點刪掉
do WHILE(.true.)
   previous=>p1Move
   p1Move=>p1Move%next
   if (.not. ASSOCIATED(p1Move%next)) then
      NULLIFY(previous%next)
      exit
   end if
end do
!列出扣掉重複後的所有可能
perm_num=0
averel_Gibbs=0
averel_quasi=0
avetime_Gibbs=0
avetime_quasi=0
```

```
do WHILE(.true.)
   WRITE(*,*) "Now r= ",(p1Head%permute(i),i=1,n)
   write(50,*) ""
   WRITE(50,*) "Now r= ",(p1Head%permute(i),i=1,n)
   call Gibbs(ii,jj,a,b,n,p1Head%permute,mm,kk,theta,time_Gibbs)
   call quasiBayes(ii,jj,a,b,n,p1Head%permute,u,time_quasi)

   !計算相對誤差及平均時間
   rel_Gibbs=ABS((theta-expec)/expec)
   rel_quasi=ABS((u-expec)/expec)
   averel_Gibbs=averel_Gibbs+rel_Gibbs
   averel_quasi=averel_quasi+rel_quasi
   avetime_Gibbs=avetime_Gibbs+time_Gibbs
   avetime_quasi=avetime_quasi+time_quasi
   perm_num=perm_num+1

   WRITE(*,"(a15,10f10.6)")"rel_Gibbs = ",rel_Gibbs
   WRITE(*,"(a15,10f10.6)")"rel_quasi = ",rel_quasi
   WRITE(50,"(a15,10f10.6)")"rel_Gibbs = ",rel_Gibbs
   WRITE(50,"(a15,10f10.6)")"rel_quasi = ",rel_quasi
   WRITE(50,*)
   write(50,*) "---------------------------------------"
   if (.not. ASSOCIATED(p1Head%next)) then
      WRITE(*,*)
      WRITE(*,"(a20,10f10.6)") "sum_rel_Gibbs is ",averel_Gibbs
      WRITE(*,"(a20,10f10.6)") "sum_rel_quasi is ",averel_quasi
      WRITE(*,*)
      WRITE(*,*) "sum of CPU-SEC of Gibbs is ",avetime_Gibbs,"sec"
      WRITE(*,*) "sum of CPU-SEC of quasi is ",avetime_quasi,"sec"
      WRITE(*,*)
      WRITE(*,*) "permute number is ", perm_num
      WRITE(50,*)
      WRITE(50,"(a20,10f10.6)") "sum_rel_Gibbs is ",averel_Gibbs
      WRITE(50,"(a20,10f10.6)") "sum_rel_quasi is ",averel_quasi
      WRITE(50,*)
      WRITE(50,*) "sum of CPU-SEC of Gibbs is ",avetime_Gibbs,"sec"
      WRITE(50,*) "sum of CPU-SEC of quasi is ",avetime_quasi,"sec"
      WRITE(50,*)
      WRITE(50,*) "permute number is ", perm_num
      do i=1,ii
         averel_Gibbs(i)=averel_Gibbs(i)/perm_num
         averel_quasi(i)=averel_quasi(i)/perm_num
      end do
      avetime_Gibbs=avetime_Gibbs/perm_num
      avetime_quasi=avetime_quasi/perm_num
      WRITE(*,*)
      WRITE(*,"(a25,10f10.6)") "average of rel_Gibbs is ",averel_Gibbs
      WRITE(*,"(a25,10f10.6)") "average of rel_quasi is ",averel_quasi
      WRITE(*,*)
      WRITE(*,*) "average of CPU-SEC of Gibbs is ",avetime_Gibbs,"sec"
      WRITE(*,*) "average of CPU-SEC of quasi is ",avetime_quasi,"sec"
      WRITE(50,*)
      WRITE(50,"(a25,10f10.6)") "average of rel_Gibbs is ",averel_Gibbs
      WRITE(50,"(a25,10f10.6)") "average of rel_quasi is ",averel_quasi
      WRITE(50,*)
```

