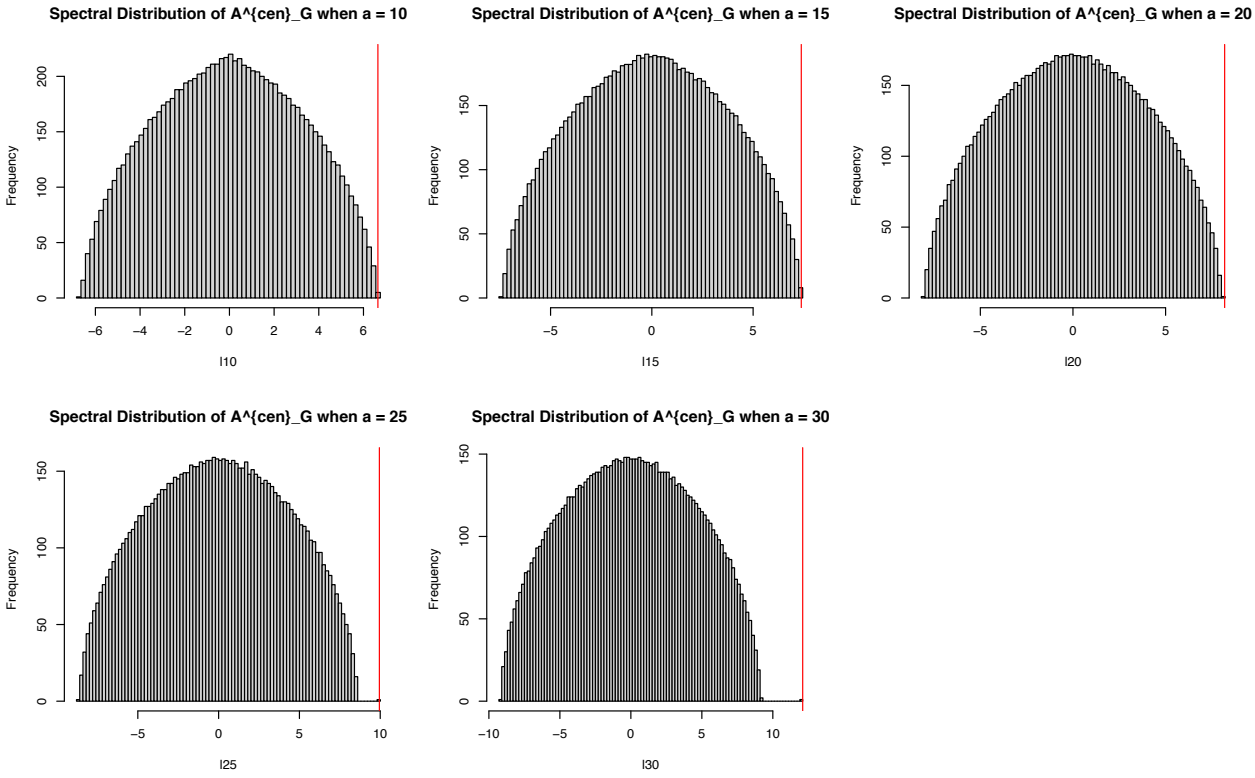
EE378B Homework 2 Solution

Due to: Isaac Gibbs

1 Part A

The desired plots are shown below. In all plots the maximum eigenvalue is marked by a red line. We see that as a increases the maximum eigenvalue becomes more and more separated from the bulk of the spectral distribution. The code for this question can be found starting on Page 5 of the document.



2 Part B

The average observed values of $Q(\sigma, \hat{\sigma})$ over the 20 samples are shown in the table below. We see that the overlap increases as a increases. The code for this part can be found starting on Page 6 of the document.

Value of a	Average Overlap
10	0.00853
15	0.0173
20	0.55027
25	0.86655
30	0.96097

3 Part C

Let $\sigma \in \{-1, 1\}^n$ denote the vector of group memberships with $\sigma_i = -1$ denoting that node i is in group 1 and $\sigma_i = 1$ denoting that node i is in group 2. In our model the nodes are partitioned uniformly at random into two groups of equal size. So, in particular, there are $n/2$ nodes with $\sigma_i = 1$ and $n/2$ nodes with $\sigma_i = -1$. I also assume that the graph is not allowed to have self-loops. Throughout the remainder of this section we will treat σ as a fixed vector (i.e., all the results that follow are conditional on the value of σ).

