

Stat Network Analysis Assignment 2

16343261

13/7/2022

Contents

Before we begin this assignment, I will load in the data and packages as required.

```
setwd("/Users/matth/Documents/Stat Network Analysis")  
load("/Users/matth/Documents/Stat Network Analysis/data_oil_2019.RData")
```

```
library(igraph)  
require(blockmodels)  
library(network)  
library(latentnet)
```

Question 0. To answer this question, we will remove the countries with no connections using the **Which** function.

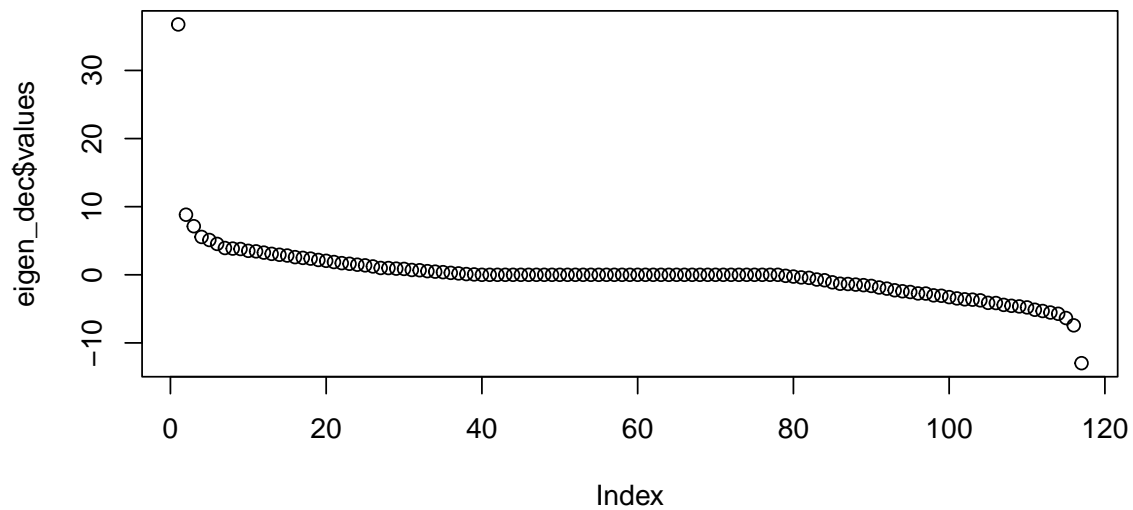
```
Y2 <- Y[, which(colSums(Y) != 0)]  
Y2 <- Y2[which(rowSums(Y) != 0), ]
```

Using the **ifelse** function, we can construct our binary matrix. Subsequently we will set our diagonal to zero and using the **graph.adjacency** function, we will produce our undirected binary adjacency matrix.

```
adj_bin <- ifelse(Y2 > 0, 1, 0)  
diag(adj_bin) = 0  
X_bin <- graph.adjacency(adj_bin, "undirected")  
layout <- layout.fruchterman.reingold(X_bin)
```

Question 1. Before we are able to perform Spectral Clustering on our adjacency matrix, we need to calculate the eigenvalues and eigenvectors of the adjacency matrix. We will then plot our eigenvalues to discover how many clusters will be required.