```
      WRITE(50,*) "average of CPU-SEC of Gibbs is ",avetime_Gibbs,"sec"
      WRITE(50,*) "average of CPU-SEC of quasi is ",avetime_quasi,"sec"

      exit
   end if
   p1Head=>p1Head%next
end do
END subroutine permute

!計算貝氏法的驗後均數
subroutine Bayes(ii,jj,a,b,n,r,expec)
USE typedef
TYPE(PermuteList), POINTER :: p1Head,p1Move
integer ii,jj,i,j,m,n,L
INTEGER(KIND=4) k
REAL t1,t2,constant_a,constant_b2,constant_b1
REAL Coef_A,sum_CoefA
integer r(n),index_k(n),c(ii,jj)
REAL a(ii),b(ii,jj),expec(ii)
REAL coef_B,sum_CoefB
!REAL coef_B(ii),sum_CoefB(ii)
!external alngam, umach,dgamma

t1=cpsec()

!計算期望值
do m=1,ii

   sum_CoefA=0.0
   sum_CoefB=0.0
   !將迴圈單一化
   do k=1,ii**n
      L=k-1+(ii**n-1)/(ii-1)
      do i=1,n
         SELECT case(MOD(L,ii))
            case (0)
               index_k(i)=ii
               L=INT(L/ii)-1
            case default
               index_k(i)=MOD(L,ii)
               L=INT(L/ii)
        end select
      end do
      c=0
      do i=1,n
         c(index_k(i),r(i))=c(index_k(i),r(i))+1
      end do
      constant_a=1.0
      constant_b2=1.0
      constant_b1=1.0
      call calculate(c,ii,jj,a,b,constant_a,constant_b1,constant_b2)
      Coef_A=constant_a*constant_b1
      sum_CoefA=sum_CoefA+Coef_A
      Coef_B=Coef_A*REAL(COUNT(index_k==m))
      sum_CoefB=sum_CoefB+Coef_B
```

```
   end do
   expec(m)=(a(m)+sum_CoefB/sum_CoefA)/(SUM(a)+REAL(n))
   WRITE(*,"(' Expect theta ',i2,'=',E20.10)") m,expec(m)
END do
!do m=1,ii
!   expec(m)=(a(m)+(sum_CoefB(m))/sum_CoefA)/(SUM(a)+REAL(n))
!end do
t2=cpsec()

WRITE(*,*)
write(*,*)" Bayes method "
write(*,"(a16,100F8.4)") " expectation = ",expec
write(*,*)" CPU-SEC : Bayes method約",t2-t1,"秒"
WRITE(*,*) " "

WRITE(50,*)
write(50,*)" Bayes method "
write(50,"(a16,100F15.12)") " expectation = ",expec
write(50,*)" CPU-SEC : Bayes method約",t2-t1,"秒"
WRITE(50,*) " "
write(50,*) "---------------------------------------"
WRITE(50,*) ""
end subroutine Bayes

!計算吉氏取樣器的估計值
subroutine Gibbs(ii,jj,a,b,nn,r,mm,kk,theta,time_Gibbs)
USE typedef
TYPE(PermuteList), POINTER :: p1Head,p1Move
integer::ii,jj,nn,mm,kk
integer::i,j,k,n,m,ans
real a(ii),b(ii,jj),aa(ii),bb(ii,jj),b_row(jj)
real u(ii),uu(ii),u_k(kk,ii),u_out(mm,ii)
real G(ii,jj),GG(ii,jj),G_row(jj),G_k(kk,ii,jj),G_out(mm,ii,jj)
real d(ii),dd(ii)
real yy(nn),p(ii),pp(ii,jj)
integer r(nn),y(nn),x(ii),z(ii,jj),total(jj)
real theta(ii)
integer delta(ii,jj)
real l(ii)
integer::iseed1,iseed2,iseed3
REAL t1,t2,time_Gibbs

aa=a
bb=b

!Gibbs sequence3方程式
!第一步：訂出起始值
!given y

t1=cpsec()
a=aa
b=bb
do i=1,II
   u(i)=a(i)/sum(a)
   do k=1,JJ
```

```
      G_row(k)=b(i,k)
   end do
   do j=1,JJ
      G(i,j)=b(i,j)/sum(G_row)
   end do
end do

do m=1,mm
   iseed1=123**m
   call rnset(iseed1)
   call RNUN(nn,yy)

   do k=1,nn
      do i=1,ii
         l(i)=u(i)*G(i,r(k))
      end do
      do i=2,ii
         l(i)=l(i)+l(i-1)
      end do
      do i=1,ii
         if (yy(k)<(l(i)/l(ii))) then
             y(k)=i
             exit
         end if
      end do
   end do