Let I_n denote the identity matrix in $\mathbb{R}^{n\times n}$ and $\mathbf{1}_n \in \mathbb{R}^n$ denote the vector of all ones. Then, we have that for all $i \neq j$

$$\mathbb{E}[(A_G)_{ij}] = \frac{a}{n}\mathbb{1}_{\sigma_i=\sigma_j} + \frac{b}{n}\mathbb{1}_{\sigma_i\neq\sigma_j}$$

Moreover, we also have that

$$\begin{aligned}
\mathbb{E}[\frac{\hat{d}}{n}] &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j\neq i}\frac{a}{n}\mathbb{1}_{\sigma_i=\sigma_j} + \frac{b}{n}\mathbb{1}_{\sigma_i\neq\sigma_j} \\
&= \frac{1}{n^2}\left(\sum_{i:\sigma_i=1}\sum_{j\neq i:\sigma_j=1}\frac{a}{n} + \sum_{i:\sigma_i=-1}\sum_{j\neq i:\sigma_j=-1}\frac{a}{n} + \sum_{i:\sigma_i=1}\sum_{j:\sigma_j=-1}\frac{b}{n} + \sum_{i:\sigma_i=-1}\sum_{j:\sigma_j=1}\frac{b}{n}\right) \\
&= 2\frac{1}{n^2}\frac{n}{2}(\frac{n}{2}-1)\frac{a}{n} + 2\frac{1}{n^2}\frac{n}{2}\frac{n}{2}\frac{b}{n} \\
&= \frac{a+b}{2n} - \frac{a}{n^2}
\end{aligned}$$

So, we find that for all $i \neq j$

$$\mathbb{E}[(A_G^{cen})_{ij}] = \mathbb{E}[(A_G)_{ij}] - \mathbb{E}[\frac{\hat{d}}{n}] = (\frac{a-b}{2n} + \frac{a}{n^2})\mathbb{1}_{\sigma_i=\sigma_j} + (\frac{b-a}{2n} + \frac{a}{n^2})\mathbb{1}_{\sigma_i\neq\sigma_j}$$

Finally, note that for $i = j$ we have that by assumption $(A_G)_{ij} = 0$ and thus that $(A_G^{cen})_{ij} = \frac{\hat{d}}{n}$. So, in total we find that

$$\begin{aligned}
\mathbb{E}[A_G^{cen}] &= \frac{a-b}{2n}\sigma\sigma^\top + \frac{a}{n^2}\mathbf{1}_n\mathbf{1}_n^\top - (\frac{a-b}{2n} + \frac{a}{n^2})I_n \\
&= \frac{a-b}{2}(\frac{\sigma}{\sqrt{n}})(\frac{\sigma}{\sqrt{n}})^\top + \frac{a}{n}(\frac{\mathbf{1}_n}{\sqrt{n}})(\frac{\mathbf{1}_n}{\sqrt{n}})^\top - (\frac{a-b}{2n} + \frac{a}{n^2})I_n
\end{aligned}$$

Now, write

$$A_G^{cen} - \mathbb{E}[A_G^{cen}] = A_G - \mathbb{E}[\frac{\hat{d}}{n}]\mathbf{1}_n\mathbf{1}_n^\top - \mathbb{E}[A_G^{cen}] + n(\mathbb{E}[\frac{\hat{d}}{n}] - \frac{\hat{d}}{n})(\frac{\mathbf{1}_n}{\sqrt{n}})(\frac{\mathbf{1}_n}{\sqrt{n}})^\top$$

Now, $A_G - \mathbb{E}[\frac{\hat{d}}{n}]\mathbf{1}_n\mathbf{1}_n^\top - \mathbb{E}[A_G^{cen}]$ is a matrix with independent entries that are bounded in the interval $[-2, 2]$. Thus, this matrix is sub-Gaussian and so by results from lecture we will have that for some constant C

$$||A_G - \mathbb{E}[\frac{\hat{d}}{n}]\mathbf{1}_n\mathbf{1}_n^\top - \mathbb{E}[A_G^{cen}]||_{op} \leq C\sqrt{n}, \text{ with probability tending to 1 as } n \to \infty$$

