# Biclustering via Sparse Orthogonal Factor Regression

Dong Liang and Grace Guinan

SOFAR can be applied to several multivariate statistical problems (Uematsu et al. 2019). In particular, it can solve the bi-clustering problem, which aims to cluster simultaneously the rows and columns of a matrix (Lee et al. 2010). An example of such a problem is clustering a group of patients according to the genotypes. These patients may form groups naturally based on the types of cancer. The algorithm attempts to group patients without using the actual disease information.

Let $X$ denote the HRMS mass matrix, let $C$ be a coefficient matrix of the same dimension which admits a sparse singular value decomposition, and $I$ an $n \times n$ identity matrix. The problem can be formulated as the following mean model.

$$X_{n\times p} = I_{n\times n}C_{n\times p} + E_{n\times p} = C_{n\times p} + E_{n\times p} \tag{1}$$

The right singular vectors of the SOFAR solution $C = UDV$ represent latent factors approximating the HRMS mass data, while the left singular vectors of the SOFAR solution represent the loadings.

## Load Data

First, load data and select common signatures (compounds with non-zero entries across data points).

```
rm(list=ls())
library("dplyr")

# common signature quant SOFAR run
X ← read.csv("data/HRMS_data.csv")

# select non-zero columns (common signature)
X ← X %>% select_if(∼min(., na.rm = TRUE) > 0)
X1 ← as.matrix(X[,−c(1:4)])
dim(X1)
```

```
[1]  145 2201
```

## Normalize Data

Apply log transformation to normalize the feature values.

```
# log transformation
X2 ← log(X1, base=10)
```

We now center and scale the data, using a quantile method with $\alpha = .05$.

```
# center and scale
alpha = .05
center ← apply(X2,2,mean)
X2b ← sweep(X2, 2, center, FUN = "−")
```

```
scale ← apply(X2b,2, function(x) diff(quantile(x,probs=c(alpha,1-alpha))))
X3 ← sweep(X2b, 2, scale, FUN = "/")
```

## Run SOFAR

Now we apply the SOFAR algorithm to fit the mean model from the equation (1). The tuning parameter includes the lower limit of the penalty parameter controlling the sparsity of the solution `lam.min.factor`, and the number of parameters to try (`nlam`). The desired rank was chosen to be at most 12. We used GIC to favor a sparser solution over BIC.

```
if(!interactive()){
  library(rrpack)
  X3 = as.matrix(X3)
  fit ← sofar(Y = X3 ,X=diag(nrow(X3)),nrank = 12,ic.type="GIC",control =
                list(nlam=100,lam.min.factor=1e-7))
  save(fit, center, scale, file = "fit_common_signature.rData")
}
```

Now we load the saved fit and look at how it did.

```
library(RColorBrewer)
library(ggplot2)

load(file="fit/fit_common_signature.rData")
source("helper_functions/sofarUtils.R")
source("helper_functions/image_sofar_3.R")
summary(fit)
```

```
rank= 7
R2= 0.866
```

```
factor1-1 factor1-2 factor1-3 factor1-4 factor1-5 factor1-6 factor1-7
      125       203       268       305       323       336       344
```

The fitted value seems to match the observed value well from the simulation study. The algorithm converged to a rank 2 solution. The $R^2$ for this solution is about 86%.

We can calculate the percent that each layer explains:

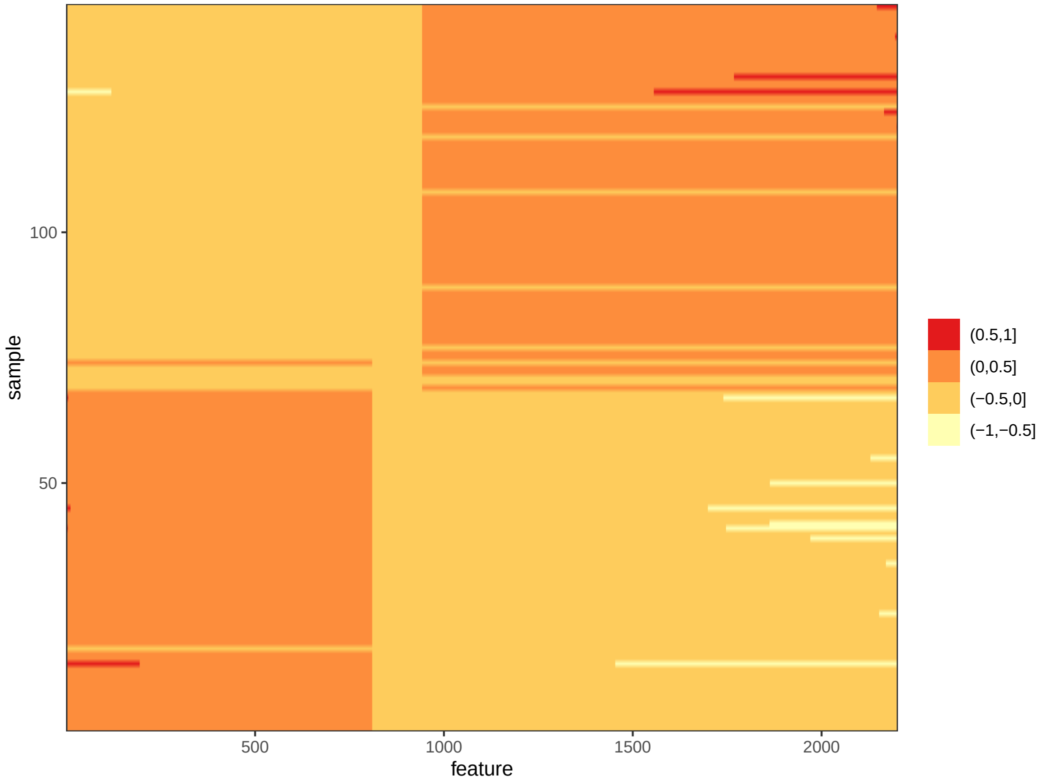
```
r2 = 86
svd1 = fit[["D"]]
layers = svd1^2/sum(svd1^2)
layers * r2
```

```
[1] 48.062 19.033 13.005  4.143  1.053  0.526  0.178
```

## Plot

We can visualize Layer 1 using helper functions `fitted.sofar.Z.rank()` and `image.sofar` from the files `sofarUtils.R` and `image_sofar_3.R`.

```
## fitted values
yhat ← fitted.sofar.Z.rank(fit,rank=1)
image.sofar(yhat, legend.title = "")+guides(fill = guide_legend(reverse=TRUE))
```



Samples 1-68 are from the BATS location and samples 69-145 are from the ALOHA location. The graph separates the samples from each location, as shown below.

```
min(which(yhat[,1]<0))
```

```
[1] 69
```

Taking a look at the plot, we can see where the feature cutoff is from orange to yellow (i.e. which compounds are associated with each location.

```
max(which(yhat[1,]>0))
```