   !第二步：用y,r造出x,z，從這裡開始做20次
   do k=1,kk
   x=0;z=0
     !compute x(i)
      do i=1,ii
         do n=1,nn
            if (y(n)==i) then
                x(i)=x(i)+1
            end if
         end do
      end do

      !compute[(zij)]
      do n=1,nn
         do i=1,II
            do j=1,JJ
               if (y(n)==i .and. r(n)==j) then
                   z(i,j)=z(i,j)+1
               end if
            end do
         end do
      end do

      !compute a+x,b+z
      a=aa;b=bb
      do i=1,ii
         a(i)=a(i)+x(i)
         do j=1,jj
```

```
      b(i,j)=b(i,j)+z(i,j)
   end do
end do

!第三步：用a+x,b+z去造出u,G
!generate u,G
iseed2=456*k*m
!Monte Carlo 第一步
call rnset(iseed2)
call rnun(ii,p)       !generate p(3)，再以p去反推u
call rnun(ii*jj,pp)   !generate pp(3*7)，再以pp去反推G

do i=1,ii
   uu(i)=chiin(p(i),2*a(i)) !再把p當成機率反推chi-square的反函數值
   do j=1,jj
  if (b(i,j)==0) then
      GG(i,j)=0
  else
         GG(i,j)=chiin(pp(i,j),2*b(i,j))
      end if
   end do
end do

!Monte Carlo第二步
do i=1,ii
   u_k(k,i)=uu(i)/sum(uu)
   do j=1,jj
      G_row(j)=GG(i,j)
   end do
   do j=1,jj
      G_k(k,i,j)=GG(i,j)/sum(G_row)
   end do
end do

!第四步：用u,G去生成新的y,再回到第四步
!generate y
iseed3=789*k*m
call rnset(iseed3)
call rnun(nn,yy)
do n=1,nn
   do i=1,ii
      dd(i)=G_k(k,i,r(n))*u_k(k,i)
   end do

   do i=2,ii
      dd(i)=dd(i)+dd(i-1)
   end do

   do i=1,ii
      if (yy(n)<(dd(i)/dd(ii))) then
          y(n)=i
          exit
      end if
   end do
end do
```

```
      if (k==kk) then
         do i=1,ii
            u_out(m,i)=u_k(k,i)
            do j=1,jj
               G_out(m,i,j)=G_k(k,i,j)
            end do
         end do

      end if
   end do !做完 k 次以後再從第一步開始
end do !完成 m 次囉！

theta=0
do i=1,ii
   do m=1,mm
      theta(i)=theta(i)+u_out(m,i)
   end do
   theta(i)=theta(i)/mm
end do
t2=cpsec()
time_Gibbs=t2-t1

write(*,*) "Gibbs sampler"
write(*,"(a10,100f8.4)")"theta=",theta
write(*,*)" CPU-SEC：Gibbs sampler約",time_Gibbs,"秒"

write(50,*) "Gibbs sampler"
write(50,"(a10,100f8.4)")"theta=",theta
write(50,*)" CPU-SEC：Gibbs sampler約",time_Gibbs,"秒"
a=aa
b=bb
end subroutine Gibbs

!計算準貝氏法的估計值
subroutine quasiBayes(ii,jj,a,b,nn,r,u,time_quasi)
USE typedef
TYPE(PermuteList), POINTER :: p1Head,p1Move
integer::ii,jj,nn
integer::i,j,k
real a(ii),b(ii,jj),b_row(jj)
real aa(ii),bb(ii,jj)
real u(ii)
real G(ii,jj),G_row(jj)
real d(ii),dd(ii)
integer r(nn)
integer delta(ii,jj),total(jj)
real::t1,t2,time_quasi
aa=a
bb=b
t1=cpsec()
do i=1,II
   u(i)=a(i)/sum(a)
   do k=1,JJ
```

