LinearRegression

NarenSuri

December 1, 2016

```
require(MASS)
```

```
## Loading required package: MASS
```

```
# Loading the data
SwimDataLoading<-read.table("swim.dat")
SwimData<-as.matrix(SwimDataLoading)
seqData = seq(from=1,to=11,by=2)
# creating a BiWeekly sequence
ByWeekly<-matrix(seqData,6,1)
ByWeekly<-cbind(rep(1,6),ByWeekly)

# Initializing the prior values
# creating the fucntion to calcualte the Gibbs Sammpling
functionlinearRegOfSwim<-function(Data,ByWeekly){
  NumberOfFeatures<-2
  Beta0<-matrix(c(23,0),2,1)
  CoVarienceSigma0<-matrix(c(1,0,0,1),2,2)
  sigma0<-1
  nu0<-2
  Data<-as.vector(Data)

#initial values
  SamplingSize<-10000
  FinalBeta<-GibbsSigmaUpdate<-NULL
  RegressionModelResult<-lm(Data~ByWeekly[,-1])
  SummaryRegression<-summary(RegressionModelResult)$coef
  SSR<-sum((Data-(ByWeekly%*%SummaryRegression))^2)
  sampleVariance<-SSR/(length(Data)-NumberOfFeatures)
# Gibbbs iterative sampling starts here
# im sampling 10,000 samples here
for(i in 1:SamplingSize){
  temp =((t(ByWeekly)%*%ByWeekly)/sampleVariance)
  VarianceOfBeta<-solve(solve(CoVarienceSigma0)+temp)
  temp2= (t(ByWeekly)%*%Data)
  mu<-VarianceOfBeta%*%((solve(CoVarienceSigma0)%*%Beta0)+temp2/sampleVariance)
  SummaryRegression<-mvrnorm(1,mu,VarianceOfBeta)
  SSR<-sum((Data-(ByWeekly%*%SummaryRegression))^2)
  phi2<-rgamma(1,(nu0+length(Data))/2,((nu0*sigma0)+SSR)/2)
  sampleVariance<-(1/phi2)
  FinalBeta<-rbind(FinalBeta,SummaryRegression)
  GibbsSigmaUpdate<-rbind(GibbsSigmaUpdate,sampleVariance)

}
  # The resultant Gibbs
GibbsResultantDataframe<-data.frame(FinalBeta,GibbsSigmaUpdate)
colnames(GibbsResultantDataframe)<-c("BetaBeta0","BetaBeta1","GibbsSigmaUpdate")
return(GibbsResultantDataframe)
}
# Call the gibbs sampling to calcualte the Swimmers info
Swimmer1BetaCoefs<-functionlinearRegOfSwim(SwimData[1,],ByWeekly)
Swimmer2BetaCoefs<-functionlinearRegOfSwim(SwimData[2,],ByWeekly)
Swimmer3BetaCoefs<-functionlinearRegOfSwim(SwimData[3,],ByWeekly)
Swimmer4BetaCoefs<-functionlinearRegOfSwim(SwimData[4,],ByWeekly)

Swimmer1Means<-colMeans(Swimmer1BetaCoefs)
Swimmer2Means<-colMeans(Swimmer2BetaCoefs)
```

```
Swimmer3Means<-colMeans(Swimmer3BetaCoefs)
Swimmer4Means<-colMeans(Swimmer4BetaCoefs)
# printing the mean of the each swimmers calculated value
print(Swimmer1Means)
```

```
##        BetaBeta0        BetaBeta1 GibbsSigmaUpdate
##       23.1642964       -0.0403233        0.4689561
```

```
print(Swimmer2Means)
```

```
##        BetaBeta0        BetaBeta1 GibbsSigmaUpdate
##      23.10578943       0.03843639       0.47793640
```

```
print(Swimmer3Means)
```

```
##        BetaBeta0        BetaBeta1 GibbsSigmaUpdate
##      22.72575656       0.01046978       0.47284365
```

```
print(Swimmer4Means)
```

```
##        BetaBeta0        BetaBeta1 GibbsSigmaUpdate
##     23.572798997     -0.008088089      0.474590738
```

```
################################################################################################
#######

#b
# Sampling one sample from all the different 10,000 gibbs iterations Beta values
Ssigma <- sqrt(mean(Swimmer2BetaCoefs$GibbsSigmaUpdate))
BetasObtainedThroughGibbs<-data.matrix(Swimmer1BetaCoefs[,1:2])
DataX<-matrix(c(1,15),2,1)
PredictedMean<-BetasObtainedThroughGibbs%*%DataX
PredictedY1<-vector()
# will return the 10,000 new samples with those particular parameters
PredictedY1<-apply(PredictedMean, 1, function(xx) rnorm(1,xx,Ssigma) )


BetasObtainedThroughGibbs<-data.matrix(Swimmer2BetaCoefs[,1:2])
DataX<-matrix(c(1,15),2,1)

PredictedMean<-BetasObtainedThroughGibbs%*%DataX
PredictedY2<-vector()
PredictedY2<-apply(PredictedMean, 1, function(xx) rnorm(1,xx,Ssigma) )

BetasObtainedThroughGibbs<-data.matrix(Swimmer3BetaCoefs[,1:2])
DataX<-matrix(c(1,15),2,1)
PredictedMean<-BetasObtainedThroughGibbs%*%DataX
PredictedY3<-vector()
PredictedY3<-apply(PredictedMean, 1, function(xx) rnorm(1,xx,Ssigma) )

BetasObtainedThroughGibbs<-data.matrix(Swimmer4BetaCoefs[,1:2])
DataX<-matrix(c(1,15),2,1)
PredictedMean<-BetasObtainedThroughGibbs%*%DataX
PredictedY4<-vector()
PredictedY4<-apply(PredictedMean, 1, function(xx) rnorm(1,xx,Ssigma) )

################################################################################################
#########
# 3. Prediction / classification
# for each row we see who is the swimmer with max value? that particualr individaul repesents th
at particualr row in the classification
Y_predict<-data.frame(PredictedY1,PredictedY2,PredictedY3,PredictedY4)
# In all 10,000 checking how many times the Swimmer 1 is greater or Swimmer one got classified
swimmer1<-mean(Y_predict$PredictedY1>Y_predict$PredictedY2&Y_predict$PredictedY1>Y_predict$Predi
ctedY3&Y_predict$PredictedY1>Y_predict$PredictedY4)
swimmer2<-mean(Y_predict$PredictedY2>Y_predict$PredictedY1&Y_predict$PredictedY2>Y_predict$Predi
ctedY3&Y_predict$PredictedY2>Y_predict$PredictedY4)
swimmer3<-mean(Y_predict$PredictedY3>Y_predict$PredictedY1&Y_predict$PredictedY3>Y_predict$Predi
ctedY2&Y_predict$PredictedY3>Y_predict$PredictedY4)
swimmer4<-mean(Y_predict$PredictedY4>Y_predict$PredictedY1&Y_predict$PredictedY4>Y_predict$Predi
ctedY2&Y_predict$PredictedY4>Y_predict$PredictedY3)

print(swimmer1)
```

```
## [1] 0.0931
```

```
print(swimmer2)
```

```
## [1] 0.4302
```

```
print(swimmer3)
```

```
## [1] 0.1497
```

```
print(swimmer4)
```

```
## [1] 0.327
```

```
# See if sum of all these approx to 1
print(swimmer1+swimmer2+swimmer3+swimmer4)
```

```
## [1] 1
```