Variable selection in topic models/ cell cycle models

Kushal K Dey

December 27, 2015

Introduction

In topic modeling or cellcycleR applications, one major issue that needs addressing is the variable selection problem. Usually the user encounters in the range of 20000 to 50000 genes and a majority of genes are non-informative and do not contribute to the clustering (topic model) or cell ordering (cellcycleR). They add undue noise to the model fitting, besides increasing the computational time and diminishing the predictive power of the models.

Besides variable selection under completely unsupervised set up as in most cases, we may encounter scenarios where we have information about which genes to focus on. For example, for cell cycle applications, one may want to focus solely on cell cycle genes and just selecting these genes and re-doing the analysis may show additional patterns in cellcycleR or topic model (as in Yoav's single cell data case). So, variable selection under partially supervised set up is expected to pool in these genes corresponding to the metadata of our interest.

We first deal with the variable selection under unsupervised set up. We present the algorithm next for the variable selction in topic models, with additional considerations stated for the cellcycleR or cell times sorted cells in cell cycle set up.

Algorithm

We present the algorithm for variable selection in topic models and then we suggest the required modifications for ordered samples as we may encounter in cellcycle data.

- Start with the counts data $c_{G\times N}$ for the topic model set up where G is the number of genes and N the number of samples.
- We mean center the columns (subtract the column mean from each column) and then mean center the rows (subtract the row mean from each row).
- We apply penalized matrix decomposition due to function *PMD()* in package **PMA** on the matrix $c_{G\times N}$ for a given K (depending on the number of factors we want to model) and take the laodings $u_{G\times K}$.
- The genes which are informative for clustering will have high loadings in one or more factors. However, the non-informative genes will not have completely 0 loading in most cases.
- To counter this, we apply adaptive shrinkage **ash** on each column of $u_{G\times K}$ with $\hat{\beta}$ being the u value and the s is chosen to be $1/G$, an ad hoc choice since we know that $u^T u$ is equal to 1.
- For each gene, we take the 2-norm of ash-shrunk loadings across the factors, and then choose a threshold (choices taken were $1e-03$, $1e-04$, $1e-06$ etc) and select the genes which manage to cross the threshold.
- These are the genes that are considered informative for topic model. If we choose our threshold large, we will miss out on genes with moderate effects, and if the threshold is too small, we may select non-informative genes as well. So, choosing the appropriate threshold is of utmost importance.

For the cell cycle data with the N samples ordered by time on the cell cycle, we follow the same mechanism, except that we do not mean center the observations, but takes a wavelet transformation of the cell cycle data for each gene and then apply the factor analysis model on the wavelet transforms instead of on the actual data, as it has been shown to be more efficient at picking up meaningful factors.

Example (topic models)

We present an example of a topic model set up where we apply the varaible selection technique. The main purpose of this example is to show how the algorithm works as of now and then to discuss on possible modifications that we may target on.

Consider the simulation design as follows

```
library(maptpx)
```

```
## Loading required package: slam
```

```
n.out <- 200
omega_sim <- cbind(seq(0.6,0.4,length.out=n.out), 1- seq(0.6,0.4,length.out=n.out));
K <- dim(omega_sim)[2];
barplot(t(omega_sim),col=2:(K+1),axisnames=F,space=0,border=NA,main=paste("No. of clusters=",K),
        las=1,ylim=c(0,1),cex.axis=1.5,cex.main=1.4)
```


```
freq <- rbind(c(0.1,0.2,rep(0.70/98,98)),c(rep(0.70/98,98), 0.1,0.2));
counts <- t(do.call(cbind,lapply(1:dim(omega_sim)[1],
                                 function(x) rmultinom(1,1000,prob=omega_sim[x,]%*%freq))));
```

By the way we have defined the set up, besides the first two and the last two genes, all the other genes are unimportant, or in other words, all the essential information required for the clustering is contained in the first two and the last two genes only.