```
      G_row(k)=b(i,k)
   end do
   do j=1,JJ
      G(i,j)=b(i,j)/sum(G_row)
   end do
end do

!貝第一步：用u, G 來計算d(i)
do k=1,nn
   do i=1,II
      dd(i)=G(i,r(k))*u(i)
      !write(*,*)G(i,r(k)),u(i)
   end do
   do i=1,II
      d(i)=dd(i)/sum(dd)
   end do
   !write(*,*)"d=",d

  !貝第二步：計算出新的 a,b
   do i=1,II
      do j=1,JJ
         if (j==r(k)) then
             delta(i,j)=1
         else
             delta(i,j)=0
         end if
      end do
   end do
   do i=1,II
      a(i)=a(i)+d(i)
      do j=1,JJ
         b(i,j)=b(i,j)+d(i)*delta(i,j)
      end do
   end do

   !貝第三步：計算出新的 u,G
   do i=1,II
      u(i)=a(i)/sum(a)
      do j=1,JJ
         b_row(j)=b(i,j)
      end do
      do j=1,JJ
        G(i,j)=b(i,j)/sum(b_row)
      end do
   end do
   !回到貝第一步
end do
t2=cpsec()
time_quasi=t2-t1

WRITE(*,*)
write(*,"(a15)")"quasi_Bayes"
write(*,"(a8,100f8.4)")"theta=",u
write(*,*)"CPU-SEC : quasi-Bayes method 約",time_quasi,"秒"
write(*,*)
```

```
WRITE(50,*)
write(50,"(a15)")"quasi_Bayes"
write(50,"(a8,100f8.4)")"theta=",u
write(50,*)"CPU-SEC : quasi-Bayes method 約",time_quasi,"秒"
WRITE(50,*)
a=aa
b=bb
end subroutine quasiBayes


!利用d矩陣判斷貝氏法的Normal Constant
subroutine calculate(c,I,J,a,b,constant_a,constant_b1,constant_b2)
implicit none
integer k1,k2,m,I,J,n
real constant_a,constant_b1,constant_b2 !準備傳回去的結果
real a(I),b(I,J) !傳入的參數
integer c(I,J) !導入的參數
real c_a(I),c_b1(I,J),c_b2(I) !內部使用

c_a=1.0
c_b2=1.0
c_b1=1.0

do k1=1,I
   if (sum(c(k1,:)) /= 0) then
      do m=1,sum(c(k1,:))
         c_a(k1)=c_a(k1)*(a(k1)+real(m-1))
         c_b2(k1)=c_b2(k1)*(sum(b(k1,:))+real(m-1))
      end do
   end if
   constant_a=constant_a*c_a(k1)
   constant_b2=constant_b2*c_b2(k1)
   do k2=1,J
      if (c(k1,k2)/=0) then
         do m=1,c(k1,k2)
            c_b1(k1,k2)=c_b1(k1,k2)*(b(k1,k2)+real(m-1))
         end do
      end if
   constant_b1=constant_b1*c_b1(k1,k2)
   end do
end do
constant_b1=constant_b1/constant_b2
return
end subroutine calculate

!輸入所需要的參數及資料
program comparision
USE imsl
USE typedef
TYPE(PermuteList), POINTER :: p1Head,p1Move
integer ii,jj,i,j,n,ans,mm,kk
REAL t1,t2
integer, allocatable::r(:)
REAL, allocatable::a(:),b(:,:),expec(:),theta(:),u(:)
```

```
REAL, ALLOCATABLE::rel_Gibbs(:),rel_quasi(:)

OPEN(UNIT=50,FILE="result.txt")
100  write(*,*) "輸入：類別數(I):"
     read(*,*) ii
     write(*,*)"類別數 I=",ii
     write(*,*)"確定嗎？(Yes=1/No=2)"
     read(*,*)ans
     if(ans/=1) then
        goto 100
     end if

