Fairness in Unsupervised Learning

A M. Tech. Report Submitted
in Partial Fulfillment of the Requirements
for the Degree
of

Master of Technology

by

Neha Afreen
(214101034)

under the guidance of

V. Vijaya Saradhi

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM

CERTIFICATE

*This is to certify that the work contained in this thesis entitled "**Fairness in Unsupervised Learning**" is a bonafide work of **Neha Afreen (Roll No. 214101034)**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **V. Vijaya Saradhi**

Assistant Professor,

Nov, 2022	Department of Computer Science & Engineering,
Guwahati.	Indian Institute of Technology Guwahati, Assam.

Abstract

Clustering algorithms are a class of unsupervised machine learning (ML) algorithms that is widely applied in modern data science, and play a key role in many application pipelines. Recently ML community are focusing on learning the fairness in clustering algorithms. Fairness in clustering algorithms depends on the choice of algorithm, fairness definitions employed, and other assumptions made regarding models. Furthermore in many real world problems we are dealing with high dimensional databases for which we have little prior knowledge. However domain knowledge will lead to a better performance in many cases. This emerges the need for constrained clustering algorithms. This paper will give a basic intuition behind the working of different clustering algorithms like k-Means and spectral clustering followed by various fairness attributes and definitions and how to incorporate those definitions into vanilla clustering algorithms. And also a brief about constrained and flexible spectral clustering algorithms and their performances.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In real world scenario while dealing with large data without having a prior knowledge of data set, grouping of similar kinds of data is often done to understand the relationship between the data points and to take more insight about the data. Data points are divided into a number of clusters where data inside one cluster holds a strong similarity between them while inter clustering similarity of data points remains low. Clustering algorithms are widely used in different real world scenario.

1.1 K Means Clustering

K means clustering is a partition based clustering algorithm where data set is divided into a user defined number of clusters, k.

1. **Basic K Means Algorithm :**

    (a) Select K points as initial centroids.

    (b) Repeat

    - Form K clusters by assigning each point to its closest centroid.
    - Recompute the centroid of each cluster.

    (c) until Centroids do not change

Initially we take k points from the data as centroids. Then we try to fit all the data points into k clusters by minimising sum of squared error(SSE) between the centroids and data points. Then centroid of each cluster is updated and again data points are distributed. These two steps are repeated until no data point is changing its cluster.

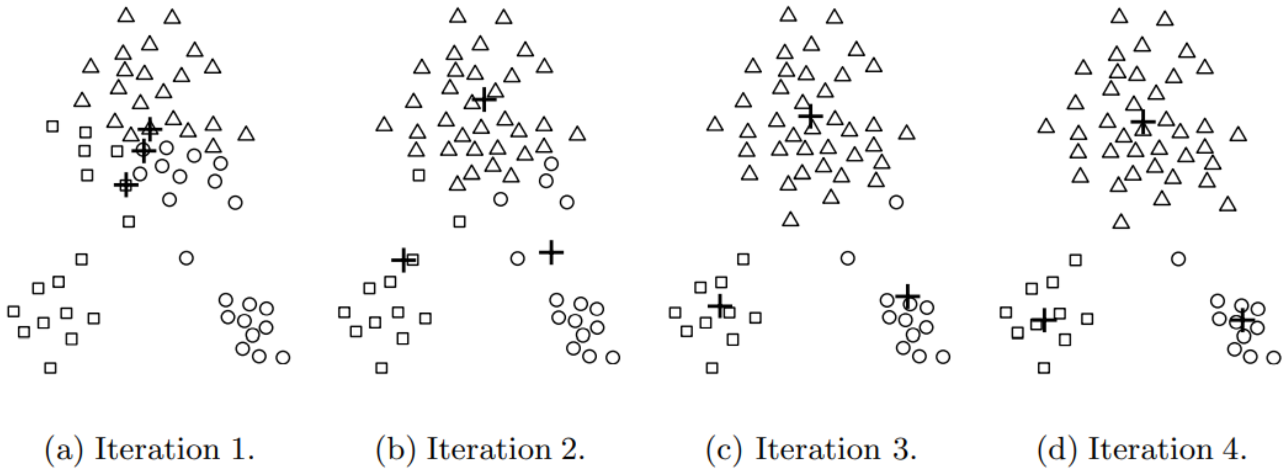

(a) Iteration 1.
(b) Iteration 2.
(c) Iteration 3.
(d) Iteration 4.

