Graph Wavelets for Multiscale Community Mining

Final Project Report
Ilia Igashov

February 2021

1 Introduction

In this project, I worked on the application of wavelets in the graph domain in the scope of the community mining problem. Multiscale community mining is a task for determining regions of a graph where nodes are well connected with each other (in different scales). I studied the method for multiscale community mining based on graph wavelets [1]. I implemented the proposed method in Python and reproduced results obtained by authors on Sales-Pardo graphs [2]. Also, I experimented with multiscale mining on swiss roll manifolds with Gaussian mixtures.

2 Method

The proposed multiscale community mining approach consists of the following steps.

1. Compute wavelets on nodes of the graph.
2. Using wavelets as features of nodes, compute a distance matrix of the graph (correlation distances between wavelet vectors of nodes).
3. Apply a hierarchical clustering algorithm to the distance matrix and choose the best partition.

Calculating wavelets on big graphs (e.g., ten thousand nodes) is a computationally complicated problem. In this work, authors propose a novel statistical approach by estimating directly correlation distances between nodes. This makes the whole method much faster and more scalable on big graphs without any significant loss of quality.

2.1 Notations

We will consider a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ with the set of nodes $\mathcal{V}$, the set of edges $\mathcal{E}$, and the adjacency matrix $\mathbf{A}$. Denote the total number of nodes as $N = |\mathcal{V}|$. We will define the graph's Laplacian matrix as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the diagonal matrix containing degrees of nodes of the graph $\mathcal{G}$: $\mathbf{D}_{ii} = \mathbf{d}_i = \sum_{j \neq i} \mathbf{A}_{ij}$. The normalized Laplacian, which is $\mathcal{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}$, is a real symmetric matrix, and therefore diagonalizable. Denonte the set of the normalized Laplacian's eigenvalues as $\{\lambda_l\}_{l=1}^N : \ 0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N \leq 2$ [3], and the matrix of the corresponding eigenvectors as $\mathbf{U} = (\mathbf{u}_1 | \dots | \mathbf{u}_N)$. We will also consider the signal $\boldsymbol{f}$ defined on the nodes of the graph $\mathcal{G}$.

2.2 Graph Wavelets

The whole theory for graph wavelets that was used in this work is introduced in [4]. Let us denote the wavelet at scale $s \in \mathbb{R}_+$ centered around node $a \in \mathcal{V}$ as $\boldsymbol{\psi}_{s,a}$. Its construction is based on bandpass filters defined in the graph Fourier domain, generated by stretching a unique band-pass wavelet filter kernel $g(\cdot)$ by a scale parameter $s > 0$. In this work, we will consider the following band-pass filter kernel:

$$g(x; \alpha, \beta, x_1, x_2) = \begin{cases} x_1^{-\alpha} x^{\alpha} & \text{for } x < x_1, \\ p(x) & \text{for } x_1 \leq x \leq x_2, \\ x_2^{\beta} x^{-\beta} & \text{for } \ x > x_2, \end{cases} \tag{1}$$

where $p(x)$ is taken as the unique cubic polynomial interpolation that respects the continuity of g and its derivative g'. The integers α and β, and the transition points x_1 and x_2 are the parameters of the filter.

The stretched filter has a matrix representation $\mathbf{G}_s = \mathrm{diag}(g(s\lambda_1), \dots, g(s\lambda_N))$ that is diagonal on eigenvectors of $\mathcal{L}$. Then the wavelet basis at scale s can be written as

$$\boldsymbol{\Psi}_s = (\boldsymbol{\psi}_{s,1} | \boldsymbol{\psi}_{s,2} | \dots | \boldsymbol{\psi}_{s,N}) = \mathbf{U} \mathbf{G}_s \mathbf{U}^{\top}, \tag{2}$$

and the wavelet coefficient at scale s and node a of the signal $\boldsymbol{f}$ has the following form,

