HW3

Michael Miller

2022-11-06

Load Libraries

```
library(bench)
library(devtools)
```

```
## Loading required package: usethis
```

```
devtools::install_github("mikemiller442/fastHierarchicalReg")
```

```
## Downloading GitHub repo mikemiller442/fastHierarchicalReg@HEAD
```

```
## curl     (4.3.2  -> 4.3.3 ) [CRAN]
## vctrs    (0.5.0  -> 0.5.1 ) [CRAN]
## ps       (1.7.1  -> 1.7.2 ) [CRAN]
## processx (3.7.0  -> 3.8.0 ) [CRAN]
## digest   (0.6.29 -> 0.6.30) [CRAN]
## xfun     (0.32   -> 0.35  ) [CRAN]
## purrr    (0.3.4  -> 0.3.5 ) [CRAN]
## knitr    (1.39   -> 1.41  ) [CRAN]
## sass     (0.4.2  -> 0.4.3 ) [CRAN]
## httpuv   (1.6.5  -> 1.6.6 ) [CRAN]
```

```
## Installing 10 packages: curl, vctrs, ps, processx, digest, xfun, purrr, knitr, sass, httpuv
```

```
## Warning: package 'knitr' is in use and will not be installed
```

```
## Installing packages into 'C:/Users/mikemillertech/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
##
##   There are binary versions available but the source versions are later:
##       binary source needs_compilation
## vctrs  0.5.0  0.5.1              TRUE
## xfun    0.34   0.35              TRUE
## sass   0.4.2  0.4.3              TRUE
##
## package 'curl' successfully unpacked and MD5 sums checked
## Warning: cannot remove prior installation of package 'curl'

## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:\Users\mikemillertech\AppData\Local\R\win-
## library\4.2\00LOCK\curl\libs\x64\curl.dll to C:
## \Users\mikemillertech\AppData\Local\R\win-library\4.2\curl\libs\x64\curl.dll:
## Permission denied

## Warning: restored 'curl'

## package 'ps' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'ps'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying C:
## \Users\mikemillertech\AppData\Local\R\win-library\4.2\00LOCK\ps\libs\x64\ps.dll
## to C:\Users\mikemillertech\AppData\Local\R\win-library\4.2\ps\libs\x64\ps.dll:
## Permission denied

## Warning: restored 'ps'

## package 'processx' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'processx'

## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:\Users\mikemillertech\AppData\Local\R\win-
## library\4.2\00LOCK\processx\libs\x64\processx.dll
## to C:\Users\mikemillertech\AppData\Local\R\win-
## library\4.2\processx\libs\x64\processx.dll: Permission denied

## Warning: restored 'processx'

## package 'digest' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'digest'

## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:\Users\mikemillertech\AppData\Local\R\win-
## library\4.2\00LOCK\digest\libs\x64\digest.dll to C:
## \Users\mikemillertech\AppData\Local\R\win-
## library\4.2\digest\libs\x64\digest.dll: Permission denied

## Warning: restored 'digest'

## package 'purrr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'purrr'
## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:\Users\mikemillertech\AppData\Local\R\win-
## library\4.2\00LOCK\purrr\libs\x64\purrr.dll to C:
## \Users\mikemillertech\AppData\Local\R\win-library\4.2\purrr\libs\x64\purrr.dll:
## Permission denied

## Warning: restored 'purrr'

## package 'httpuv' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'httpuv'

## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:\Users\mikemillertech\AppData\Local\R\win-
## library\4.2\00LOCK\httpuv\libs\x64\httpuv.dll to C:
## \Users\mikemillertech\AppData\Local\R\win-
## library\4.2\httpuv\libs\x64\httpuv.dll: Permission denied

## Warning: restored 'httpuv'

##
## The downloaded binary packages are in
##  C:\Users\mikemillertech\AppData\Local\Temp\RtmpsbPAn8\downloaded_packages

## installing the source packages 'vctrs', 'xfun', 'sass'

## Warning in i.p(...): installation of package 'vctrs' had non-zero exit status

## Warning in i.p(...): installation of package 'xfun' had non-zero exit status

##          checking for file 'C:\Users\mikemillertech\AppData\Local\Temp\RtmpsbPAn8\remotes67f44ebc50\
##       -  preparing 'fastHierarchicalReg':
##    checking DESCRIPTION meta-information ...  v  checking DESCRIPTION meta-information
##       -  checking for LF line-endings in source and make files and shell scripts
##   -  checking for empty or unneeded directories
##    Omitted 'LazyData' from DESCRIPTION
##       -  building 'fastHierarchicalReg_0.1.0.tar.gz'
##
##

## Installing package into 'C:/Users/mikemillertech/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
devtools::install_github("kangjian2016/fastBayesReg")
```

```
## Skipping install of 'fastBayesReg' from a github remote, the SHA1 (5ffa15cd) has not changed since la
##   Use `force = TRUE` to force installation
```

```
library(fastHierarchicalReg)
```

```
## Loading required package: parallel
```

```
## Loading required package: foreach
```

```
## Loading required package: roxygen2
```

```
library(fastBayesReg)
```

```
## Loading required package: Rcpp
```

```
## Loading required package: RcppArmadillo
```

```
## Loading required package: glmnet
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```