Fig. 1.1 Using the K-means algorithm to find three clusters in sample dat

2. **Derivation of K-means as an Algorithm to Minimize the SSE:**

The goal of the clustering is typically expressed by an objective function that depends on the proximities of the points to one another or to the cluster centroids; e.g., minimize the squared distance of each point to its closest centroid. K Means algorithm can be mathematically derived when proxity measure between the data points is Euclidean distance and Objective function is to minimise sum of squared error(SSE).

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} (c_i - x)^2 \tag{1.1}$$

Differentiating and setting to 0

$$SSE = \frac{\partial}{\partial c_k} \sum_{i=1}^{K} \sum_{x \in C_i} (c_i - x)^2 \tag{1.2}$$

$$= \sum_{i=1}^{K} \sum_{x \in C_i} \frac{\partial}{\partial c_k} (c_i - x)^2 = \sum_{x \in C_k} 2 * (c_i - x_k) = 0 \tag{1.3}$$

$$2 * \sum_{x \in C_k} (c_i - x_k) = 0 \implies m + c_k = \sum_{x \in C_k} x_k \implies c_k = \frac{1}{m_k} \sum_{x \in C_k} x_k \tag{1.4}$$

Thus the best centroid for minimizing SSE is mean of the points in the cluster.

3. **Choosing Initial Centroids :**

   Randomly choosen centroids often gives poor clusters. One technique to overcome this problem is to run multiple iterations with different set of initial centroids and then select centroids which gives the lowest SSE. But this technique may not completely overcome the problem as this depends on the number of clusters and also on the data set hence a different and better approach is needed to be employeed depending upon the data set. One of the method is to take sample of points and cluster them using a hierarchical clustering technique. K clusters are extracted from the hierarchical clustering, and the centroids of those clusters are used as the initial centroids.

4. **Outliers :**

   Outliers often affect the performance of K means algorithm. In particular, when outliers are present, the resulting cluster centroids may not be as representative as they otherwise would be and thus, the SSE will be higher as well. In order to deal with this situation outliers are generally identified and removed from the data set. However in some cases removing outliers might not make sense.

5. **Time and Space Complexity :**

   The Storage required is O((m + K)n), where m is the number of points and n is the number of attributes.

   The time requirements for K-means are also modest—basically linear in the number of data points. In particular, the time required is O(I K mn), where I is the number of iterations required for convergence.

1.2 Spectral Clustering

Spectral Clustering often outperforms different simple and traditional clustering algorithms. It uses standard linear algebra tools to cluster data points. When someone look into its working it is not very intuitive to understand how it actually works. So lets discuss briefly about its working.

1. **Similarity Graphs :**

   The very first step in spectral clustering is to represent the data points in the form of a graph (Similarity graph). Similarity graph is a graph with all the data points as vertices and edges between any two vertices are the similarity between those two data points.

   There are different types of similarity graphs. Selection of which depends on certain factors like the nature and structure of data points used. However deciding the kind of similarity graph and even the similarity measure is not a trivial task. Spectral clustering algorithm strongly depends on the kind of similarity function and similarity graph used. These are are different types of Similarity graphs

   - Epsilon neighborhood graph
   - k-nearest neighbor graph
   - The fully connected graph

2. **Graph Laplacian :** The most important matrix in spectral clustering is laplacian graph.There are certain properties which makes it important for the clustering algorithm

3. **Unnormalized Graph Laplacian :**

   Unnormalized graph laplacian is defined as

$$L = D - W \tag{1.5}$$

4. **Properties of unnormalised graph laplacian :**

   - $$f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2;\ \forall f \in \mathbb{R}^n \tag{1.6}$$
   - L is symmetric and positive semi definite.
   - The smallest eigen vector is 0 and corresponding eigen vector is constant vector.
   - L has n non negative real valued eigen values.

