

SCRIPT MOD6S3D: CONVERGENCE PLOTS, PART II

INSTRUCTOR: KLAUS MOELTNER

COMPARE GS PERFORMANCE UNDER CORRELATED DATA

Orthogonal data.

```
R> n<-1000 #sample size
R> x1<-rep(1,n)
R> x2<-rnorm(n,-1.4,1)
R> x3<-rnorm(n,3,2)
R> btrue<-c(1.2,0.4,0.8)
R> X<-cbind(x1,x2,x3)
R> k<-ncol(X)
R> sig2true<-1.44
R> eps<-rnorm(n,0,sqrt(sig2true))
R> y<-X %*% btrue + eps
R> #
R> #
R> r1<-0 #burn-ins; here: keep all for diagnostics
R> r2<-5000 #keepers
R> R<-r1+r2
R> # of Gibbs Sampler
R> #
R> #PRIORS:
R> #####
R> #for beta:
R> mu0<-rep(0,k)
R> V0<-100*diag(k)
R> #for sig2:
R> v0<-0.5
R> tau0<-0.5
R> #
R> # STARTING VALUES
R> #####
R> bols<-solve(t(X) %*% X) %*% (t(X) %*% y)
R> e<-y-X%*%bols
R> s2<-(t(e)%*%e)/(n-k)
R> betastart<-bols
R> sig2start<-as.vector(s2)
R> #
R> #GIBBS SAMPLER
R> #####
R> betamat<-matrix(0,k,r2) #will collect draws of beta
R> sig2mat<-matrix(0,1,r2) #will collect draws of sig2
R> #
```

```

R> # Call for a progress bar to monitor progress of Gibbs Sampler
R> pb<-winProgressBar(title="progress bar", min=0,max=R,width= 300)
R> #
R> for (i in 1:R) {
  setWinProgressBar(pb,i,title=paste(round((i/R)*100,0),"% done"))
  #
  sig2draw<-sig2start
  # draw betas
  #####
  V1<-solve(solve(V0)+(1/sig2draw)*t(X)%*%X)
  mu1<-V1 %*% (solve(V0) %*% mu0 + (1/sig2draw)* t(X) %*% y)
  betadraw<-mvrnorm(n=1,mu1,V1)
  if (i>r1) {
    betamat[, (i-r1)]<-betadraw
  }
  # draw sig2
  #####
  v1<-(n+2*v0)/2
  tau1<-(1/2)*(t(y-X %*% betadraw) %*% (y-X %*% betadraw)+2*tau0)
  sig2draw<-rinvgamma(1,v1,scale=tau1)
  if (i>r1) {
    sig2mat[i-r1]<-sig2draw
  }
}
R> close (pb) #close progress bar
R> #
R> # add starting draw to GS chain. Let's focus on beta3
R> b3chain1<-c(betastart[3],betamat[3,])
R> sig2chain1<-c(sig2start,sig2mat)

```

Correlated data.

```

R> n<-1000 #sample size
R> x1<-rep(1,n)
R> x2<-rnorm(n,-1.4,1)
R> x3<-0.8*x2+rnorm(n,0,0.5)
R> btrue<-c(1.2,0.4,0.8)
R> X<-cbind(x1,x2,x3)
R> k<-ncol(X)
R> sig2true<-1.44
R> eps<-rnorm(n,0,sqrt(sig2true))
R> y<-X %*% btrue + eps
R> #
R> #TUNERS
R> #####
R> r1<-0 #burn-ins; here: keep all for diagnostics
R> r2<-5000 #keepers
R> R<-r1+r2
R> # of Gibbs Sampler
R> #
R> #PRIORS:
R> #####

```

```

R> #for beta:
R> mu0<-rep(0,k)
R> V0<-100*diag(k)
R> #for sig2:
R> v0<-0.5
R> tau0<-0.5
R> #
R> # STARTING VALUES
R> #####
R> bols<-solve(t(X) %*% X) %*% (t(X) %*% y)
R> e<-y-X%*%bols
R> s2<-(t(e)%*%e)/(n-k)
R> betastart<-bols
R> sig2start<-as.vector(s2)
R> #
R> #GIBBS SAMPLER
R> #####
R> betamat<-matrix(0,k,r2) #will collect draws of beta
R> sig2mat<-matrix(0,1,r2) #will collect draws of sig2
R> #
R> # Call for a progress bar to monitor progress of Gibbs Sampler
R> pb<-winProgressBar(title="progress bar", min=0,max=R,width= 300)
R> #
R> for (i in 1:R) {
  setWinProgressBar(pb,i,title=paste(round((i/R)*100,0),"% done"))
  #
  sig2draw<-sig2start
  # draw betas
  #####
  V1<-solve(solve(V0)+(1/sig2draw)*t(X)%*%X)
  mu1<-V1 %*% (solve(V0) %*% mu0 + (1/sig2draw)* t(X) %*% y)
  betadraw<-mvrnorm(n=1,mu1,V1)
  if (i>r1) {
    betamat[, (i-r1)]<-betadraw
  }
  # draw sig2
  #####
  v1<-(n+2*v0)/2
  tau1<-(1/2)*(t(y-X %*% betadraw) %*% (y-X %*% betadraw)+2*tau0)
  sig2draw<-rinvgamma(1,v1,scale=tau1)
  if (i>r1) {
    sig2mat[i-r1]<-sig2draw
  }
}
R> close (pb) #close progress bar
R> #
R> # add starting draw to GS chain. Let's focus on beta3
R> b3chain2<-c(betastart[3],betamat[3,])
R> sig2chain2<-c(sig2start,sig2mat)

R> proc.time()-tic