Moreover, note that $\hat{d}/n$ is equal to the average of n^2 bounded independent random variables. So, by Chernoff's bound we have that

$$|n(\mathbb{E}[\frac{\hat{d}}{n}] - \frac{\hat{d}}{n})| \leq 1, \text{ with probability tending to 1 as } n \to \infty$$

So, by applying the triangle inequality we find that with probability tending to 1 as $n \to \infty$

$$||A_G^{cen} - \mathbb{E}[A_G^{cen}]||_{op} \leq C\sqrt{n} + 1 \leq C'\sqrt{n}$$

Moreover, we also clearly have that

$$||\mathbb{E}[A_G^{cen}] - \frac{a-b}{2}(\frac{\sigma}{\sqrt{n}})(\frac{\sigma}{\sqrt{n}})^\top||_{op} \leq \frac{a}{n} + \frac{a-b}{2n} + \frac{a}{n^2} = O(n^{-1})$$

and thus in total we find that with probability tending to 1 as $n \to \infty$

$$||A_G^{cen} - \frac{a-b}{2}(\frac{\sigma}{\sqrt{n}})(\frac{\sigma}{\sqrt{n}})^\top||_{op} \leq C'\sqrt{n} + O(n^{-1}) \leq C''\sqrt{n}$$

By Weyl's inequality this further implies that with probability tending to 1 as $n \to \infty$ we have that for all $1 \leq i \leq n$

$$|\lambda_i(A) - \lambda_i\left(\frac{a-b}{2}(\frac{\sigma}{\sqrt{n}})(\frac{\sigma}{\sqrt{n}})^\top\right)| \leq C''\sqrt{n}$$

Thus, with high probability we have that $\lambda_1(A_G^{cen}) \in [(a-b)/2 - C''\sqrt{n}, (a-b)/2 + C''\sqrt{n}]$ and $\lambda_2(A_G^{cen}), \ldots, \lambda_n(A_G^{cen}) \in [-C''\sqrt{n}, C''\sqrt{n}]$. So, when $(a-b)/2$ is not much larger than $C''\sqrt{n}$ we would expect to see all the eigenvalues of A_G^{cen} scattered in the interval $[-C''\sqrt{n}, C''\sqrt{n}]$ and for larger values of $(a-b)/2$ we should expect to see one large eigenvalue in the interval $[(a-b)/2 - C''\sqrt{n}, (a-b)/2 + C''\sqrt{n}]$ and the rest of the eigenvalues scattered in the interval $[-C''\sqrt{n}, C''\sqrt{n}]$. This is consistent with what is observed in part A, where I saw that as a increased the maximum eigenvalue separated from the bulk of the spectral distribution.

Moreover, note that the above analysis shows that we can write

$$A_G^{cen} = \frac{a-b}{2}(\frac{\sigma}{\sqrt{n}})(\frac{\sigma}{\sqrt{n}})^\top + W$$

where with high probability $||W||_{op} \leq C''\sqrt{n}$. So, by Wedin's $\sin(\theta)$ theorem we have that when $\frac{a-b}{2}$ is large relative to $C''\sqrt{n}$ the first eigenvector of A_G^{cen} will be close to either $\sigma/\sqrt{n}$ or $-\sigma/\sqrt{n}$. Let

v denote the first eigenvector of A_G^{cen}. For simplicity in the notation assume that $||v - \sigma/\sqrt{n}||_2 \leq ||v + \sigma/\sqrt{n}||_2$. Note that

$$\max_{\pi} \frac{1}{4n} \sum_{i=1}^{n} \mathbb{1}_{\pi(\hat{\sigma}_i) \neq \sigma_i} = \frac{1}{n} ||\hat{\sigma} - \sigma||_2^2 = \frac{1}{n} \sum_{i=1}^{n} (\mathbb{1}_{v_i \geq 0} - \mathbb{1}_{v_i < 0} - (\mathbb{1}_{\sigma_i/\sqrt{n} \geq 0} - \mathbb{1}_{\sigma_i/\sqrt{n} < 0}))^2$$

$$\leq \frac{1}{n} \sum_{i=1}^{n} (\mathbb{1}_{|v_i - \sigma_i/\sqrt{n}| > n^{-1/2}})^2 \leq \sum_{i=1}^{n} (v_i - \frac{\sigma_i}{\sqrt{n}})^2 = ||v - \sigma||_2^2$$