   The most important property that comes from graph laplacian is :

   Multiplicity of 0 is equal to the number of connected components in the graph. That is we can divide the data points into a number of clusters which is equal to the Multiplicity of 0

5. **Normalised graph laplacian :**

   $$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \tag{1.7}$$

   $$L_{rw} = D^{-1} L \tag{1.8}$$

6. **Unnormalized spectral clustering Algorithm :**

   - Construct symmetric matrix W
   - Compute unnormalised laplacian
   - Compute first K eigen vectors u1,u2,...,uk
   - Construct a matrix U containing k eigen vectors as column
   - use k means clustering to cluster the data points

7. **Normalized spectral clustering Algorithm :**

   - Construct symmetric matrix W
   - Compute unnormalised laplacian
   - Compute first K eigen vectors u1,u2,...,uk
   - Construct a matrix U containing k eigen vectors as column
   - Form a normalized matrix T from U

- Use k means clustering on T to cluster the data points

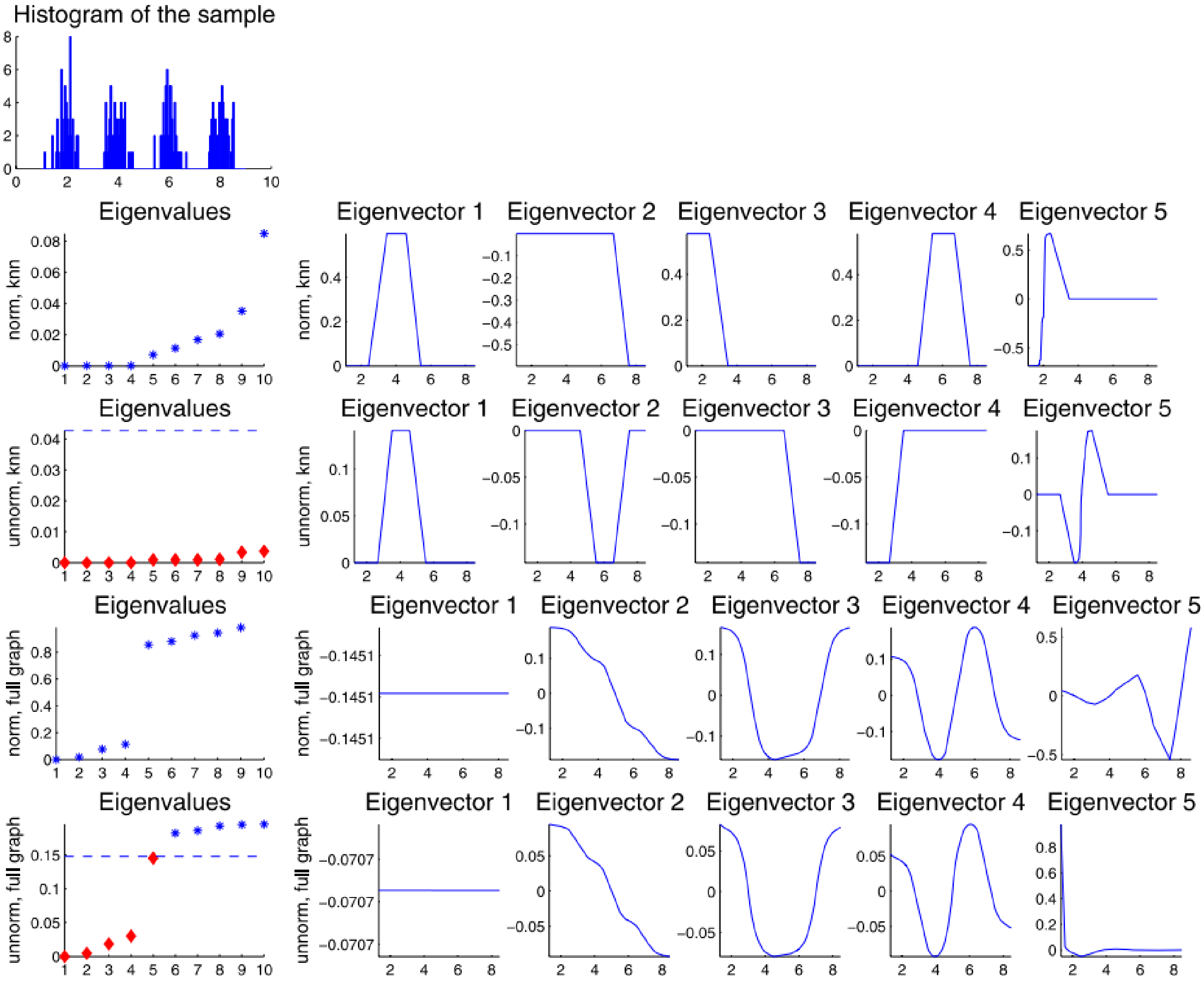


Fig. 1.2 Performance of spectral Clustering

Chapter 2

Fairness in Unsupervised Learning

ML models are trained on large amount of data. These data may contains biases which may be amplified when use in a high-impact applications. As a result several surveys have been done to remove the bias factor from the data and to make ML models fair.
Ensuring fairness in Supervised learning may be less complex than unsupervised learning due to presence of label but fair unsupervised learning algorithms arises a challenging task due to lack of labels. However there are very less research and works to make clustering algorithms fair but the need of making them fair is highly a matter of concern.

2.1 Fairness notions for clustering

Fairness notions are application specific and one notion may perform oppositely when used in different application.

1. **Group-Level Notions**