$$W_f(s, a) = \boldsymbol{\psi}_{s,a}^{\top} \boldsymbol{f}. \tag{3}$$

2.3 Correlation Distances

The aim of the community mining task is to detect groups of nodes with topologically similar environments. Since the local neighborhood information of a nodes is encoded in wavelets, for each node a, we can consider the corresponding wavelet $\psi_{s,a}$ as a relevant descriptor that can help in community mining at a given scale s. To compare two nodes a and b, one can compute correlation between the corresponding wavelets:

$$\mathbf{D}_s(a,b) = 1 - \frac{\boldsymbol{\psi}_{s,a}^\top \boldsymbol{\psi}_{s,b}}{\left\|\boldsymbol{\psi}_{s,a}\right\|_2 \left\|\boldsymbol{\psi}_{s,b}\right\|_2}. \tag{4}$$

However, computation of N wavelets for big values of N can be infeasible. Moreover, in fact, we finally need only the correlation matrix $\mathbf{D}_s$. Thus, authors proposed an efficient way of direct estimating the correlation matrix using a finite number of random vectors.

Let us consider a random vector $\boldsymbol{r} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma^2, \sigma^2, \ldots, \sigma^2)$. For scale s and node a, we can define the feature $f_{s,a} \in \mathbb{R}$ as a projection of the vector $\boldsymbol{r}$ on the wavelet $\boldsymbol{\psi}_{s,a}$:

$$f_{s,a} = \boldsymbol{\psi}_{s,a}^\top \boldsymbol{r}. \tag{5}$$

Consider the correlation between features of nodes a and b:

$$\operatorname{Cov}(a,b) = \frac{\mathbb{E}\left[(f_{s,a} - \mathbb{E}(f_{s,a}))(f_{s,b} - \mathbb{E}(f_{s,b}))\right]}{\sqrt{\operatorname{Var}(f_{s,a})\operatorname{Var}(f_{s,b})}}. \tag{6}$$

Taking into account that the mean of a node's feature equals to zero:

$$\mathbb{E}(f_{s,a}) = \mathbb{E}(\boldsymbol{\psi}_{s,a}^\top \boldsymbol{r}) = \boldsymbol{\psi}_{s,a}^\top \mathbb{E}(\boldsymbol{r}) = 0, \tag{7}$$

the variance can be simplified as follows,

$$\begin{aligned} \operatorname{Var}(f_{s,a}) &= \mathbb{E}\left[(f_{s,a} - \mathbb{E}(f_{s,a}))^2\right] \\ &= \mathbb{E}f_{s,a}^2 - \left(\mathbb{E}f_{s,a}\right)^2 \\ &= \mathbb{E}f_{s,a}^2 = \boldsymbol{\psi}_{s,a}^\top \mathbb{E}(\boldsymbol{r}^\top \boldsymbol{r}) \boldsymbol{\psi}_{s,a} = \sigma^2 \left\|\boldsymbol{\psi}_{s,a}\right\|_2^2, \end{aligned} \tag{8}$$

and that

$$\begin{aligned} \mathbb{E}(f_{s,a} f_{s,b}) = \mathbb{E}\left[(\boldsymbol{\psi}_{s,a}^\top \boldsymbol{r})(\boldsymbol{\psi}_{s,b}^\top \boldsymbol{r})\right] = \mathbb{E}\left[(\sum_{k=1}^{N} \psi_{s,a}(k) r(k))(\sum_{k'=1}^{N} \psi_{s,b}(k') r(k'))\right] \\ = \sum_{k \neq k'} \psi_{s,a}(k)\psi_{s,b}(k')\mathbb{E}\left[r(k)r(k')\right] + \sum_{k=1}^{N} \psi_{s,a}(k)\psi_{s,b}(k)\mathbb{E}\left[r(k)^2\right] = \sigma^2 \boldsymbol{\psi}_{s,a}^\top \boldsymbol{\psi}_{s,b}, \end{aligned} \tag{9}$$

we get:

