STAT_37810_HW2

Boxin

10/18/2018

Some preparation work before test

```
source("E:/Likelihood.R")
source("E:/slopevalues.R")
source("E:/prior.R")
source("E:/posterior.R")
source("E:/proposalfunction.R")
source("E:/run_metropolis_MCMC.R")
source("E:/MHsummary.R")

trueA <- 5
trueB <- 0
trueSd <- 10
sampleSize <- 31

# create independent x-values
x <- (-(sampleSize-1)/2):((sampleSize-1)/2)
# create dependent values according to ax + b + N(0,sd)
y <-  trueA * x + trueB + rnorm(n=sampleSize,mean=0,sd=trueSd)
```

Let's now compare different iteration times, we first print out the results in each loop

```
# comparison between different iteration numbers
compare_outcomes <- function(iteration.numbers){
  num.of.iteration <- length(iteration.numbers)  # record the number of different iteration times
  loop.num <- 10

  record.compare.mean <- array(dim=c(loop.num, num.of.iteration))
  record.compare.std <- array(dim=c(loop.num, num.of.iteration))

  for (i in c(1:loop.num)){
    startvalue <- c(1:3)
    startvalue[1] <- runif(n=1, min=0, max=10)
    startvalue[2] <- rnorm(n=1, sd = 5)
    startvalue[3] <- runif(n=1, min=0, max=30)

    print(paste("This is the", i, "loop"))

    for (j in c(1:num.of.iteration)){
      chain = run_metropolis_MCMC(startvalue, iteration.numbers[j])
      a.mean <- mean(chain[, 1])
      record.compare.mean[i, j] <- a.mean
      a.std <- sd(chain[, 1])
      record.compare.std[i, j] <- a.std
      print(paste("For iteration times as:", iteration.numbers[j], "the mean is:", a.mean,
                  "The std is:", a.std))
    }
  }
  record.compare <- array(dim=c(loop.num, num.of.iteration))

  for (i in c(1:loop.num)){
    for (j in c(1:num.of.iteration)){
      record.compare[i, j] = paste("mean:", round(record.compare.mean[i, j], 3),
                                   "std:", round(record.compare.std[i, j], 3))
    }
  }

  record.compare <- data.frame(record.compare)

  for (j in c(1:num.of.iteration)){
    colnames(record.compare)[j] <-  paste("iteration:", iteration.numbers[j])
  }

  for (j in c(1:num.of.iteration)){
    colnames(record.compare)[j] <-  paste("iteration:", iteration.numbers[j])
  }

  for (i in c(1:loop.num)){
    rownames(record.compare)[i] <- paste("loop ", i)
  }

  return(record.compare)
}

iteration.numbers <- c(1000, 10000, 100000)
m.compare.out <- compare_outcomes(iteration.numbers)
```

```
## [1] "This is the 1 loop"
## [1] "For iteration times as: 1000 the mean is: 4.67540543771507 The std is: 0.54419247808401"
## [1] "For iteration times as: 10000 the mean is: 4.56024298164523 The std is: 0.29263306803807"
## [1] "For iteration times as: 1e+05 the mean is: 4.56532920808802 The std is: 0.226428201202933"
## [1] "This is the 2 loop"
## [1] "For iteration times as: 1000 the mean is: 4.49996250463098 The std is: 0.414908761233062"
## [1] "For iteration times as: 10000 the mean is: 4.52731107792138 The std is: 0.261771112975661"
## [1] "For iteration times as: 1e+05 the mean is: 4.55515043728047 The std is: 0.228013750236193"
## [1] "This is the 3 loop"
## [1] "For iteration times as: 1000 the mean is: 4.52694418402824 The std is: 0.337193333535687"
## [1] "For iteration times as: 10000 the mean is: 4.55459920091779 The std is: 0.241980080630027"
## [1] "For iteration times as: 1e+05 the mean is: 4.5555331157737 The std is: 0.222799166523802"
## [1] "This is the 4 loop"
## [1] "For iteration times as: 1000 the mean is: 4.94028576966752 The std is: 1.12566735225914"
## [1] "For iteration times as: 10000 the mean is: 4.61188581904433 The std is: 0.412472279942246"
## [1] "For iteration times as: 1e+05 the mean is: 4.56214297444578 The std is: 0.254361196646297"
## [1] "This is the 5 loop"
## [1] "For iteration times as: 1000 the mean is: 4.61739867033079 The std is: 0.217800964566418"
## [1] "For iteration times as: 10000 the mean is: 4.57757816419117 The std is: 0.23325889038722"
## [1] "For iteration times as: 1e+05 the mean is: 4.55705943492219 The std is: 0.224213262950296"
## [1] "This is the 6 loop"
## [1] "For iteration times as: 1000 the mean is: 4.55374258329566 The std is: 0.490825367348866"
## [1] "For iteration times as: 10000 the mean is: 4.57755485109151 The std is: 0.278713624766187"
## [1] "For iteration times as: 1e+05 the mean is: 4.5562286104056 The std is: 0.227477714234521"
## [1] "This is the 7 loop"
## [1] "For iteration times as: 1000 the mean is: 4.12065383781514 The std is: 1.0041952588898"
## [1] "For iteration times as: 10000 the mean is: 4.52960055215659 The std is: 0.394680679065199"
## [1] "For iteration times as: 1e+05 the mean is: 4.55405508008708 The std is: 0.252998637797392"
## [1] "This is the 8 loop"
## [1] "For iteration times as: 1000 the mean is: 4.51576663492788 The std is: 0.304084853513676"
## [1] "For iteration times as: 10000 the mean is: 4.54793954995577 The std is: 0.23479302342204"
## [1] "For iteration times as: 1e+05 the mean is: 4.55698453267074 The std is: 0.223908787626111"
## [1] "This is the 9 loop"
## [1] "For iteration times as: 1000 the mean is: 4.40400134252617 The std is: 0.431732020573833"
## [1] "For iteration times as: 10000 the mean is: 4.53592532495927 The std is: 0.240055034696123"
## [1] "For iteration times as: 1e+05 the mean is: 4.56562606663568 The std is: 0.229649902377944"
## [1] "This is the 10 loop"
## [1] "For iteration times as: 1000 the mean is: 4.40046373740028 The std is: 0.528567835568634"
## [1] "For iteration times as: 10000 the mean is: 4.54891291432418 The std is: 0.287912127641201"
## [1] "For iteration times as: 1e+05 the mean is: 4.55599761857281 The std is: 0.22890670734211"
```

