

ST 790 Assignment 2

David Elsheimer and Jimmy Hickey

9/29/2020

Instruction

This assignment consists of 4 problems. The assignment is due on **Tuesday, September 29** at 11:59pm EDT. Please submit your assignment electronically through the **Moodle** webpage. The assignment can be done as a group with at most 3 members per group (please include the name of the group members on the front page of the assignment).

Problem 1.

Let \mathbf{D} be a $n \times n$ symmetric dissimilarity matrix, i.e., the entries d_{ij} of \mathbf{D} are non-negative for all i, j , $d_{ii} = 0$ for all i . Let $\mathbf{B} = -\frac{1}{2}(\mathbf{I} - \mathbf{1}\mathbf{1}^\top/n)\mathbf{D}(\mathbf{I} - \mathbf{1}\mathbf{1}^\top/n)$ where \mathbf{I} is the $n \times n$ identity matrix and $\mathbf{1} \in \mathbb{R}^n$ is the vector of all ones.

Given an integer $r \leq n$, the classical multidimensional scaling (CMDS) procedure produces a configuration $\mathbf{Z}_* \in \mathbb{R}^{n \times r}$ representing n points in \mathbb{R}^r that best approximate \mathbf{D} with respect to the STRAIN criterion, i.e.,

$$\mathbf{Z}_* = \arg \min_{\mathbf{Z} \in \mathbb{R}^{n \times r}} \|\mathbf{B} - \mathbf{Z}\mathbf{Z}^\top\|_F. \quad (1)$$

Consider the following eigendecomposition of \mathbf{B} .

$$\mathbf{B} = \sum_{i=1}^n \lambda_i u_i u_i^\top; \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$$

Then the CMDS procedure returns the $n \times r$ configuration

$$\mathbf{Z}_* = \left[(\lambda_1)_+^{1/2} u_1, (\lambda_2)_+^{1/2} u_2, \dots, (\lambda_r)_+^{1/2} u_r \right] \quad (2)$$

where $(\lambda_i)_+ = \max\{\lambda_i, 0\}$.

Show that the expression \mathbf{Z}_* in Eq.(2) is really the minimizer of the STRAIN criterion for \mathbf{B} . More specifically, show the following result.

Proposition Let \mathbf{B} be a $n \times n$ symmetric matrix with eigendecomposition

$$\mathbf{B} = \sum_{i=1}^n \lambda_i u_i u_i^\top; \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$$

Then the best rank r , positive semidefinite approximation to \mathbf{B} , with respect to the Frobenius norm, is the matrix

$$\mathbf{B}_r = \sum_{i=1}^r (\lambda_i)_+ u_i u_i^\top.$$

Hint: Let \mathbf{M} be any $n \times n$ matrix. Then

$$\|\mathbf{B} - \mathbf{M}\|_F^2 = \|\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top - \mathbf{V}\mathbf{\Sigma}\mathbf{V}^\top\|_F^2 = \|\mathbf{\Lambda} - \mathbf{W}\mathbf{\Sigma}\mathbf{W}^\top\|_F^2$$

where $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ and $\mathbf{V}\mathbf{\Sigma}\mathbf{V}^\top$ are the eigendecomposition of \mathbf{B} and \mathbf{M} , respectively, and \mathbf{W} is a $n \times n$ orthogonal matrix.

Now, for fixed diagonal matrices $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$, show that a minimizer of

$$\min_{\mathbf{W}} \|\mathbf{\Lambda} - \mathbf{W}\mathbf{\Sigma}\mathbf{W}^\top\|_F^2$$

over the set of orthogonal matrices \mathbf{W} is given by a permutation matrix. Using the [rearrangement inequality](#), show that this permutation matrix will rearrange the diagonal entries of $\mathbf{\Sigma}$ to coincide with the ordering of the diagonal entries of $\mathbf{\Lambda}$.