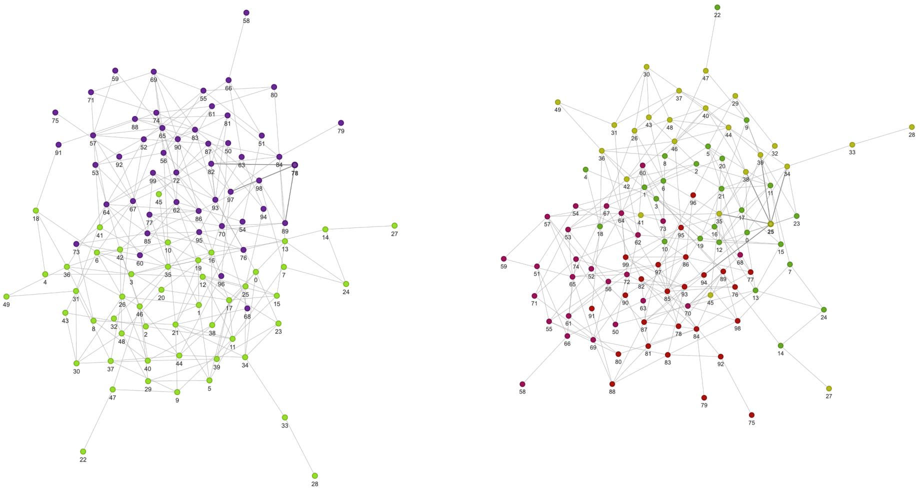

Figure 1: Example of a toy Sales-Pardo graph with two scales. At the first scale (on the left), we defined communities of size 50, at the second scale (on the right), we defined comminities of size 25.

$$\mathrm{Cov}(a,b) = \frac{\mathbb{E}\left[f_{s,a} f_{s,b}\right]}{\sigma^2 \left\|\boldsymbol{\psi}_{s,a}\right\|_2 \left\|\boldsymbol{\psi}_{s,b}\right\|_2} = \frac{\boldsymbol{\psi}_{s,a}^\top \boldsymbol{\psi}_{s,b}}{\left\|\boldsymbol{\psi}_{s,a}\right\|_2 \left\|\boldsymbol{\psi}_{s,b}\right\|_2}. \tag{10}$$

Thus, we see that the correlation between features $f_{s,a}$ and $f_{s,b}$ is exactly the correlation between wavelets $\boldsymbol{\psi}_{s,a}$ and $\boldsymbol{\psi}_{s,b}$. And it means that instead of computing formula (4), we can estimate the covariance matrix with entries provided by (6) using a sample of independent normally distributed random vectors.

Consider η realisations of the random vector $\boldsymbol{r}$ written as a matrix $\boldsymbol{R} = (\boldsymbol{r}_1|\boldsymbol{r}_2|\ldots|\boldsymbol{r}_\eta)$. Having η realisations of a random vector $\boldsymbol{r}$, for each scale s and node a we now have η features $f^i_{s,a} = \boldsymbol{\psi}_{s,a}^\top \boldsymbol{r}_i,\ i \in \{1,\ldots,\eta\}$ that can be written as a feature vector $\boldsymbol{f}_{s,a}^\top = \boldsymbol{\psi}_{s,a}^\top \boldsymbol{R}$.

The sample correlation coefficient estimator between $\boldsymbol{f}_{s,a}$ and $\boldsymbol{f}_{s,b}$ can be written as

$$\hat{C}_\eta(a,b) = \frac{(\boldsymbol{f}_{s,a} - \bar{\boldsymbol{f}}_{s,a})^\top (\boldsymbol{f}_{s,b} - \bar{\boldsymbol{f}}_{s,b})}{\left\|\boldsymbol{f}_{s,a} - \bar{\boldsymbol{f}}_{s,a}\right\|_2 \left\|\boldsymbol{f}_{s,b} - \bar{\boldsymbol{f}}_{s,b}\right\|_2}, \tag{11}$$

where $\bar{\boldsymbol{f}}_{s,a} = \frac{1}{\eta}\mathbb{I}^\top \boldsymbol{f}_{s,a}\mathbb{I}$, and $\mathbb{I}$ is a constant vector equal to 1.