In particular, v being close to $\sigma/\sqrt{n}$ implies that $\hat{\sigma}$ has good overlap with σ. Now, our calculations above suggest that v will be closer to $\sigma/\sqrt{n}$ when $\frac{a-b}{2}$ is larger. Thus, we expect to see better overlap between σ and $\hat{\sigma}$ when a is larger. This is consistent with the empirical results from part B where the overlap was seen to increase as a increased.

EE378B Homework 2

Isaac Gibbs

Code For Part A

```
library(rARPACK)

generateProbsPerNode2 <- function(nodelab,labels,a,b,n){
  return(sapply(labels,function(x){if(nodelab==x){return(a/n)}else{return(b/n)}}))
}

generateGraph2 <- function(n,b,a){
  labels <- sample(c(rep(0,n/2),rep(1,n/2)))
  m1 <- generateProbsPerNode2(0,labels,a,b,n)
  m2 <- generateProbsPerNode2(1,labels,a,b,n)
  v <- rbind(m1,m2)
  edges <- sapply(1:length(labels),function(x){rbinom(n,1,c(rep(0,x),rep(1,n-x))*v[labels[x]+1,])})
  edges <- edges + t(edges)
  return(list(labels,edges))
}


centerAdjMat <- function(A,n){
  hatd <- mean(A %*% rep(1,n))
  return(A - matrix(hatd/n,nrow=n,ncol=n))
}

laplacian <- function(A,n){
  degs <- A %*% rep(1,n)
  Dgrootminus <- diag(as.vector(sqrt(1/degs)),nrow=length(degs))
  zg <- sqrt(degs)/sqrt(sum(degs))
  L <- Dgrootminus %*% A %*% Dgrootminus - zg %*% t(zg)
  return(L)
}

unNormlaplacian <- function(A,n){
  degs <- A %*% rep(1,n)
  return(diag(degs) - A)
}

## n = 10000 was used for all results produced in my homework
n <- 100
neigsToCompute <- n

l10 <- eigs(centerAdjMat(generateGraph2(n,10,10)[[2]],n),neigsToCompute)$values
```

```
## Warning in eigs_real_sym(A, nrow(A), k, which, sigma, opts, mattype =
```

```
## "sym_matrix", : all eigenvalues are requested, eigen() is used instead
```

```
l15 <- eigs(centerAdjMat(generateGraph2(n,10,15)[[2]],n),neigsToCompute)$values
```

```
## Warning in eigs_real_sym(A, nrow(A), k, which, sigma, opts, mattype =
## "sym_matrix", : all eigenvalues are requested, eigen() is used instead
```

```
l20 <- eigs(centerAdjMat(generateGraph2(n,10,20)[[2]],n),neigsToCompute)$values
```

```
## Warning in eigs_real_sym(A, nrow(A), k, which, sigma, opts, mattype =
## "sym_matrix", : all eigenvalues are requested, eigen() is used instead
```

```
l25 <- eigs(centerAdjMat(generateGraph2(n,10,25)[[2]],n),neigsToCompute)$values
```

```
## Warning in eigs_real_sym(A, nrow(A), k, which, sigma, opts, mattype =
## "sym_matrix", : all eigenvalues are requested, eigen() is used instead
```

```
l30 <- eigs(centerAdjMat(generateGraph2(n,10,30)[[2]],n),neigsToCompute)$values
```

```
## Warning in eigs_real_sym(A, nrow(A), k, which, sigma, opts, mattype =
## "sym_matrix", : all eigenvalues are requested, eigen() is used instead
```

```
## Plotting Code
# par(mfrow = c(2,3))
# hist(l10,breaks=seq(min(l10)-0.2,max(l10)+0.2,by=0.2),
# main="Spectral Distribution of A^{cen}_G when a = 10")
# abline(v=max(l10),col="red")
# hist(l15,breaks=seq(min(l15)-0.2,max(l15)+0.2,by=0.2),
# main="Spectral Distribution of A^{cen}_G when a = 15")
# abline(v=max(l15),col="red")
# hist(l20,breaks=seq(min(l20)-0.2,max(l20)+0.2,by=0.2),
# main="Spectral Distribution of A^{cen}_G when a = 20")
# abline(v=max(l20),col="red")
# hist(l25,breaks=seq(min(l25)-0.2,max(l25)+0.2,by=0.2),
# main="Spectral Distribution of A^{cen}_G when a = 25")
# abline(v=max(l25),col="red")
# hist(l30,breaks=seq(min(l30)-0.2,max(l30)+0.2,by=0.2),
# main="Spectral Distribution of A^{cen}_G when a = 30")
# abline(v=max(l30),col="red")
```