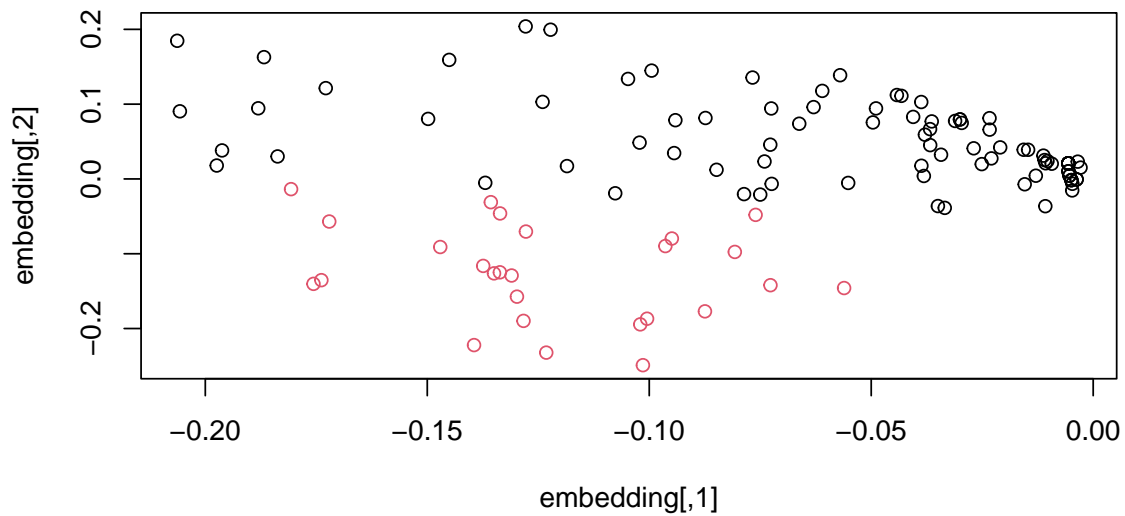
```
# Spectral Decomposition on our Adjacency Matrix.  
eigen_dec <- eigen(adj_bin)  
# Plot Eigenvalues  
plot(eigen_dec$values)
```



Here we see that either 1 or 2 clusters are our most reasonable dimensions. We will now perform Clustering by embedding these eigenvectors into 2 Kmeans clusters.

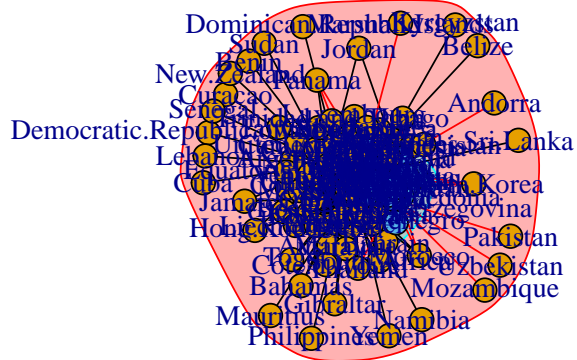
```
# Embed first 2 eigenvectors  
embedding <- eigen_dec$vectors[,1:2]  
# Run K-Means clustering on our eigenvectors.  
memberships <- kmeans(embedding, 2, nstart = 1000)$cluster
```

```
# Plot vector values, coloured by cluster members.  
plot(embedding, col = memberships)
```



Here we see our Clusters. One is noticeably bigger than the other.

```
# Make Clusters for Plotting.
res <- make_clusters(X_bin, memberships)
plot(x = res, y = X_bin, layout = layout)
```



Here we see considerable overlap between our clusters. This is disappointing as it makes our clusters difficult to interpret. One notable observation that can be made in relation to the geographical

locations of countries is that island nations appear to be more common on our outer cluster with New Zealand, Marshall Islands, Sri Lanka, Cuba, Jamaica all on this outer area. These countries are also largely Asia-Pacific Nations.