We first apply topic model without the variable selection and then compare the estimated Structure plots with the true one.

```
system.time(Topic_clus <- topics(counts, K=2,tol=0.001));
```

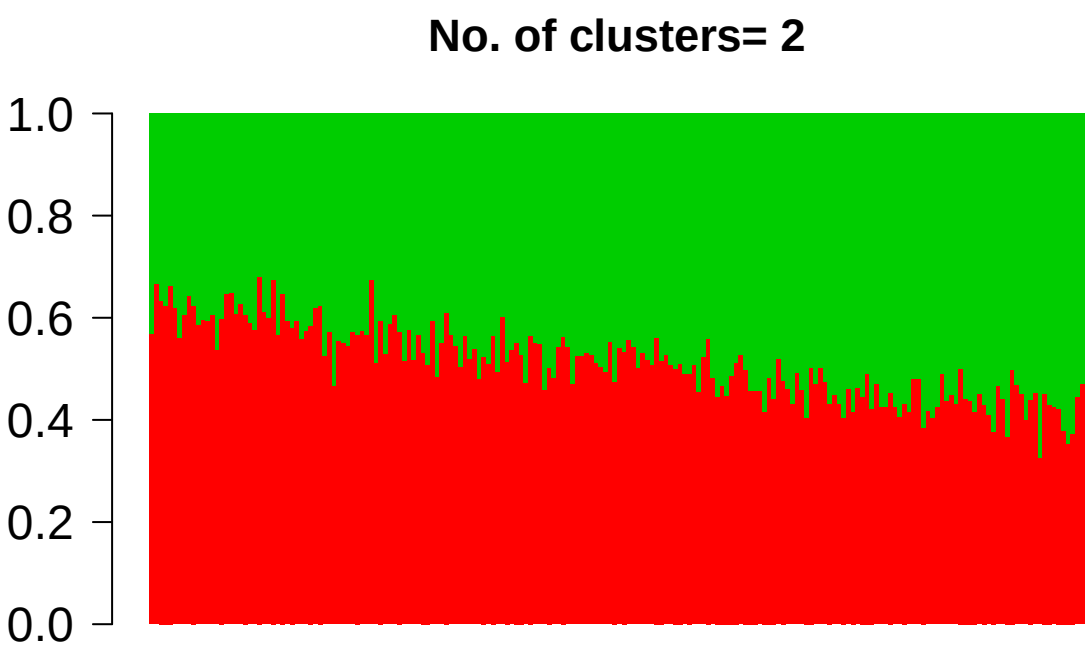
```
##
## Estimating on a 200 document collection.
## Fitting the 2 topic model.
## log posterior increase: 16.151, 23.936, 0.664, 0.628, 0.611, 0.612, 0.632, 0.672, 0.735, 0.828, 0.95
```

```
##    user  system elapsed
##   2.526   0.113   2.665
```

```
K=2
docweights <- Topic_clus$omega;
library(permute);
library("BioPhysConnectoR");
```

```
## Loading required package: snow
## Loading required package: matrixcalc
```

```
perm_set=rbind(1:K,allPerms(1:K));
diff=array(0,dim(perm_set)[1]);
for (p in 1:dim(perm_set)[1])
{
    temp=docweights[,perm_set[p,]];
    diff[p]=fnorm(temp,omega_sim);
}

p_star=which(diff==min(diff));
docweights=docweights[,perm_set[p_star,]];

barplot(t(docweights),col=2:(K+1),axisnames=F,space=0,border=NA,main=paste("No. of clusters=",K),las=1,
```



Next we perform our algorithm. First we mean center the rows and columns (Step 2).

```
data_norm1 <- apply(counts,2,function(x) return(x-mean(x)));
data_norm2 <- apply(data_norm1, 1, function(x) return(x-mean(x)));
```