Since $f_{s,a}$ and $f_{s,b}$ are jointly Gaussian, the estimator $\hat{C}_\eta$ is asymptotically consistent, i.e.

$$\lim_{\eta\to+\infty} \hat{C}_\eta(a,b) = \mathrm{Cor}(f_{s,a}, f_{s,b}) = \frac{\boldsymbol{\psi}_{s,a}^\top \boldsymbol{\psi}_{s,b}}{\left\|\boldsymbol{\psi}_{s,a}\right\|_2 \left\|\boldsymbol{\psi}_{s,b}\right\|_2}, \tag{12}$$

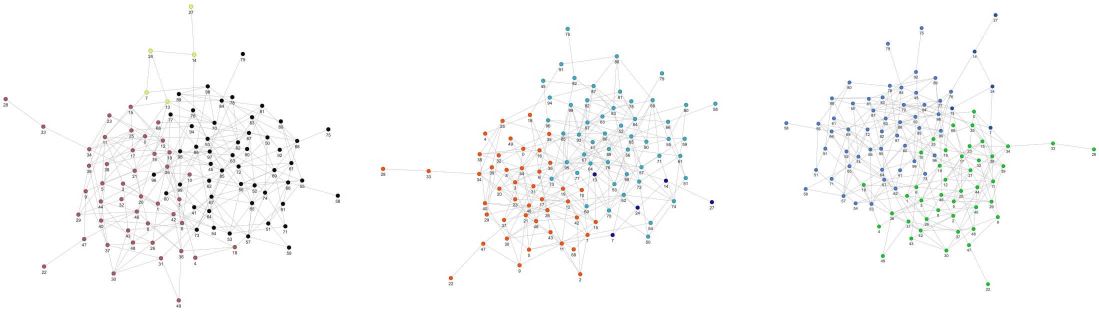

Figure 2: Communities computed for the toy Sales-Pardo graph by the proposed algorithm for three different scales.

and therefore

$$\lim_{\eta \to +\infty} 1 - \hat{C}_\eta(a, b) = \mathbf{D}_s(a, b). \tag{13}$$

2.4 Clustering

In order to detect communities, authors apply a hierarchical "average-linkage" clustering algorithm [5, 6] on top of the distance matrix $\mathbf{D}_s$ acquired in the previous step. The clustering algorithm builds a dendrogram of the graph, and the main question that arises is where to cut the dendrogram. Authors address this problem by introducing a novel criterion based on averaging the maximal gaps of all the root-leaf paths of the dendrogram.

Let us consider a node a and define its dendrogram-path: it is the path between the leaf of the dendrogram corresponding to node a and the root of the dendrogram (the node of the dendrogram that has the highest correlation distance). For this node a, one can plot its gap function Γ_a built in the following way: follow the dendrogram-path starting at zero correlation distance. For each correlation distance, the path is between two dendrogram nodes: plot the gap between them. By averaging all gap functions corresponding to all nodes, one obtains the global gap function [1]:

$$\Gamma = \frac{1}{N \max(\text{corr. dist.})} \sum_{a \in \mathcal{V}} \Gamma_a. \tag{14}$$

Following the gap statistics intuition [7], authors considered that the best possible partition given this dendrogram was obtained by cutting the dendrogram at the maximum of Γ.

Repeating this clustering procedure for different scales s, we obtain the multiscale set of partitions.