200  write(*,*) "輸入：報告共分多少種類(J)"
     read(*,*) jj
     write(*,*)"報告種類 J=",jj
     write(*,*)"確定嗎？(Yes=1/No=2)"
     read(*,*)ans
     if(ans/=1) then
        goto 200
     end if

     allocate(a(ii))
     ALLOCATE(expec(ii))
     ALLOCATE(rel_Gibbs(ii))
     ALLOCATE(rel_quasi(ii))
     ALLOCATE(theta(ii))
     ALLOCATE(u(ii))
     allocate(b(ii,jj))

300  write(*,*) "輸入prior a: "
     read(*,*) (a(i),i=1,ii)
     write(*,"(a7,100f8.2)")"a=",a
     write(*,*)"確定嗎？(Yes=1/No=2)"
     read(*,*)ans
     if(ans/=1) then
        goto 300
     end if

400  do i=1,ii
        write(*,*) "輸入prior bi*,i=",i," "
        read(*,*) (b(i,j),j=1,jj)
     end do
     WRITE(*,*) "b="
     do i=1,ii
        write(*,"(a7,2000f8.2)")"  ",(b(i,j),j=1,jj)
     end do
     write(*,*)"確定嗎？(Yes=1/No=2)"
     read(*,*)ans
     if(ans/=1) then
        goto 400
     end if

500  write(*,*) "輸入總樣本數 (n):"
     read(*,*) n
     WRITE(*,*) "總樣本數 n=", n
```

```
      write(*,*)"確定嗎？(Yes=1/No=2)"
      read(*,*)ans
      if(ans/=1) then
         goto 500
      end if

      allocate(r(n))

600   write(*,*) "輸入報告值："
      read(*,*) (r(i),i=1,n)
      write(*,*)"報告值 r=", r
      write(*,*)"確定嗎？(Yes=1/No=2)"
      read(*,*)ans
      if(ans/=1) then
         goto 600
      end if

700   write(*,*)" 輸入：Gibbs 取起始值的次數 (m)"
      read(*,*)mm
      write(*,*)"Gibbs 取起始值的次數 (m)=",mm
      write(*,*)"確定嗎？(Yes=1/No=2)"
      read(*,*)ans
      if(ans/=1) then
         goto 700
      end if

800   write(*,*)" 輸入：每次疊代的次數 (k)"
      read(*,*)kk
      write(*,*)"每次疊代的次數 (k)=",kk
      write(*,*)"確定嗎？(Yes=1/No=2)"
      read(*,*)ans
      if(ans/=1) then
         goto 800
      end if

      WRITE(*,*) "類別數 I= ",ii
      write(*,*) "報告種類 J=",jj
      write(*,"(a7,100f8.2)")"prior a= ",a
      WRITE(*,*) "prior b="
      do i=1,ii
         write(*,"(a7,2000f8.2)")"  ",(b(i,j),j=1,jj)
      end do
      WRITE(*,*) "總樣本數 n=", n
      write(*,*)"報告值 r=", r
      write(*,*)"Gibbs 取起始值的次數 (m)=",mm
      write(*,*)"每次疊代的次數 (k)=",kk
      write(*,*)"所有的資料都輸入正確嗎嗎？(Yes=1/No=2)"
      read(*,*)ans
      if(ans/=1) then
         WRITE(*,*) "所有資料重新輸入"
         goto 100
      end if

WRITE(50,*) "類別數 I= ",ii
write(50,*) "報告種類 J=",jj
```

```
write(50,"(a9,100f8.2)")"prior a= ",a
WRITE(50,*) "prior b="
do i=1,ii
   write(50,"(a7,2000f8.2)")"  ",(b(i,j),j=1,jj)
end do
WRITE(50,*) "總樣本數 n=", n
write(50,*)"報告值 r=", r
write(50,*)"Gibbs 取起始值的次數 (m)=",mm
write(50,*)"每次疊代的次數 (k)=",kk
WRITE(50,*) ""
write(50,*) "---------------------------------------"
WRITE(50,*) ""

t1=cpsec()
call Bayes(ii,jj,a,b,n,r,expec)
call permute(ii,jj,a,b,n,r,mm,kk,expec,theta,u)
t2=cpsec()
WRITE(*,*) "---------------------------------------"
WRITE(*,*) "此程式所花總時間爲 ",t2-t1, "秒"
WRITE(50,*) "此程式所花總時間爲 ",t2-t1, "秒"
end
```