Problem 2

Download the dataset of games between American College Football teams available [here](#). Now perform community detection using (1) a spectral clustering algorithm using normalized cut (2) spectral clustering using modularity and (3) Louvain algorithm. Evaluate the performance of your clustering (compared to the given ground truth).

```
g<-read.graph("football.gml",format=c("gml"))

g2 <- as.undirected(g)
## Largest connected component
lcc <- decompose.graph(g2, min.vertices = max(components(g2)$csize))[[1]]
Xhat2 <- embed_laplacian_matrix(lcc, type = "I-DAD", which = "sa", no = 3, scaled = FALSE)

Xhat2$D ## The three smallest eigenvalue of the normalized Laplacian

## [1] 0.0000000 0.1368043 0.1829191
table(vertex_attr(lcc)$value) ## Ground truth

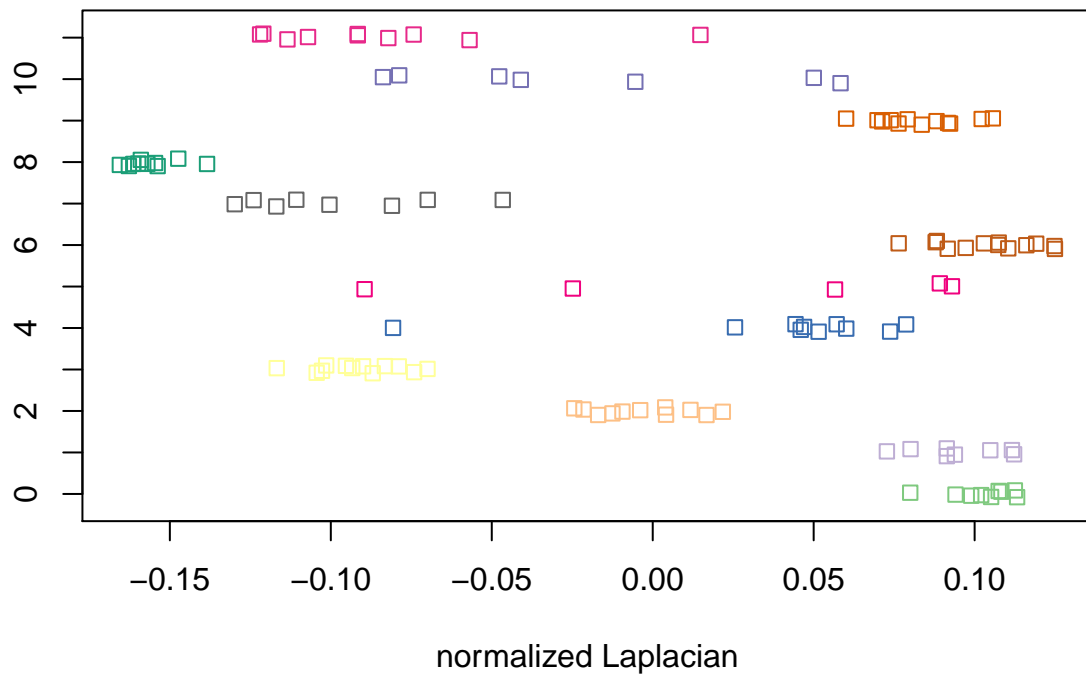
##
## 0  1  2  3  4  5  6  7  8  9 10 11
## 9  8 11 12 10  5 13  8 10 12  7 10

gt <- vertex_attr(lcc)$value
gt

##      [1]  7  0  2  3  7  3  2  8  8  7  3 10  6  2  6  2  7  9  6  1  9  8  8  7 10
##     [26]  0  6  9 11  1  1  6  2  0  6  1  5  0  6  2  3  7  5  6  4  0 11  2  4 11
##     [51] 10  8  3 11  6  1  9  4 11 10  2  6  9 10  2  9  4 11  8 10  9  6  3 11  3
##     [76]  4  9  8  8  1  5  3  5 11  3  6  4  9 11  0  5  4  4  7  1  9  9 10  3  6
##    [101]  2  1  3  0  7  0  2  3  8  0  4  8  4  9 11

### Getting a vector of distinct colors for use in normalized cut
n <- 12
qual_col_pals = brewer.pal.info[brewer.pal.info$category == 'qual',]
col_vector = unlist(mapply(brewer.pal, qual_col_pals$maxcolors, rownames(qual_col_pals)))
col_vector<- col_vector[1:12]

stripchart(Xhat2$X[,2] ~ vertex_attr(lcc)$value, method = "jitter",
col = col_vector[1:12], xlab = "normalized Laplacian")
```



```
clusters2 <- kmeans(Xhat2$X, centers = 12)

table(clusters2$cluster, vertex_attr(lcc)$value)
```

```
##
##      0  1  2  3  4  5  6  7  8  9 10 11
##  1    0  0  0  0  0  0  0  1  0  0  3  1
##  2    8  0  0  0  0  0  0  0  0  0  0  0
##  3    0  0  0  5  0  0  0  0  0  0  0  0
##  4    0  0  0  0  5  0  0  0  0  0  0  0
##  5    0  8  0  0  0  3  0  0  0  0  0  0
##  6    0  0  0  0  0  0 13  0  0  0  0  0
##  7    1  0  0  0  4  0  0  0  0  0  2  0
##  8    0  0  0  3  1  1  0  3  0  0  2  5
##  9    0  0  0  0  0  0  0  0 10  0  0  0
## 10    0  0  0  0  0  0  0  0  0 12  0  0
## 11    0  0 11  0  0  1  0  0  0  0  0  0
## 12    0  0  0  4  0  0  0  4  0  0  0  4
```

```
###louvain
cc <- cluster_louvain(g2)

table(cc$membership)
```

```
##
##  1  2  3  4  5  6  7  8  9 10
## 12 11  9 15  9 14 10 16  9 10
```

```

###Normal cut comparison
## Compare the clustering using normalized mutual information
compare(clusters2$cluster, gt, method = c("nmi"))

## [1] 0.796521

## Compare the clustering using Rand index
compare(clusters2$cluster, gt, method = c("rand"))

## [1] 0.9470633

## Compare the clustering using adjusted Rand index
compare(clusters2$cluster, gt, method = c("adjusted.rand"))

## [1] 0.6507787

###Louvain comparison
## Compare the clustering using normalized mutual information
compare(cc$membership, gt, method = c("nmi"))

## [1] 0.8903166

## Compare the clustering using Rand index
compare(cc$membership, gt, method = c("rand"))

## [1] 0.9688787

## Compare the clustering using adjusted Rand index
compare(cc$membership, gt, method = c("adjusted.rand"))

## [1] 0.8069409