Next we perform penalized matrix decompsition (Step 3).

```
library(PMA)
```

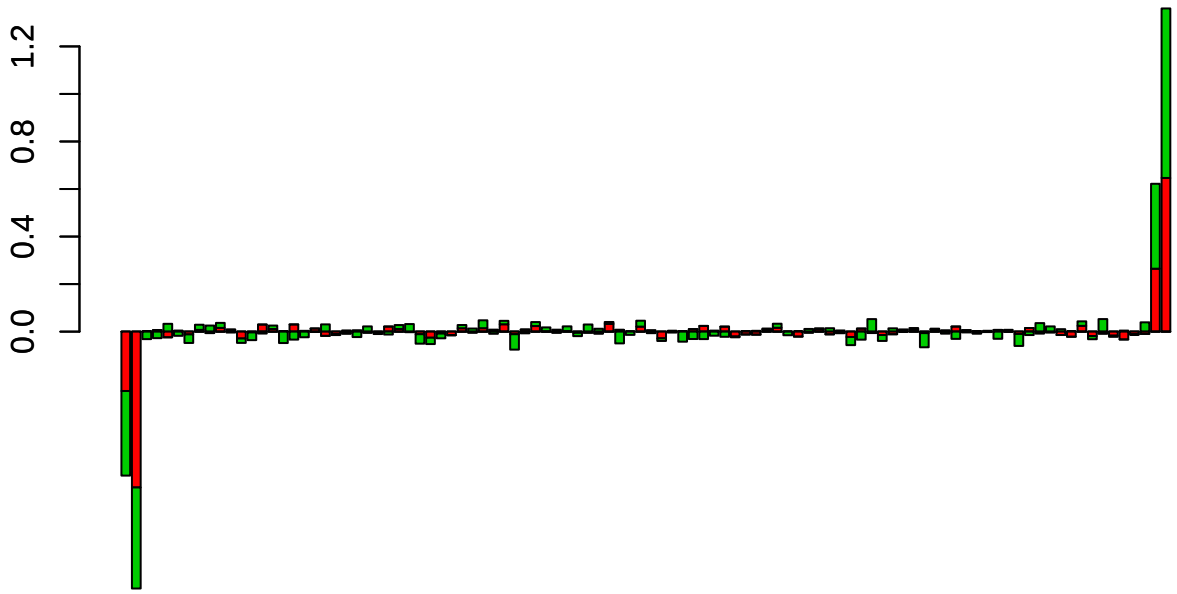
```
## Loading required package: plyr
## Loading required package: impute
```

```
pmd1 <- PMD(data_norm2, type="standard", K=2, niter=50);
```

```
## 1234567891011121314
## 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849 50
```

We explore the loadings (Step 4).

```
pos_u <- apply(pmd1$u, c(1,2), function(x) return (max(x,0)))
neg_u <- apply(pmd1$u, c(1,2), function(x) return (min(x,0)))

par(mar=c(12,2,2,1))
barplot(t(pos_u), col=2:3)
barplot(t(neg_u), col=2:3, add=TRUE)
```



We apply the *ash* type shrinkage (Step 5).

```
library(ashr)
shrunk_u <- suppressWarnings(apply(pmd1$u, 2,
                                   function(x) return(ash(x,sqrt(1/length(x)))$PosteriorMean)));
```

```
## Due to absence of package REBayes, switching to EM algorithm
## Loading required package: Rcpp
## Due to absence of package REBayes, switching to EM algorithm
```

```
pos_shrunk_u <- apply(shrunk_u, c(1,2), function(x) return (max(x,0)))
neg_shrunk_u <- apply(shrunk_u, c(1,2), function(x) return (min(x,0)))

par(mar=c(12,2,2,1))
barplot(t(pos_shrunk_u), col=2:3)
barplot(t(neg_shrunk_u), col=2:3, add=TRUE)
```

We selct the genes based on thresholds (Step 6). Two choices considered here - 0.001 and 0.0001. First we do this for threshold 0.001.

```
ss_loadings <- apply(shrunk_u,1,function(x) return(sqrt(sum(x^2))))
which(ss_loadings > 1e-03)
```

```
## [1]   1   2  99 100
```