Code For Part B

```
specClustSimple <- function(AdjMat,n,mode="Center"){
  if(mode=="Center"){
    cadjMat <- centerAdjMat(AdjMat,n)
  }else{
    cadjMat <- laplacian(AdjMat,n)
  }
  topEigenVec <- Re(eigs(cadjMat,1,which="LR")$vectors[,1])
  labs <- as.numeric(topEigenVec >= 0)
  return(labs)
}

overlapSimple <- function(labels1,labels2){
  meanOver <- max(mean(as.numeric(labels1==labels2)),mean(as.numeric((1-labels1)==labels2)))
  return(2*(meanOver - 1/2))
}
```

```
oneRun <- function(n,b,a){
  G <- generateGraph2(n,b,a)
  specLabs <- specClustSimple(G[[2]],n)
  return(overlapSimple(G[[1]],specLabs))
}

## n = 10000 was used for all results produced in my homework
n <- 100
b <- 10
ntrials <- 20

for(a in c(10,15,20,25,30)){
  #sprintf("Result with value %d",a)
  #print(mean(sapply(1:ntrials,function(x){oneRun(n,b,a)})))
}
```

Part D

Code for computing the desired result is shown below. Note that before running the clustering algorithm I pre-processed the data. In particular, the given data forms a directed multigraph. To make this data consistent with our simulations from parts a-c I removed all the self loops and multiple edges and I turned all the directed edges into undirected edges. This gives me a symmetric adjancency matrix with 0-1 entries and zeros along the diagonal. After doing this I additionally remove all isolated nodes from the graph. By running spectral clustering on the centered adjacency matrix I compute an overlap value of $Q(\sigma, \hat{\sigma}) = 0.302$.

```
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
G <- read.graph("/Users/isaacgibbs/Downloads/polblogs/polblogs.gml",format="gml")
polyAdjMat <- as.matrix(get.adjacency(G))
polyAdjMat <- polyAdjMat + t(polyAdjMat)
polyAdjMat <- pmin(polyAdjMat,1)
for(i in 1:nrow(polyAdjMat)){
  polyAdjMat[i,i] <- 0
}
degs <- polyAdjMat %*% rep(1,nrow(polyAdjMat))
polyAdjMat <- polyAdjMat[degs !=0, degs!=0]
polyLabs <- get.vertex.attribute(G)[[3]]
polyLabs <- polyLabs[degs!=0]
specClustPoly <- specClustSimple(polyAdjMat,nrow(polyAdjMat))
overlapSimple(specClustPoly,polyLabs)
```

```
## [1] 0.3022876
```

Part E

I use the same preprocessing steps as in part d above. By running spectral clustering on the normalized Laplacian I compute an overlap value of $Q(\sigma, \hat{\sigma}) = 0.042$.