```
## Loading required package: horseshoe
```

```
## Loading required package: pgdraw
```

Simulate Data 1

```
n <- 10000   # subjects
numBeta <- 100   # covariates
betaSD <- 0.75   # standard deviation of the randomly sampled covariates
XSD <- 0.5   # standard deviation of the generated features
errorSD <- 2.0   # regression standard deviation
e <- rnorm(n, mean = 0, sd = errorSD)
beta <- rnorm(numBeta, mean = 0, sd = betaSD*errorSD)
Z <- matrix(NA, nrow = n, ncol = numBeta)
for (i in 1:ncol(Z)) {
  Z[,i] <- rnorm(n, mean = 0, sd = XSD)
}

y <- Z %*% beta + e
output <- y
```

Run Gibbs Sampler

```r
X <- Z
testX <- Z
resp <- output
testResp <- output

numEpochs <- 10000
numDiscard <- 2000
numChains <- 4
numCores <- 8
lambdaSqPrior <- 1.0
regVarPrior <- 1.0

res <- fastHierarchicalReg::linRegGibbsProcessed(X = X,
                                                 testX = testX,
                                                 Y = resp,
                                                 testY = testResp,
                                                 lambdaSqPrior = lambdaSqPrior,
                                                 regVarPrior = regVarPrior,
                                                 numEpochs = numEpochs,
                                                 numDiscard = numDiscard,
                                                 numChains = numChains,
                                                 numCores = numCores)
```

```
## socket cluster with 8 nodes on host 'localhost'
```

Compare means between model output and true values

```r
# Comparing beta
paramBeta <- as.numeric(res$postMeanList$beta)
all.equal(beta,paramBeta)
```

```
## [1] "Mean relative difference: 0.0214177"
```

```r
# Comparing lambda
paramLambda <- sqrt(as.numeric(res$postMeanList$lambdaSq))
all.equal(betaSD,paramLambda)
```

```
## [1] "Mean relative difference: 0.1697782"
```

```r
# Comparing sigma
paramSigma <- sqrt(as.numeric(res$postMeanList$regVar))
all.equal(errorSD,paramSigma)
```

```
## [1] "Mean relative difference: 0.003511942"
```

Rhat Convergence Diagnostic - Check Convergence to 1.0

```r
all.equal(as.numeric(unlist(res$RhatList)),rep(1.0,length(unlist(res$RhatList))))
```