```

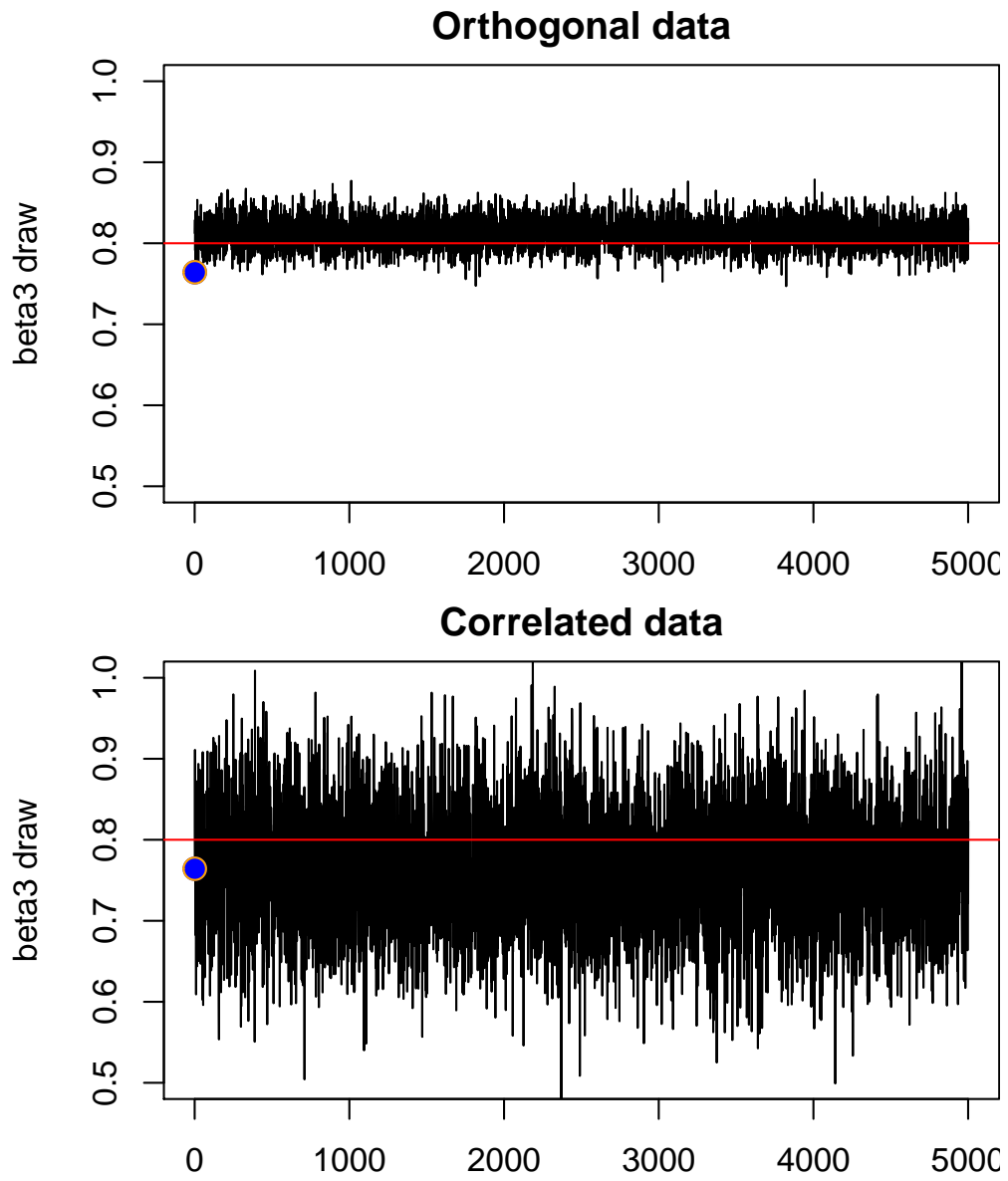


FIGURE 1. Convergence of β_3 for orthogonal and correlated data

```

user  system elapsed
7.99  2.80  12.27

```