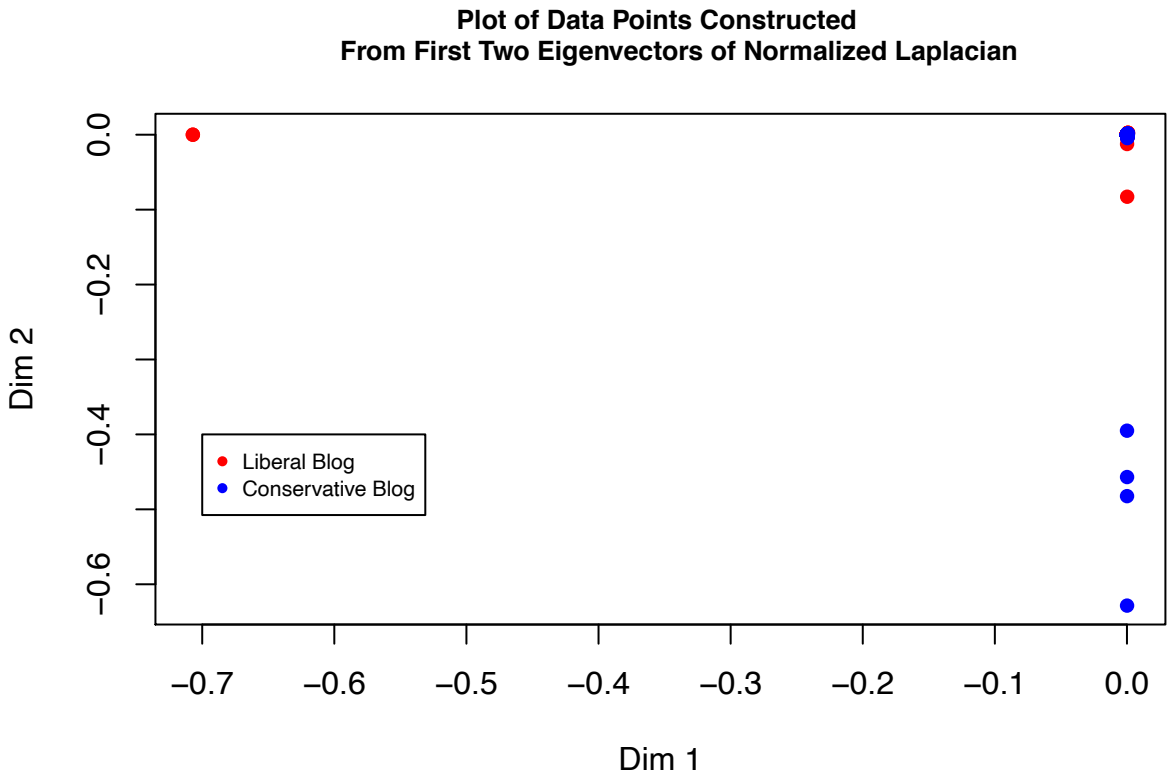
```
specClustPoly <- specClustSimple(polyAdjMat,nrow(polyAdjMat),mode="Laplacian")
overlapSimple(specClustPoly,polyLabs)
```

```
## [1] 0.04248366
```

