Example 5: Parallel Computing

Setting Up

First load the data and R packages you need into the R workspace.

```
load("PlayData_for_Examples.RData") ### load the data for the examples
source("LoadFunctions.R") #### Source the functions required to run the block bootstrap


#Install packages if required
if(!require(pROC)) { install.packages("pROC", repos = "http://cran.us.r-project.org"); require(pROC) }

library(pROC)
```

Motivation

As in Example 4, imagine we want to know the standard error of the AUC (area under the ROC curve) of our model. However we wish to run the bootstrap in parallel on a cluster computer. For NBoot = 500, we could run 10 batches, each with 50 bootstrap resamples. We then assemble the results using the function **CombineBatchesofBootstraps**.

To use **CombineBatchesofBootstraps**, each batch should contain an equal number of bootstrap resamples.

```
set.seed(42)

fit1 = glm (PresAbsresponse ~ temperature, data = dat, family="binomial")
```

Coding StatFunction argument

As in Example 4, define getAUC a function which will be the *StatFunction* argument to *BlockBootApply*.

```
GetAUC = function(dat){
fit1 = glm ("PresAbsresponse~temperature", data=dat, family="binomial")
AUC = auc(response=dat$PresAbsresponse, predictor= predict(fit1))
;
AUC
}
```

Run Block Bootstrap

First the user should create a folder called e.g. LookupTables in their R working directory. As all the batches of bootstraps use the same site coordinates they all share the same lookup tables.

```
lookuptables.folderpathname = "LookupTables/"
```

```
batchID = 1
assign (paste0("Results.example4.batch", batchID ),
        BlockBootApply (x = x ,y = y ,block_Ls = c(0,0.05,0.1,0.2), Grid_space = 0.01 ,
                        dat = dat, Stat.function = GetAUC, tuning_block_length = 2,
                        NBoot = 250, method.block.length.select = "Lahiri",type="SE",
                        lookuptables.folderpath=lookuptables.folderpathname))
```

```
batchID = 2
assign (paste0("Results.example4.batch", batchID ),
        BlockBootApply (x = x ,y = y ,block_Ls = c(0,0.05,0.1,0.2), Grid_space = 0.01 ,
                        dat = dat, Stat.function = GetAUC, tuning_block_length = 2,
                        NBoot = 250, method.block.length.select = "Lahiri",type="SE",
                        lookuptables.folderpath=lookuptables.folderpathname))
```

```
batchIDs=1:2

list_of_boot_batches = lapply(paste0("Results.example4.batch", batchIDs), get)

Results.example4.all = CombineBatchesofBootstraps(list_of_boot_batches=
list_of_boot_batches )
```

Check sensitivity to tuning parameter. See that all columns are minimised by block length 0.1, so tuning block length wasn't terribly imporant here

```
Results.example4.all$Empirical.MSE
```

```
##                 tuninglength0 tuninglength0.05 tuninglength0.1
## blocklength0     0.0003802906       0.02173633      0.08486327
## blocklength0.05  0.0093301256       0.02342151      0.07875828
## blocklength0.1   0.0266339887       0.03785125      0.09010597
## blocklength0.2   0.0382523887       0.08122859      0.16753968
##                 tuninglength0.2
## blocklength0          0.1325652
## blocklength0.05       0.1225259
## blocklength0.1        0.1323171
## blocklength0.2        0.2269502
```

The block size parameter chosen is:

```
Results.example4.all$Lahiri_block_size
```

```
## [1] 0
```

This is the standard error (estimated via block bootstrap to account for spatial autocorrelation) of AUC

```
Results.example4.all$SE.estimate
```

```
## [1] 0.0194651
```