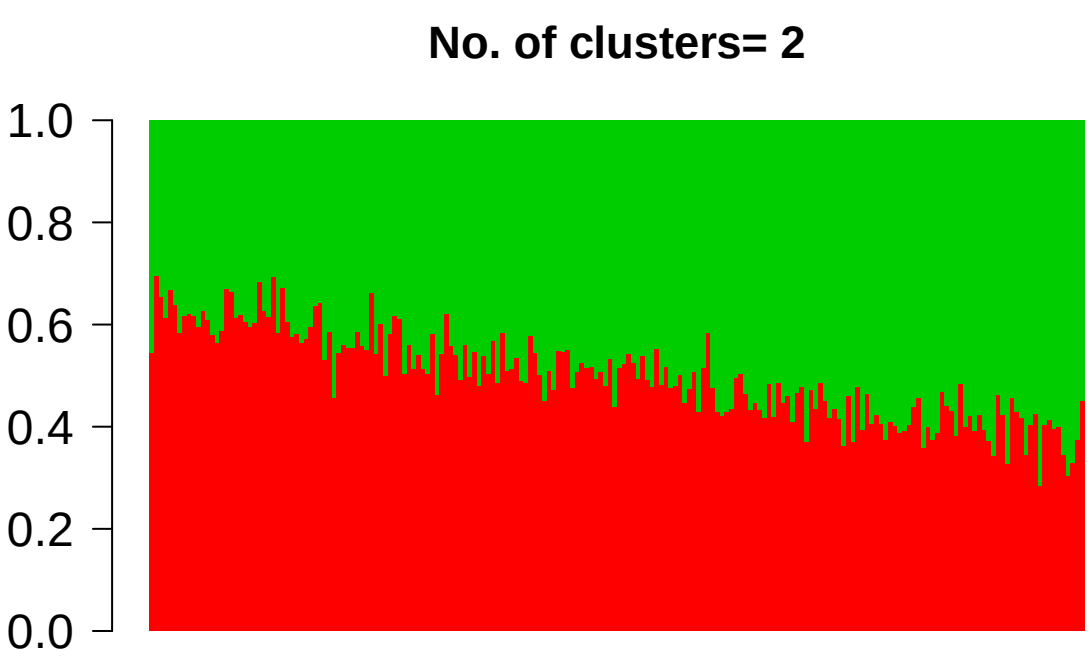
We plot the Structure plot for the reduced data taking only the selected genes.

```
vs_counts <- counts[,which(ss_loadings > 1e-03)];
system.time(Topic_clus <- topics(vs_counts, K=2,tol=0.001));
```

```
##
## Estimating on a 200 document collection.
## Fitting the 2 topic model.
## log posterior increase: 32.074, 1.146, 0.525, 0.001, done.

##    user  system elapsed
##   0.052   0.001   0.054
```

```
K=2
docweights <- Topic_clus$omega;
library(permute);
library("BioPhysConnectoR");
```

```
perm_set=rbind(1:K,allPerms(1:K));
diff=array(0,dim(perm_set)[1]);
for (p in 1:dim(perm_set)[1])
{
    temp=docweights[,perm_set[p,]];
    diff[p]=fnorm(temp,omega_sim);
}

p_star=which(diff==min(diff));
docweights=docweights[,perm_set[p_star,]];

barplot(t(docweights),col=2:(K+1),axisnames=F,space=0,border=NA,main=paste("No. of clusters=",K),las=1,
```



Next we consider the case with threshold 0.0001.

```
ss_loadings <- apply(shrunk_u,1,function(x) return(sqrt(sum(x^2))))
which(ss_loadings > 1e-04)
```

```
##  [1]   1   2   3   4   5   6   7   8   9  10  12  13  14  15  16  17  18
## [18]  20  23  24  26  27  28  29  30  31  33  35  37  38  40  41  43  44
## [35]  45  47  48  49  50  52  54  55  56  57  58  63  64  68  70  71  72
## [52]  73  74  77  80  84  86  87  88  89  92  93  94  96  98  99 100
```