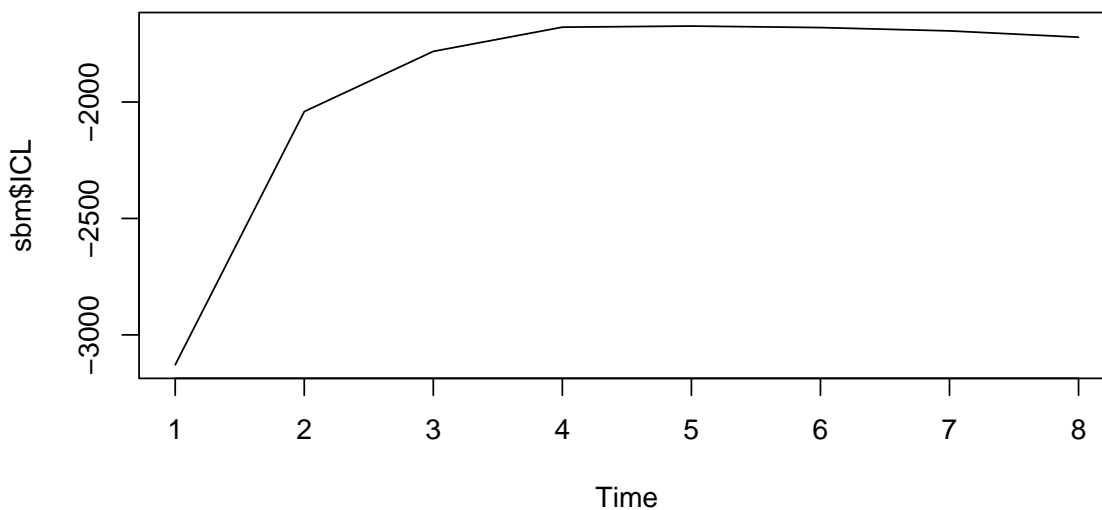
Question 2. We will now fit our Stochastic Block Model using the **BM_Bernoulli** function. We have defined $K = 8$ as the maximum number of groups that we will consider.

```
# Fit stochastic block model for adjacency for K=8 as max.
sbm <- BM_bernoulli(membership_type = "SBM_sym",
adj = adj_bin,
verbosity = 0,
plotting = "",
explore_max = 8)
```

Now let's find our estimates and let's look at the Integrated Completed Likelihood to choose the number of groups.

```
# ICL Plot
set.seed(12345)
sbm$estimate()
```

```
ts.plot(sbm$ICL)
```

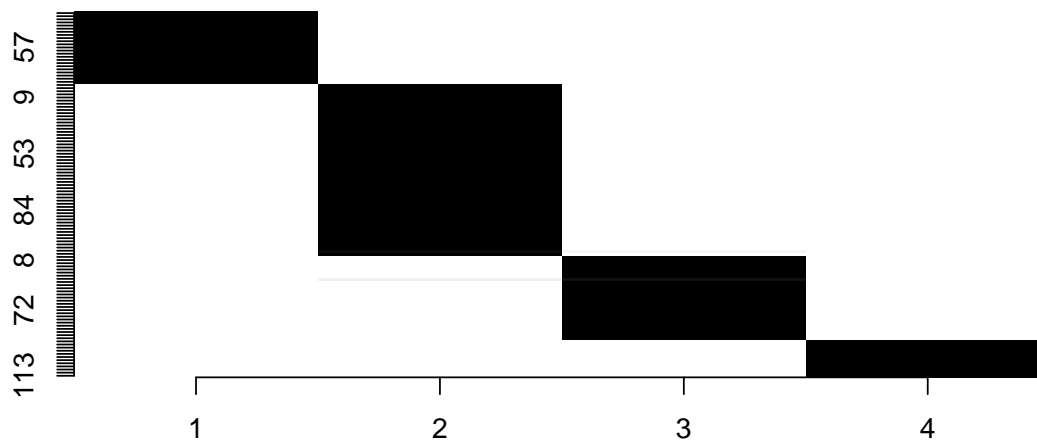


Here we find that our ICL peaks at $K = 4$. This means we will consider the solution with 4 groups.

