

รายการอ้างอิง

ภาษาไทย

เสกสรร เกียรติสุไพบูลย์. เอกสารประกอบการสอนวิชาการจำลองแบบเชิงสถิติ. กรุงเทพมหานคร: จุฬาลงกรณ์มหาวิทยาลัย, 2552.

ภาษาอังกฤษ

Anant Chiarawongse, Seksan Kiatsupaibul, Sunti Tirapat and Benjamin Van Roy.

Portfolio Selection with Qualitative Input. (2009).(working paper)

Bélisle C.J.P., Romeijn H.E. and Smith R.L. Hit-and-Run Algorithms for Generating Multivariate Distributions. Mathematics of Operations Research 18 (1993) : 255-266.

Brooks S.P. and Gelman A.. General methods for monitoring convergence of iterative simulations. Journal of Computational and Graphical Statistics 7, 4 (1998) : 434-455.

Chen M.-H. and Schmeiser B.W. Performance of the Gibbs, Hit-and-Run, and Metropolis Samplers. Journal of Computational and Graphical Statistics 2, 3 (1993) : 251-272.

Flegal J.M., Haran M. and Jones G.L. Markov Chain Monte Carlo: Can We Trust the Third Significant Figure?. Statistical Science 23, 2 (2008) : 250-260.

Gelman A. and Rubin D.B. Inference from iterative simulation using multiple sequences. Statistical Science 7, 4 (1992) : 457-511.

Geman S. and Geman D. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1984) : 721-741.

Lovász L. Hit-and-Run Mixes Fast. Mathematical Programming 86 (1998) : 443-461.

Romeijn H.E. and Smith R.L. Simulated annealing for constrained global optimization. Journal of Global Optimization 5 (1994) : 101-126.

Smith R.L.. Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. Operations Research 32 (1984) : 1296-1308.

ภาคผนวก

ตัวอย่างการใช้โปรแกรม R ในการดำเนินงานวิจัย

การจำลองข้อมูลด้วยการสุ่มตัวอย่างแบบฮิตแอนด์รัน และคำนวณค่าครึ่งช่วงความเชื่อมั่นจากวิธี
ค่าเฉลี่ยกลุ่ม กรณี 2 มิติ และค่าสัมประสิทธิ์สหสัมพันธ์เป็น 0

```
##สร้างฟังก์ชัน hitandrunk
```

```
hitandrunk<-function(x0,dim,sigmainv){
  d0<-rnorm(dim,0,1)
  d<-d0/sqrt(sum(d0*d0))
  mutrunk<-(d%*%sigmainv%*%matrix(-x0,nrow=dim))
    /(d%*%sigmainv%*%matrix(d,nrow=dim))
  vartrunk<-1/(d%*%sigmainv%*%matrix(d,nrow=dim))
  sdtrunk<-sqrt(vartrunk)
  lambdabound<-(-x0[1:dim-1]-x0[2:dim])/(d[2:dim]-d[1:dim-1])
  pvec<-c(lambdabound[lambdabound >0],Inf)
  nvec<-c(lambdabound[lambdabound <0],-Inf)
  infval<-max(nvec)
  supval<-min(pvec)
  lamda<-qnorm((pnorm(supval,mean=mutrunk,sd=sdtrunk)
    -pnorm(infval,mean=mutrunk,sd=sdtrunk))*runif(1)
    +pnorm(infval,mean=mutrunk,sd=sdtrunk),mean=mutrunk,sd=sdtrunk)
  x0<-x0+lamda*d
  x0
}
```

```
maxround<-10000;      ##กำหนดจำนวนรอบในการจำลองข้อมูล 10,000 รอบ
burnin<-0.1*maxround  ##burn in 10 %
x<-c(0.41,-0.10)
dim<-length(x);
sigma<-matrix(0,dim,dim);diag(sigma)<-1;  ##กำหนดค่าสัมประสิทธิ์สหสัมพันธ์เป็น0
sigmainv<-solve(sigma)
```

```

xarray<-x;
for (i in 1:maxround){
  x<-hitandrun(x,dim,sigmainv);
  xarray<-rbind(xarray,x)
}
mub<-c()
##กำหนดจำนวนbatches=30
for(i in 1:30){
  mub<-rbind(mub, colMeans(xarray[((i-1)*((0.9*maxround)/30)
    +burnin+1):(i*((0.9*maxround)/30)+burnin),]))
}
##คำนวณค่าครึ่งช่วงความเชื่อมั่น
error<-qt(0.975,29)*sd(mub)/sqrt(30)
mu<-colMeans(mub)

```

การจำลองข้อมูลด้วยการสุ่มตัวอย่างแบบกิบส์ และคำนวณค่า MPSRF กรณี 2 มิติ และค่าสัมประสิทธิ์สหสัมพันธ์เป็น 0

#สร้างฟังก์ชัน gibbs

```

gibbs<-function(x0,dim,sigmainv){
  nchoose<-floor(dim*runif(1))+1
  mutrun<--sum((x0[-nchoose])*(sigmainv[nchoose,][-nchoose]))
  /sigmainv[nchoose,nchoose]
  vartrun<-1/(sigmainv[nchoose,nchoose])
  sdtrun<-sqrt(vartrun)
  infval<-ifelse(nchoose==dim,-Inf,x0[nchoose+1])
  supval<-ifelse(nchoose==1,Inf,x0[nchoose-1])
  x[nchoose]<-qnorm((pnorm(supval,mean=mutrun,sd=sdtrun)
    -pnorm(infval,mean=mutrun,sd=sdtrun))*runif(1)
    +pnorm(infval,mean=mutrun,sd=sdtrun),mean=mutrun,sd=sdtrun)
  x<-c(x0[-(nchoose:dim)],x[nchoose],x0[-(1:nchoose)])
}

```

```

        x0<-x
    }

xseed<-rnorm(2,0,1)
x<-xseed[c(order(-xseed))];
dim<-length(x);
sigma<-matrix(0,dim,dim);diag(sigma)<-1; ##กำหนดค่าสัมประสิทธิ์สหสัมพันธ์เป็น0
sigmainv<-solve(sigma)
xarray<-x;

gb<-function(maxround,burnin){
  for (i in 1:maxround){
    x<-gibbs(x,dim,sigmainv);
    xarray<-rbind(xarray,x)
  }
  xuse<-xarray[(burnin+1):maxround,]
}

##สร้างลูกโซ่ 5 ลูกโซ่ แต่ละลูกโซ่ burn in 50%
gb.draws1<-mcmc(gb(1000,500))
gb.draws2<-mcmc(gb(1000,500))
gb.draws3<-mcmc(gb(1000,500))
gb.draws4<-mcmc(gb(1000,500))
gb.draws5<-mcmc(gb(1000,500))

gb.list<-mcmc.list(list(gb.draws1,gb.draws2,gb.draws3,gb.draws4,gb.draws5))
## คำนวณค่า MPSRF จากฟังก์ชัน gelman.diag() ใน package coda
gelman.diag(gb.list)

```

ประวัติผู้เขียนวิทยานิพนธ์

นายพงศ์ศักดิ์ สถิตรุ่งพรชัย เกิดวันที่ 13 สิงหาคม 2524 สำเร็จการศึกษาระดับปริญญาบัณฑิต หลักสูตรวิทยาศาสตร์บัณฑิต (วท.บ.) จากภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2548 และเข้าศึกษาต่อในระดับปริญญาโทหลักสูตรวิทยาศาสตรมหาบัณฑิต ที่คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2551