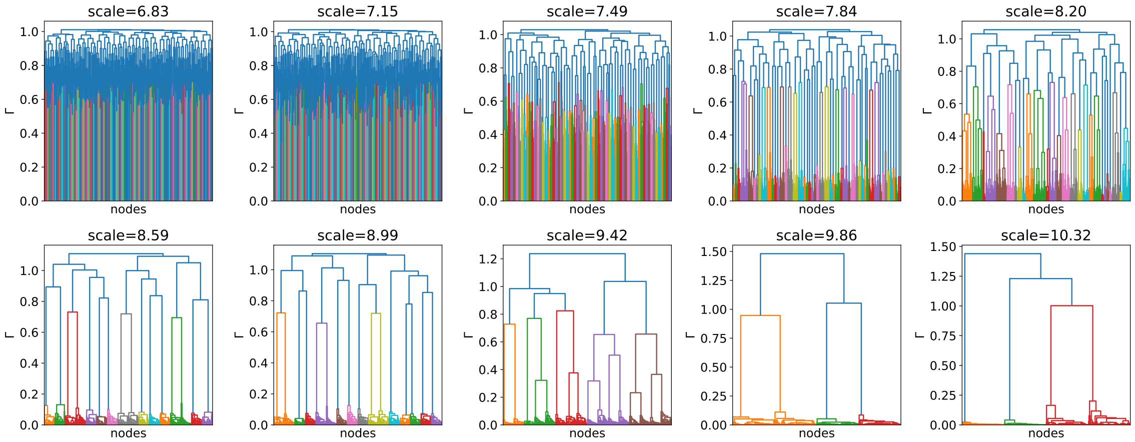

Figure 3: Dendrograms built by the "average-linkage" clustering algorithm based on correlation matrices estimated for ten different scales.

3 Experiments

In order to test the method, I considered two examples: Sales-Pardo graphs [2] and a graph built on a swiss roll manifold with Gaussian mixtures.

3.1 Sales-Pardo Graphs

Sales-Pardo graph [2] is a very convenient benchmark for testing graph clustering algorithms and especially multiscale community mining techniques. Sales-Pardo graph is parametrized by two values: ρ and $\bar{k}$. Parameter ρ is responsible for sparseness of scales, and parameter $\bar{k}$, the average degree, controls how dense the graph is. In order to create a graph, one needs to define the number of scales (we consider three scales), sizes of groups within each scale N_1, N_2, and N_3, and parameters ρ and $\bar{k}$. Natural constraints that should hold for parameters are the following:

$$N_3 < N_2 < N_1, \tag{15}$$

$$\exists l_1, l_2 \in \mathbb{N}: \; N_1 = l_1 N_2 = l_2 N_3, \tag{16}$$

$$\frac{\bar{k}}{1+\rho} \leq S_3, \tag{17}$$

where taking any node i, we define S_3 as the number of nodes (different than i) that are in i's small community hold three common community memberships with i: $S_3 = N_3 - 1$.

To begin with, I generated a small Sales-Pardo graph with $N = 100$ nodes and $m = 2$ scales, groups' sizes $N_1 = 50$, $N_2 = 25$, and parameters $\rho = 0.5$, $\bar{k} = 5$. The graph and its communities (for both scales) are illustrated in Figure 1. Examples of communities that were computed by the algorithm proposed in [1] are provided in Figure 2 (for three different scales).

Next, I generated a graph with the same parameters as authors provided in order to reproduce the

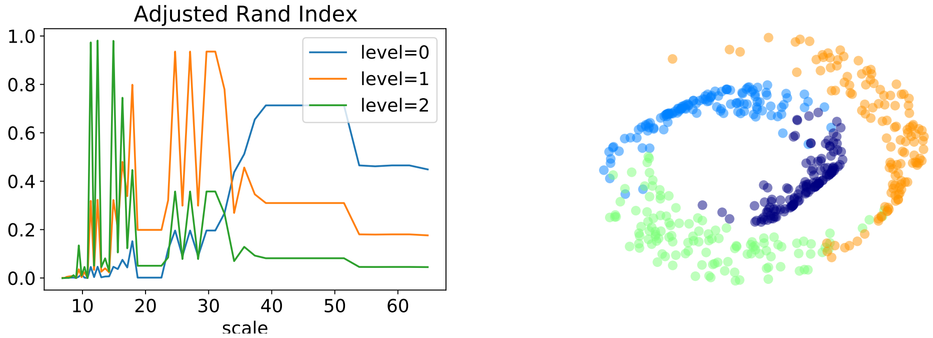

Figure 4: (left) The plot of Adjusted Rand Index. Levels correspond to initial scales: “level=1” corresponds to communities with size $N_1 = 160$, “level=2” – to communities with size $N_2 = 40$, etc. (right) Gaussian mixture on a a non-uniform swiss roll manifold (4 clusters).

experiment described in the paper. Namely, for a Sales-Pardo graph with parameters $N = 600,\ m = 3,\ N_1 = 160,\ N_2 = 40,\ N_3 = 10,\ \rho = 1,$ and $\bar{k} = 16$, I built communities by the algorithm with number of scales 50 and number of random vectors $\eta = 60$. Dendrograms built by the ”average-linkage” clustering algorithm based on correlation matrices estimated for ten different scales are represented in Figure 3. We can clearly see that number of clusters reduces as the value of scale grows. In order to establish this dependency more clearly, we can compute a partition similarity measure Adjusted Rand Index [1]. We will compute this metrics for each of 50 partitions we got and for each of three initial partitions. The plot in Figure 4 (left) clearly demonstrates how computed communities from different scales correspond to initial three communities.