Now Let’s see the result in the table

```
m.compare.out
```

```
##                 iteration: 1000       iteration: 10000
## loop  1  mean: 4.675 std: 0.544  mean: 4.56 std: 0.293
## loop  2    mean: 4.5 std: 0.415 mean: 4.527 std: 0.262
## loop  3  mean: 4.527 std: 0.337 mean: 4.555 std: 0.242
## loop  4   mean: 4.94 std: 1.126 mean: 4.612 std: 0.412
## loop  5  mean: 4.617 std: 0.218 mean: 4.578 std: 0.233
## loop  6  mean: 4.554 std: 0.491 mean: 4.578 std: 0.279
## loop  7  mean: 4.121 std: 1.004  mean: 4.53 std: 0.395
## loop  8  mean: 4.516 std: 0.304 mean: 4.548 std: 0.235
## loop  9  mean: 4.404 std: 0.432  mean: 4.536 std: 0.24
## loop  10   mean: 4.4 std: 0.529 mean: 4.549 std: 0.288
##                iteration: 1e+05
## loop  1  mean: 4.565 std: 0.226
## loop  2  mean: 4.555 std: 0.228
## loop  3  mean: 4.556 std: 0.223
## loop  4  mean: 4.562 std: 0.254
## loop  5  mean: 4.557 std: 0.224
## loop  6  mean: 4.556 std: 0.227
## loop  7  mean: 4.554 std: 0.253
## loop  8  mean: 4.557 std: 0.224
## loop  9   mean: 4.566 std: 0.23
## loop  10 mean: 4.556 std: 0.229
```