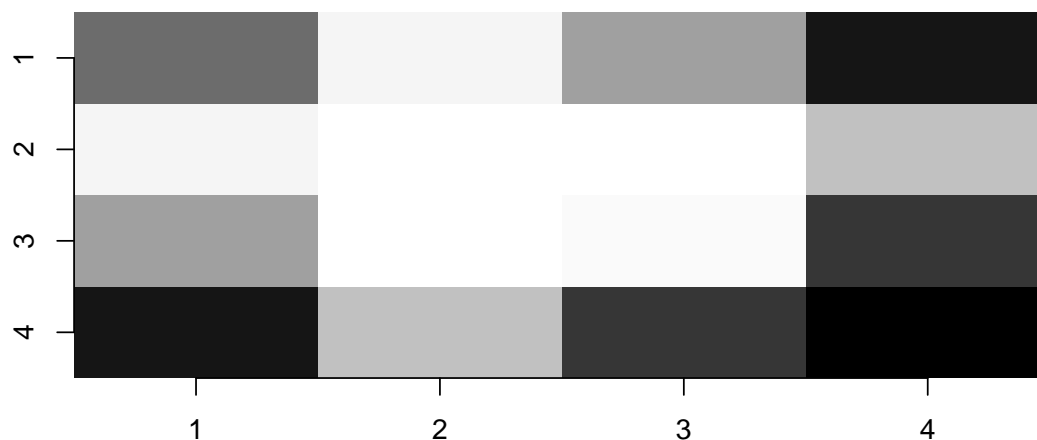
```
K_star <- 4
```

Now we start a characterisation of this clustering solution. We will also define our Soft and subsequently Hard Clustering for our Block Model Plot.

```
# Block Model and Parameters plot needed for interpretation of model.
sbm$memberships[[K_star]]$plot()
```



```
sbm$plot_parameters(K_star)
```

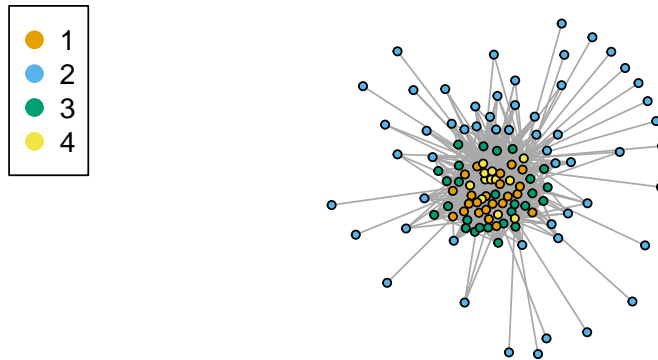


```
# Posterior probability that node i belongs to group g.
soft_clustering <- sbm$memberships[[K_star]]$Z
```

```

# Maximum-A-Posteriori
hard_clustering <- apply(soft_clustering, 1, which.max)
# Plot X and characterise by Cluster groups.
plot(X_bin, vertex.label = NA, vertex.size = 5, vertex.color = hard_clustering)
legend("topleft", legend = 1:K_star,
col = categorical_pal(4)[1:K_star],
pch = 20, pt.cex = 2)

```



We can make the following observations:

- Group 1 exhibit community structures and is a medium sized group of nodes with a strong connection to Group 4 and itself.
- Group 2 is a large group of nodes/countries with few connections. The nodes in this group have very low connection probabilities, overall, towards any group (including group 2, itself). Community structure is not exhibited by this group.
- Group 3 exhibits a clear community structure. No connections to group 2 and many connections to groups 1 and 4. When we look at relations between the remaining three groups, we see many connections to group 3 itself, and few connections to the others, which highlights a community structure for group 3.
- Group 4 is a small cluster of hubs. Not only these nodes/countries have very high connection probability between themselves, but they also have very high connection probabilities towards all other groups. Community structure can be noticed for this group, but it is not that relevant because block connection probabilities are very high anyway.

Question 3. To answer this question, we will first fit our X network using to the **Network** function from the network package. We will define our vertex names under the country variable. Lastly, we will use the **ergmm** function to fit our Latent Position Model. Here we will select MLE and set

our Euclidean dimensions to 2. Subsequently we will plot the MLE positions of our model using a generic **plot** function.

```
# X network
```

```
X_net <- network(adj_bin)
```

```
countries <- network.vertex.names(X_net)
```

```
knitr::opts_chunk$set(cache = TRUE)
```

```
# Fitting Latent Position Model.
```

```
lpm_x <- ergmm(X_net ~ euclidean(d=2), tofit = c("mle"), verbose=TRUE)
```

```
Generating initial values for MCMC:
```

```
Computing geodesic distances... Finished.
```

```
Computing MDS locations... Finished.
```

```
Computing other initial values... Finished.
```

```
Finding the conditional posterior mode... Finished.
```

```
Post-processing the MCMC output:
```

```
Double-checking conditional posterior mode estimate... Finished.
```

