Supplementary File "Multi-class Vector AutoRegressive Models for Multi-store Sales Data"

Ines Wilms*, Luca Barbaglia, Christophe Croux

1 Code to reproduce results for multi-store sales application

1.1 Configuration

- Software: `R` version 3.3.1
- Platform: x86_64-w64-mingw32/x64 (64-bit)
- Packages: `corrplot` version 0.77, `doRNG` version 1.6.6, `doSNOW` version 1.0.14, `foreach` version 1.4.3, `ggplot2` version 2.1.0, `igraph` version 1.0.1, `JGL` version 2.3 , `magic` version 1.5-6, `MASS` version 7.3-45, `Matrix` version 1.2-6, `MTS` version 0.33, `plyr` version 1.8.4, `rARPACK` version 0.11-0, `scales` version 0.4.0

1.2 Access the code

Download the zip-file "Code.zip" from the publisher's website. When extracting this zip-file, the directory "Code" will be created. The directory "Code" consists of

- The directory "Data" containing the multi-store sales data set, analyzed in the paper in Section 5. These data from Dominick's Finer Foods are publicly available at https://research.chicagobooth.edu/kilts/marketing-databases/dominicks.
- The `R`-script "Application.R" containing all `R`-code to replicate the results for the multi-store sales application.
- The directory "Functions" containing three `R`-scripts (.R files) with all necessary functions to execute the code in the "Application.R" `R`-script. The main function to obtain the multi-class VAR estimator is in the `R`-script "Multiclass_VAR.R".

*ines.wilms@kuleuven.be

1.3 Replicate results

We give the code to replicate the results discussed in Section 5 of the paper. Set the working directory in `R` to the directory "Code" (In `R`: File → Change dir... In `RStudio`: Session → Set Working Directory → Choose Directory). Open the `R`-script "Application.R". Below, we give step-by-step instructions on how to use the code to replicate the application results.

In the `R`-script "Application.R", we first install and/or load all necessary packages, source all functions, and import the multi-store sales data set "Data.RData". The object "Data.RData" is an $T \times J \times K$ array containing $J = 3 \times G = 15$ time series of $G = 5$ product categories, each of length $T = 76$ and for $K = 15$ stores.

To obtain the multi-class VAR estimator for the multi-store sales data set "Data", we use the function `MultiClass_VAR`:

```
FIT <- MultiClass_VAR(Data, P=1)
```

This function works for any $T \times J \times K$ "Data" array and only requires the user to specify the order P of the multi-class VAR model. We did select the order of the VAR using the BIC, which gave $P = 1$. colorblack Note that on an Intel Core i7-3720QM @ 2.60GHz machine, around 15 minutes are needed to obtain the estimator. We admit that this is a high computational cost.

To obtain the significant parameter estimates based on the bootstrap procedure of Appendix B with `Nboot` = 1000 bootstrap runs, use the function `bootSE`:

```
set.seed(20170516)
bootFIT <- bootSE(FITOBJECT=FIT, Nboot=1000, parallell=TRUE)
```

Setting the argument `parallell` equal to `TRUE` ensures the use of parallel computing. colorblack Note that we fix the seed of the pseudo-random generator used in the bootstrap exercise in order to guarantee full reproducibility of the results.

Figures

Figure 1 panels (a)-(c), page 11, containing the store clustering are obtained with the function `CLUSTER.PLOT`:

```
J <- dim(Data)[2]
K <- dim(Data)[3]
G <- 5
catnames <- c("BER", "BJC", "RFJ", "FRJ", "SDR")
```

```
CLUSTER.PLOT(i=which(catnames=="BER"), j=which(catnames=="RFJ"), bootFIT=bootFIT,
             type="PRICE", G=G, K=K, J=J)
CLUSTER.PLOT(i=which(catnames=="BER"), j=which(catnames=="FRJ"), bootFIT=bootFIT,
             type="PROMO", G=G, K=K, J=J)
CLUSTER.PLOT(i=which(catnames=="BER"), j=which(catnames=="BJC"), bootFIT=bootFIT,
             type="SLS", G=G, K=K, J=J)
```

Figure 2, page 13, containing the product category networks of prices on sales for each of the $K = 15$ stores is obtained using the function `NETWORK.PLOT`:

```
Kstores <- c(4,9,1,10,2,6,8,12,13,14,15,3,5,7,11) # Store Numbers Dominick Data
NETWORK.PLOT(bootFIT=bootFIT, type="PRICE", Kclasses=Kstores, G=G, K=K, J=J)
```

