Some Examples of our Sorting Procedure

Contents

An example for our parametric LiNGAM sorting procedure ($p = 5$)

Compiling necessary code for this example.

The below code chunk assumes that the helper function `R` file and algorithm source code are in the same directory as this `R` markdown file.

```r
# helper functions for generating LiNGAM data and checking correctness of order
source("source/scorelingam/helperFunctions.r")
require(Rcpp)
```

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

```r
# compile sorting algorithm source, and allow it to be used in R
Rcpp::sourceCpp("source/scorelingam/source.cpp")
```

Generate data from a LiNGAM

LiNGAM Parameters

```r
p <- 5;numRoots <- 1; numParentsMin <- 1; numParentsMax <- 2
scaleParam <- runif(n=p,min=0.5,max=1.2)
causalOrder <- 5:1
lingamParams <- rand.wtd.adj.mat(p=p,num.roots=numRoots,pa.min=numParentsMin,pa.max=numParentsMax,
                    pa.wt.min = 0.25,pa.wt.max = 0.9,prob.pos = 0.5,perm = causalOrder)
# weighted adjacency matrix
print(lingamParams$B)
```

```
##            [,1]       [,2]       [,3]      [,4] [,5]
## [1,]  0.0000000  0.0000000  0.0000000 0.0000000    0
## [2,]  0.0000000  0.0000000  0.0000000 0.0000000    0
## [3,]  0.4731721  0.0000000  0.0000000 0.0000000    0
## [4,]  0.0000000  0.7084167 -0.3461217 0.0000000    0
## [5,] -0.4955223 -0.5259022 -0.7714625 0.4824493    0
```

Data matrix with $n = 5000$

When generating the data matrix, the possible options for the family argument are ‘laplace’, ‘logistic’, and ‘t’, in which case the additional argument df is needed (df=10 is the default)

```
n <- 5000
X <- genSCM.data(B=lingamParams$B,shape = scaleParam,perm = causalOrder,n = n,family = 'laplace')
dim(X)
```

```
## [1] 5000    5
```

Obtain an estimated ordering and check proportion of parents sorted after a child

When estimating a topological ordering for the LiNGAM, the possible options for the family argument are ‘laplace’, ‘logistic’, and ‘t’, in which case the additional argument df is needed (df=10 is the default)

```
# neighborhoods specified to be all other nodes
mbhat <- lapply(1:ncol(X),function(j){(1:ncol(X))[-j]})
(estOrder <- sort_llrmbCPP(Xmat=X,mb=mbhat,numUpdates=ncol(X),family='laplace'))
```

```
##      [,1]
## [1,]    5
## [2,]    4
## [3,]    3
## [4,]    1
## [5,]    2
```

```
# check errors (ideally close to zero)
check.valid.sort(estOrder=estOrder,M=t(lingamParams$B))
```

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

Check accuracy of estimated weighted adjacency matrix

```
(Bhat <- getParams(X=X,mbhat=mbhat,ordering=estOrder)$Bhat)
```

```
##             [,1]         [,2]       [,3]      [,4] [,5]
## [1,]  0.00000000 -0.005090228  0.0000000 0.0000000    0
## [2,]  0.00000000  0.000000000  0.0000000 0.0000000    0
## [3,]  0.49642798 -0.002582958  0.0000000 0.0000000    0
## [4,] -0.03517106  0.713507708 -0.3516732 0.0000000    0
## [5,] -0.45187722 -0.541288295 -0.7849962 0.4834605    0
```

```
# maximum entry-wise difference in absolute value
norm(x=lingamParams$B-Bhat,type = 'i')
```

```
## [1] 0.07357604
```

A higher dimensional example ($p = 10,000$)

Generate data from a LiNGAM

LiNGAM Parameters

```
p <- 10000;numRoots <- 100; numParentsMin <- 1; numParentsMax <- 2
scaleParam <- runif(n=p,min=0.5,max=1.2)
causalOrder <- c(seq(2,p,by=2),seq(1,p-1,by=2))
```

```
lingamParams <- rand.wtd.adj.mat(p=p,num.roots=numRoots,pa.min=numParentsMin,pa.max=numParentsMax,
                    pa.wt.min = 0.25,pa.wt.max = 0.9,prob.pos = 0.5,perm = causalOrder)
```

Data matrix with $n = 5000$

When generating the data matrix, the possible options for the family argument are ‘laplace’, ‘logistic’, and ‘t’, in which case the additional argument df is needed (df=10 is the default)

```
n <- 5000
X <- genSCM.data(B=lingamParams$B,shape = scaleParam,perm = causalOrder,n = n,family = 'laplace')
dim(X)
```

```
## [1]  5000 10000
```

Obtain an estimated ordering and time the algorithm

When estimating a topological ordering for the LiNGAM, the possible options for the family argument are ‘laplace’, ‘logistic’, and ‘t’, in which case the additional argument df is needed (df=10 is the default)

```
# neighborhoods specified to be true markov blankets
mbhat <- moralize(lingamParams$B)
#
start <- Sys.time()
estOrder <- sort_llrmbCPP(Xmat=X,mb=mbhat,numUpdates=10,family='laplace')
end <- Sys.time()
difftime(end,start,units='mins')
```

```
## Time difference of 1.54217 mins
```

Check accuracy of estimated ordering

```
# check sorting errors (ideally close to zero)
# proportion of parents sorted after child
check.valid.sort(estOrder=estOrder,M=t(lingamParams$B))
```

```
## [1] 0.08199584
```

“‘