3.2 Swiss Roll Manifold

Another graph example is a non-uniformly sampled swiss roll manifold. Points on this manifold follow the one of C normal distributions (with equal variances) which are equiprobable. Having such manifold, one can build a weighted graph by connecting the points on a manifold by edges with the following weights:

$$A_{ij} = \exp - \frac{\|x_i - x_j\|^2}{2\sigma^2}, \tag{18}$$

where $x_i, x_j \in \mathbb{R}^3$ are coordinates of points i and j, and σ^2 is a hyperparameter. Figure 4 (right) illustrates points generated on a swiss roll manifold with $C = 4$ clusters. Applying the multiscale communities mining algorithm for 10 scales with $\eta = 10$ random vectors, we can see that clusters can be detected well if to properly adjust the scale value. Figure 5 represents clustering results along with the corresponding dendrograms for different scales.

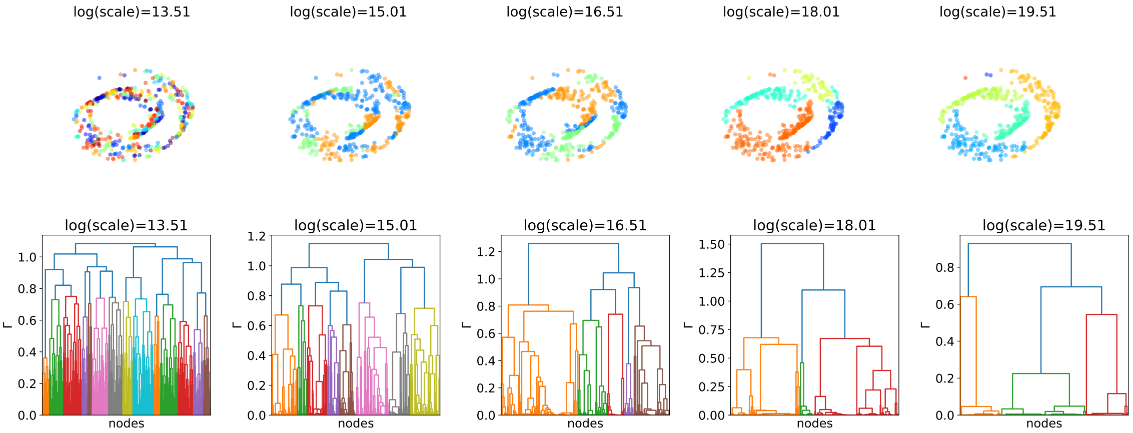

Figure 5: Results of the multiscale community mining algorithm with $\eta = 10$ for the suiss manifold example.

4 Conclusion

In this project, I studied and implemented the multiscale community mining method based on graph wavelets [1]. Authors proposed a procedure of estimating the correlation matrix between nodes using a finite number of random vectors. I reproduced experiments provided in the original paper by testing the method on Sales-Pardo graphs and on a non-uniform suiss roll manifold. Results of experiments demonstrated theat the proposed method detects regions of well-connected nodes in a graph with high quality being at the same time computationally efficient and lightweight.

References

[1] Nicolas Tremblay and Pierre Borgnat. Graph wavelets for multiscale community mining. *IEEE Transactions on Signal Processing*, 62(20):5227–5239, 2014.

[2] Marta Sales-Pardo, Roger Guimera, André A Moreira, and Luís A Nunes Amaral. Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39):15224–15229, 2007.

[3] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[4] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[5] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[6] Benjamin King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 62(317):86–101, 1967.

[7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.