Figure 3 panels (a)-(c), page 15, containing the similarity matrix for the within- and cross-category effects are obtained using the function `SIMILARITY.PLOT`:

```
SIMILARITY.PLOT(bootFIT=bootFIT, type="PRICE", Kclasses=Kstores, G=G, K=K, J=J)
SIMILARITY.PLOT(bootFIT=bootFIT, type="PROMO", Kclasses=Kstores, G=G, K=K, J=J)
SIMILARITY.PLOT(bootFIT=bootFIT, type="SLS", Kclasses=Kstores, G=G, K=K, J=J)
```

1.4 Extensions

We give the code to obtain the three extensions of the multi-class VAR approach, as discussed in Section 6.

To use the adaptive group lasso (Section 6.1), one has to change the `type` of estimator to `"GrLasso"` and fill in the argument `group`, which is an PJ^2K vector, of the same length as $\boldsymbol{\beta}$, indicating the group to which each corresponding coefficient belongs. Since there are $GJK = 5 \times 15 \times 15 = 1125$ groups of size 3 for the multi-store sales application, each element of `groupindex` is a label in-between 1 and 1125. This gives

```
groupindex <- c()
i.index <- seq(from=1, to=(G*J*K-(G-1)), by=G)
for(i in i.index){
  groupindex <- c(groupindex, rep(seq(from=i,length=G,by=1), J/G))
}
FITGroupLasso <- MultiClass_VAR(Data, P=1, type="GrLasso", group=groupindex)
bootFITGroupLasso <- bootSE(FITOBJECT=FITGroupLasso, Nboot=1000, parallell=TRUE,
                            type="GrLasso", group=groupindex)
```

To incorporate additional clustering information (Section 6.2), one has to provide a $K \times K$ distance matrix `Distances`, and put the argument `lambda_weights` equal to `TRUE` to adjust the regularization parameter λ_1:

```
load("Data/Distances.RData")
FITDemo <- MultiClass_VAR(Data, P=1, lambda_weights=TRUE, Clusterinfo=Distances)
bootFITDemo <- bootSE(FITOBJECT=FITDemo, Nboot=1000, parallell=TRUE, lambda_weights=TRUE,
                      Clusterinfo=Distances)
```

To allow for cross-error correlations (Section 6.3), use the function `MultiClass_VAR_generalOmega` instead of the basic function `MultiClass_VAR`:

```
FITOmega <- MultiClass_VAR_generalOmega(Data, P=1)
bootFITOmega <- bootSE_GeneralOmega(FITOBJECT= FITOmega, Nboot=1000,parallell=TRUE)
```

2 Subset of estimated parameters with corresponding standard errors

Table 1: Estimated effects of (a) Beer prices on Refrigerated Juices sales, (b) Beer promotion on Frozen Juices sales, (c) Beer sales on Bottled Juices sales. Corresponding standard errors are in parentheses. The 95% confidence intervals are given in Figure 1, page 11 of the manuscript.

Store	BER prices on RFJ sales		BER promotion on FRJ sales		BER sales on BJC sales	
1	-0.09	(0.06)	-0.11	(0.06)	0.01	(0.05)
2	-0.07	(0.06)	-0.15	(0.06)	-0.03	(0.05)
3	-0.03	(0.06	-0.06	(0.06)	0.04	(0.05)
4	-0.11	(0.05)	-0.06	(0.05)	0.03	(0.04)
5	-0.05	(0.06)	-0.11	(0.06)	0.00	(0.05)
6	-0.12	(0.06)	-0.08	(0.06)	-0.05	(0.05)
7	-0.05	(0.06)	-0.05	(0.06)	-0.04	(0.05)
8	-0.03	(0.05)	-0.11	(0.06)	0.02	(0.04)
9	0.07	(0.06)	-0.03	(0.06)	0.01	(0.05)
10	0.00	(0.05)	-0.06	(0.06)	0.02	(0.05)
11	-0.10	(0.05)	-0.12	(0.06)	0.04	(0.05)
12	-0.04	(0.06)	-0.11	(0.06)	0.03	(0.05)
13	-0.12	(0.05)	-0.10	(0.06)	0.03	(0.04)
14	-0.07	(0.06)	-0.09	(0.06)	0.06	(0.05)
15	0.16	(0.08)	0.00	(0.05)	0.00	(0.05)