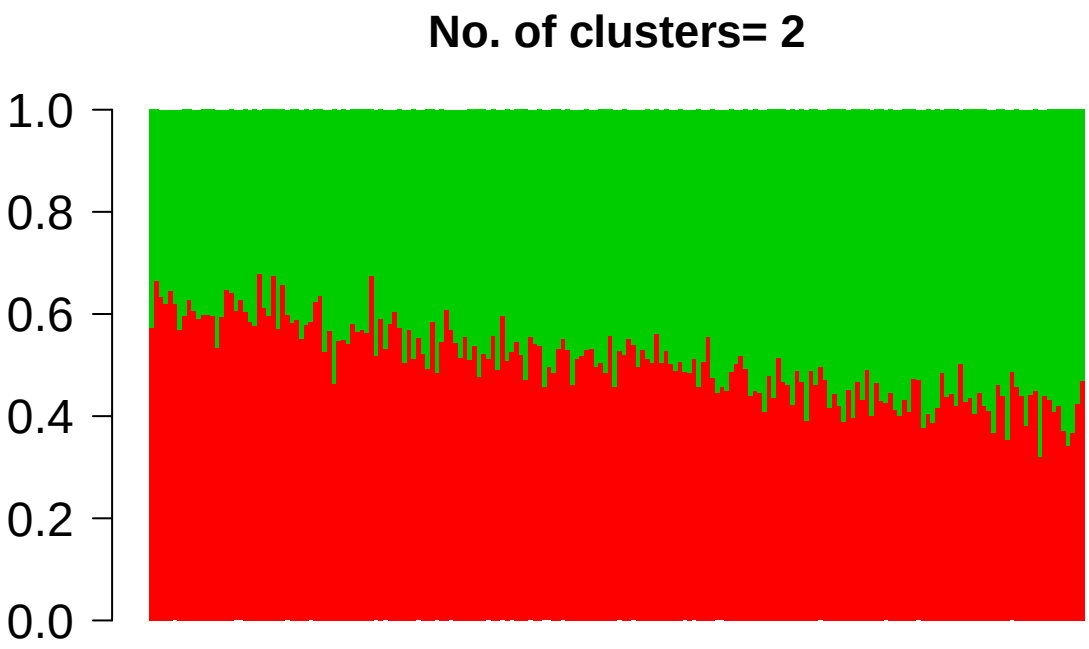
We now perform topic model on the selected genes.

```
vs_counts <- counts[,which(ss_loadings > 1e-04)];
system.time(Topic_clus <- topics(vs_counts, K=2,tol=0.001));
```

```
##
## Estimating on a 200 document collection.
## Fitting the 2 topic model.
## log posterior increase: 17.063, 1.186, 0.475, 0.436, 0.408, 0.391, 0.384, 0.387, 0.402, 0.431, 0.478
```

```
##    user  system elapsed
##   1.600   0.020   1.641
```

```
K=2
docweights <- Topic_clus$omega;
library(permute);
library("BioPhysConnectoR");
perm_set=rbind(1:K,allPerms(1:K));
diff=array(0,dim(perm_set)[1]);
for (p in 1:dim(perm_set)[1])
{
    temp=docweights[,perm_set[p,]];
    diff[p]=fnorm(temp,omega_sim);
}

p_star=which(diff==min(diff));
docweights=docweights[,perm_set[p_star,]];

barplot(t(docweights),col=2:(K+1),axisnames=F,space=0,border=NA,main=paste("No. of clusters=",K),las=1,y
```



Discussion

The model we have proposed here has many loopholes that need to be taken care of appropriately for the method to work on a generic basis.

- We used an ad-hoc approach to shrink the loadings data. The loadings are constrained as their sum of squares is equal to 1. So, we need a better strategy to do the shrinkage. Also, we want to probably do a hard thresholding and put the loadings for genes that are non-informative to 0 completely, thereby selecting only the genes with non-zero loadings.
- Off the shrunk loadings, if we do not do hard thresholding, we may need a threshold for choosing the genes in Step 6, so then we have to fix that threshold appropriately as the choice is very imporant for determining the cluster patterns.

- We need to have a strategy to incorporate the metadata information that would help us in choosing the genes that satisfy certain conditions besides selecting other genes. For example, we may want to choose the cell cycle genes, besides choosing other genes from non cell cycle genes into our subset of genes of interest.
- We applied the **PMA** package for performing the factor analysis or penalized matrix decomposition in this case. The other option would be to use the **SFA** software, but the aim here is to make this variable selection startegy a part of the **CountClust** package and hence, probably a R package to implement this factor analysis model would be more appropriate.
- For cellcycleR, we would usually not have cells in sorted time order on the cell cycle, rather the ordring is something we need to figure out. So for such a scenario, it is not meaningful to apply wavelet transform. It could be of interest to figure out how to adjust for the wavelet transform for the unsynchronized cell experiment.

To know more about the **PMA** package, check this paper and package info and for further example uses of our method check the Variable Selection section in our webpage.