```

Problem 2.5 (optional)

Now try to do community detection on the Youtube network dataset available [here](#) using any algorithm you want. Now try to evaluate the performance of your clustering compared to the given ground truth (it is not going to be easy :-))

Problem 3.

Download the USPS digits dataset from [here](#). Next do the following steps.

- Choose the digits for class 4, 7 and 9.
- Randomly sample 800 digits from each class.
- Do an embedding of these 2400 data points into \mathbb{R}^d for some d , say $d = 10$, using **Laplacian eigenmaps**. You are free to choose your choice of ϵ or k -NN or weights.
- Run a 3-class SVM classifier on the embeddings for these 2400 datapoints. Use a linear kernel for your SVM.
- With the remaining 900 data points (300 from each class), evaluate the performance of your SVM classifier on this holdout data.
- Compare this performance with say SVM classification in the original 256 dimension data using say a linear kernel or a Gaussian kernel.

Hint To evaluate the performance on the hold-out data, you will need to do an out-of-sample embedding. Read [this paper](#) to see how to construct an out-of-sample embedding for Laplacian eigenmaps.

Problem 4.

Complete the proof of Theorem 7 on page 55 of the dimension reduction lecture slides. In particular, show that if x_1, x_2, \dots, x_n are **distinct** then the matrix $\mathbf{H} = f(\|x_i - x_j\|^2)$ is positive definite.