Part F

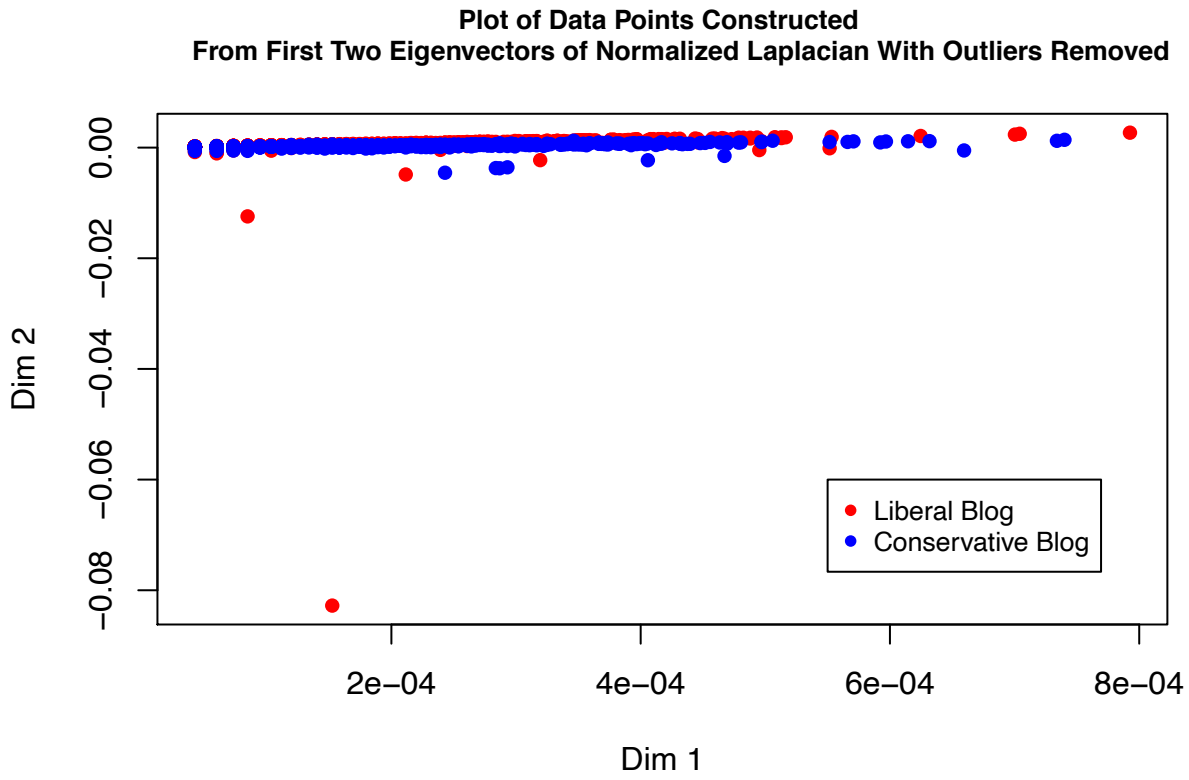
I use the same preprocessing steps as in parts D and E. A plot of the data points $x_1, \ldots, x_n \in \mathbb{R}^2$ computed from the first two eigenvectors of the normalized Laplacian is shown below. Points in the plot are coloured based on the value of their true label.

```
specClustK3 <- function(AdjMat,n,mode="Center"){
  if(mode=="Center"){
    cadjMat <- centerAdjMat(AdjMat,n)
  }else if(mode=="UnNormLaplace"){
        cadjMat <- unNormlaplacian(AdjMat,n)
  }else{
    cadjMat <- laplacian(AdjMat,n)
  }
  topEigenVecs <- Re(eigs(cadjMat,2,which="LR")$vectors)
  return(topEigenVecs)
}
xpoints <- specClustK3(polyAdjMat,nrow(polyAdjMat),mode="Laplacian")
colVec <- as.vector(polyLabs)
colVec[colVec==0] <- "red"
colVec[colVec==1] <- "blue"
plot(xpoints,col=colVec,pch=16,xlab="Dim 1",ylab="Dim 2",
     main="Plot of Data Points Constructed
     From First Two Eigenvectors of Normalized Laplacian",cex.main=0.8)
legend(-0.7,-0.4, legend=c("Liberal Blog", "Conservative Blog"),
       col=c("red", "blue"), pch=16,cex=0.7)
```



There are a few large outlier points on this plot that make it difficult to view the bulk of the data. In order to get a more clear picture of the data we remove these outliers and make a second plot shown below. As you can see the points are not well separated and both left-leaning and right-leaning political blogs appear next to one another.