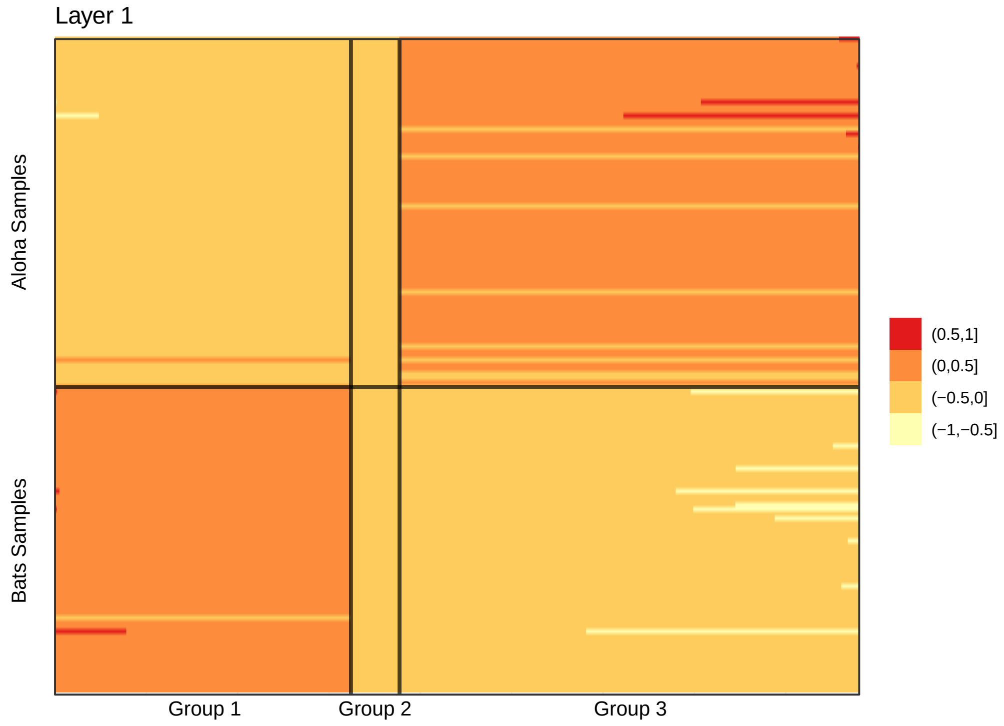
```
[1] 810
```

```
min(which(yhat[1,]<0))
```

```
[1] 943
```

Annotations are manually added to the plot to show the differentiation of groups visualized.

```
## fitted values
yhat ← fitted.sofar.Z.rank(fit,rank=1)
image.sofar(yhat, legend.title = "")+guides(fill = guide_legend(reverse=TRUE))
   +
  annotate("segment", x = 810, xend = 810, y = 0, yend = 145,colour = "black",
      linewidth = 1, alpha = .7)+
  annotate("segment", x = 943, xend = 943, y = 0, yend = 145,colour = "black",
      linewidth = 1, alpha = .7)+
  annotate("segment", x = 0, xend = 2201, y = 68, yend = 68, colour = "black",
      linewidth = 1, alpha = .7)+
  annotate(”text”, x = −100, y = 104.5, label = ”Aloha Samples”, angle = 90,
      size = 4)+
  annotate(”text”, x = −100, y = 34, label = ”Bats Samples”, angle = 90, size
      = 4)+
  annotate(”text”, x =410 , y = −3, label = ”Group 1”, size = 4)+
  annotate(”text”, x =876 , y = −3, label = ”Group 2”, size = 4)+
  annotate(”text”, x =1575 , y = −3, label = ”Group 3”, size = 4)+
  ggtitle(”Layer 1”)+ylab(””)+xlab(””)+
  coord_cartesian(xlim = c(0, 2201), ylim = c(0,145), clip = ’off’)+
  theme(axis.text.x = element_blank(), axis.ticks = element_blank(),
       axis.text.y = element_blank(), axis.title=element_text(size=45))
```



This is layer 1 of SOFAR bi-clustering solution visualized. Layer 1 makes up the largest percentage (43 percent) of the total variation in HRMS data. The mass variable is on the x axis and organized sorted by loading value. After calculating the coefficient matrix for the first level, values were split into discrete color groups. From least to most expressed, these are, (-1, -.5], (-.5, 0], (0, .5], and (.5, 1]. This layer indicates that BATS and ALOHA are associated with different masses. The masses are split into three groups: (1) more expressed in BATS, (2) same expression, (3) more expressed in ALOHA.

## Reference

1. Lee, M., Shen, H., Huang, J.Z. and Marron, J.S., 2010. Biclustering via sparse singular value decomposition. Biometrics, 66(4), pp.1087-1095.

2. Uematsu, Y., et al. (2019). ”SOFAR: Large-scale association network learning.” IEEE transactions on information theory 65(8): 4924-4939.