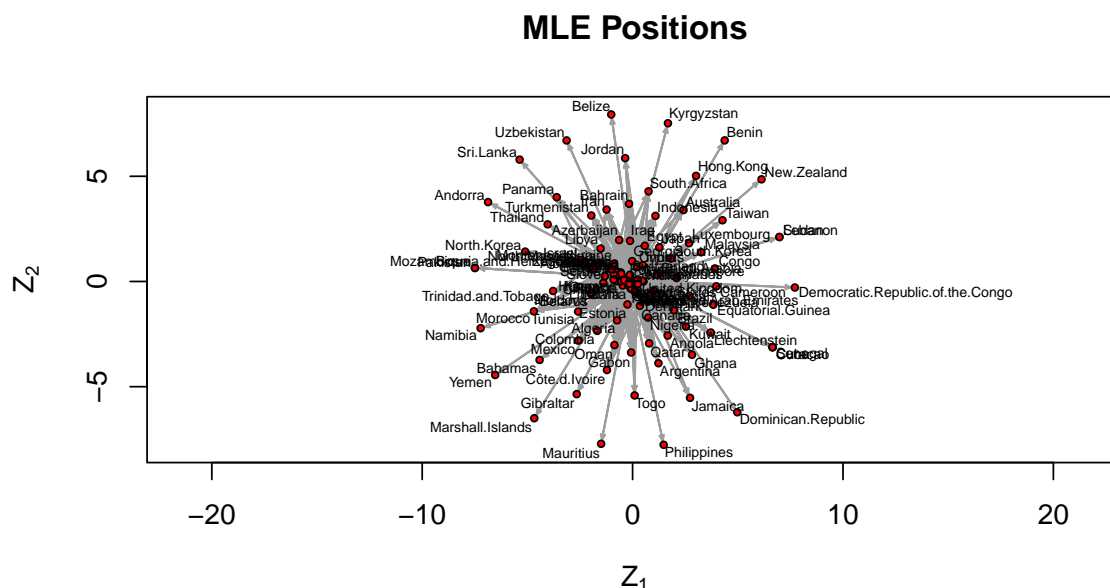
```
Using the conditional posterior mode to seed an MLE fit... Finished.
```

```
# MLE Plot for Latent Position Model
```

```
plot(lpm_x, label = countries,
```

```
what = "mle", main = "MLE Positions",
```

```
print.formula = FALSE, labels = TRUE, label.cex = 0.5, label.pad = 0.1)
```



Question 4. To run our simulation, I will be adapting the rscript given to us for generating LPM models to run for 100 random network simulations. I will then use these generated values and compare them with the network values of our X Network.

```
knitr::opts_chunk$set(cache = TRUE)
# Set Function
LPM_gen <- function(Z, beta){
  # 100 Simulations.
  counter = 1
  while(counter <= 100){
    N <- nrow(Z)
    adj <- matrix(0, N, N)
    # allocate an empty adjacency matrix
    for (i in 1:N) for (j in 1:N) if (i != j)
    {
      d <- sqrt( sum( (Z[i,]-Z[j,])^2 ) )
      # calculate the Euclidean distance between the two nodes
      logodds <- beta - d
      # calculate the logodds for this particular edge
      p <- exp(logodds) / (1+exp(logodds))
      # calculate the corresponding edge probability
      adj[i,j] = sample(c(0,1), 1, T, c(1-p,p))
      # test whether the corresponding edge appears or not
      adj[j,i] =adj[i,j]
    }
    return(adj)
    # return the adjacency matrix
  }
  counter = counter + 1
}
```

```
# Fit new LPM model using simulations.
lpm <- LPM_gen(lpm_x$mle$Z, lpm_x$mle$beta)
```

```
# Simulated Density
network.density(network(lpm))
```

```
[1] 0.1730032
```

```
# Empirical Density
network.density(X_net)
```

```
[1] 0.1728559
```

Here we find extremely similar network density between our empirical X network and the networks simulated using our latent position model.