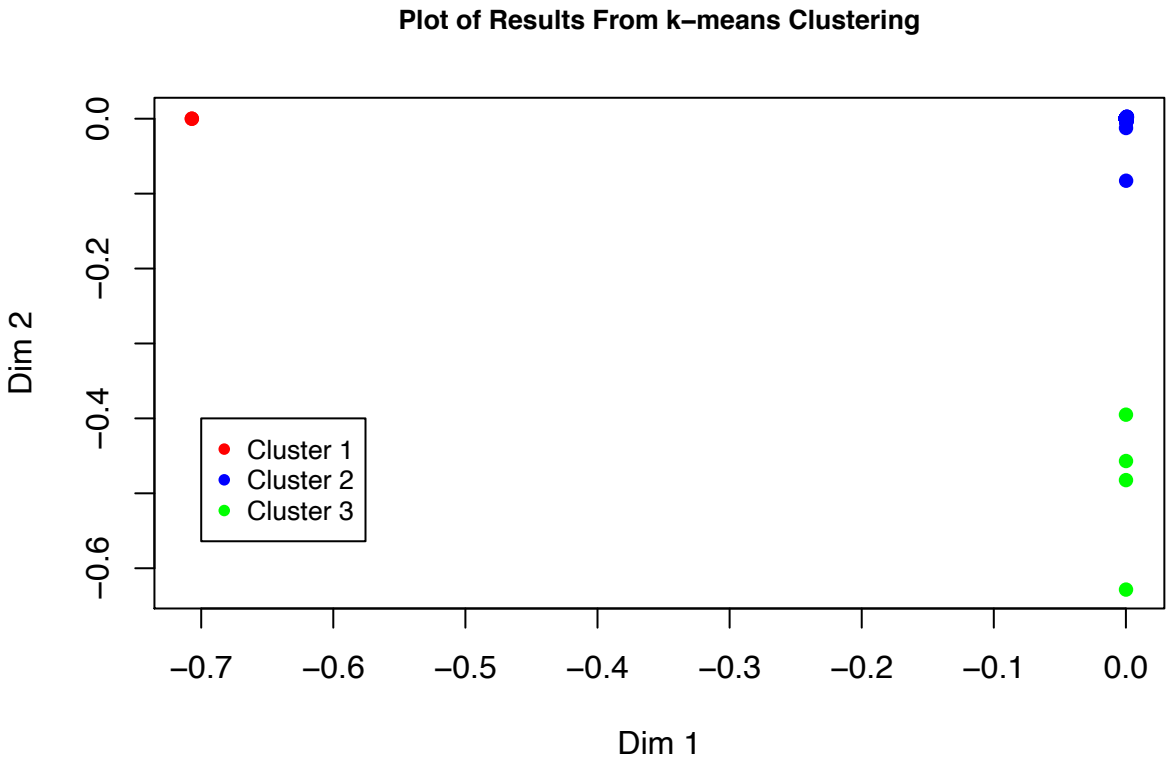
```
plot(xpoints[xpoints[,1]>-0.3 & xpoints[,2] > -0.2,],col=colVec[xpoints[,1]>-0.3 & xpoints[,2] > -0.2],
     pch=16,xlab="Dim 1",ylab="Dim 2",
     main="Plot of Data Points Constructed
     From First Two Eigenvectors of Normalized Laplacian With Outliers Removed",cex.main=0.8)
legend(5.5e-04, -0.06, legend=c("Liberal Blog", "Conservative Blog"),
       col=c("red", "blue"), pch=16, cex=0.8)
```



Finally, we cluster the blogs. A plot of the resulting clusters is shown below. Unsurprisingly, the clustering is uninformative because the two sets of outlier points get their own clusters and all of the rest of the data gets put into a single large cluster. Thus, it seems safe to conclude that the clustering was not successful in identifying meaningful information about the blogs.

If we had obtained a more interesting clustering we could have tried to evaluate it by e.g. looking at the maximum value of the overlap after merging two of the three clusters together. However, since our clustering is so poor it is safe to conclude that the clustering was not successful without any further analysis.

```
clust <- kmeans(xpoints,3)
colVec2 <- as.vector(clust$cluster)
colVec2[colVec2=="1"] <- "red"
colVec2[colVec2=="2"] <- "blue"
colVec2[colVec2=="3"] <- "green"
plot(xpoints,col=colVec2,pch=16,xlab="Dim 1",ylab="Dim 2",
     main = "Plot of Results From k-means Clustering", cex.main=0.8)
legend(-0.7,-0.4, legend=c("Cluster 1", "Cluster 2", "Cluster 3"),
       col=c("red", "blue", "green"), pch=16, cex=0.8)
```


Plot of Results From k–means Clustering
Dim 2
0.0
−0.2
−0.4
−0.6
Cluster 1
Cluster 2
Cluster 3
−0.7
−0.6
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
Dim 1