```
## [1] "Mean relative difference: 3.948415e-05"
```

Simulate Data 2

```r
n <- 10000
numBeta <- 5
betaSD <- 0.75
XSD <- 0.5
errorSD <- 2.0
e <- rnorm(n, mean = 0, sd = errorSD)
beta <- rnorm(numBeta, mean = 0, sd = betaSD*errorSD)
Z <- matrix(NA, nrow = n, ncol = numBeta)
for (i in 1:ncol(Z)) {
  Z[,i] <- rnorm(n, mean = 0, sd = XSD)
}

y <- Z %*% beta + e
output <- y
```

Run Gibbs Sampler For Methods Packages

```r
X <- Z
testX <- Z
resp <- output
testResp <- output

numEpochs <- 4000
numDiscard <- 2000
numChains <- 4
lambdaSqPrior <- 1.0
regVarPrior <- 1.0

res <- fastHierarchicalReg::linRegGibbsProcessed(X = X,
                                                 testX = testX,
                                                 Y = resp,
                                                 testY = testResp,
                                                 lambdaSqPrior = lambdaSqPrior,
                                                 regVarPrior = regVarPrior,
                                                 numEpochs = numEpochs,
                                                 numDiscard = numDiscard,
                                                 numChains = numChains,
                                                 numCores = numCores)
```

```
## socket cluster with 8 nodes on host ’localhost’
```

```r
numEpochs <- 10000
numDiscard <- 2000

resKJ <- fastBayesReg::fast_normal_lm(y = resp,
                                      X = X,
                                      mcmc_sample = numEpochs,
                                      burnin = numDiscard,
                                      a_sigma = 0.1,
                                      b_sigma = 0.1)
```

Compare regression coefficient posterior means between model output and Stan

```r
# Comparing posterior means
kjBeta <- as.numeric(resKJ$post_mean$betacoef)
paramBeta <- as.numeric(res$postMeanList$beta)
all.equal(kjBeta,paramBeta)
```

```
## [1] "Mean relative difference: 0.002954919"
```

Benchmark High Dimensional Example

```r
n <- 1000
numBeta <- 500
betaSD <- 0.05
XSD <- 0.5
errorSD <- 2.0
e <- rnorm(n, mean = 0, sd = errorSD)
beta <- rnorm(numBeta, mean = 0, sd = betaSD*errorSD)
Z <- matrix(NA, nrow = n, ncol = numBeta)
for (i in 1:ncol(Z)) {
  Z[,i] <- rnorm(n, mean = 0, sd = XSD)
}

y <- Z %*% beta + e
output <- y

funMod <- function() {
  X <- Z
  testX <- Z
  resp <- output
  testResp <- output

  numEpochs <- 4000
  numDiscard <- 2000
  lambdaSqPrior <- 1.0
  regVarPrior <- 1.0

  res <- fastHierarchicalReg::linRegGibbs(X = X,
                                          testX = testX,
                                          Y = resp,
                                          testY = testResp,
                                          numEpochs = numEpochs,
                                          regVarPrior = regVarPrior,
                                          lambdaSqPrior = lambdaSqPrior)
  postBeta <- res$coefBeta[,(numDiscard+2):(numEpochs+1)]
  paramBeta <- as.numeric(rowMeans(postBeta))
  return(paramBeta)
}

funKJ <- function() {
  numEpochs <- 4000
  numDiscard <- 2000
  resKJ <- fastBayesReg::fast_normal_lm(y = resp,
                                        X = X,
                                        mcmc_sample = numEpochs,
                                        burnin = numDiscard,
                                        a_sigma = 0.1,
                                        b_sigma = 0.1)
  kjBeta <- as.numeric(resKJ$post_mean$betacoef)
  return(kjBeta)
}

benchMarkRes <- bench::mark(funMod(),
                            funKJ(),
                            iterations = 2,
                            check = FALSE)
```

```
## Warning: Some expressions had a GC in every iteration; so filtering is disabled.
```

```r
benchMarkTable <- benchMarkRes[c("expression", "min", "median","mem_alloc","n_gc")]
knitr::kable(benchMarkTable)
```

expression	min	median	mem_alloc	n_gc
funMod()	24.4s	26.3s	22.8GB	846
funKJ()	57.8ms	59.4ms	299.6KB	0