   According to this notion no group of individuals should be given least and highest priority in terms of predicted output.Group level fairness ensures the fairness of protected groups. There are different examples of group level fairness which we be discussing in the chapter.

This can be understood through a credit card data set example. If a bank is using any data set which contains gender, earning, profession and other sensitive attribute then the result obtain may be not fair to certain group of people. Like if a bank wants to predict the scheme of loan that should be offered to an individual based on their earnings then a group of people (women) who are earning less will not get a chance to be offered a good scheme. This causes group level unfairness and this notion is for taking care of this kind of behaviour.

2. **Individual-Level Notions**

   In this notion we do not deal with protected group but we make sure that similar kind of individual should be treated similarly. Individual level fairness has not yet been explored so deeply so far as group level fairness.There are different kinds of notions that satisfies individual level fairness we will be discussing them in this chapter

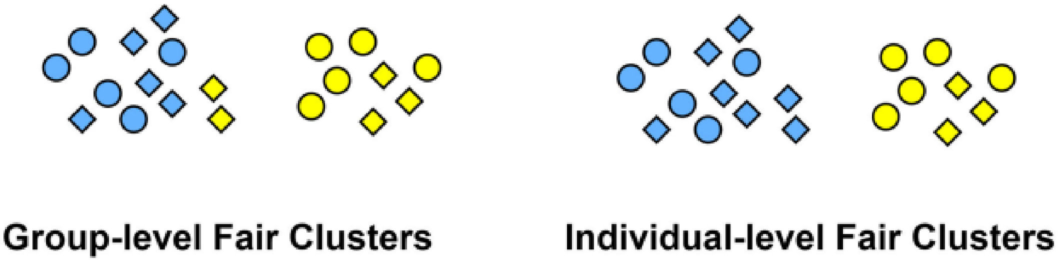


Fig. 2.1 Different kinds of notion may perform differently on same data set

3. **Algorithm Agnostic Notion and Algorithm Specific Notion**

   Algorithmic agnostic notion of fairness includes those fairness notions that can be generalized for all types o clustering algorithms .On the other hand algorithm specific notion is for a particular algorithm as the name suggests.

2.2 Some commonly used notions

In this section we will see examples of different types of notions which are commonly used

1. **Balance**

   This notion is algorithm agnostic and it needs to be maximized. m = Number of protected groups in the data set

   r = Number of samples in the entire data set belonging to group b

   ra = Number of samples in a cluster belonging to group b

   $$Balance = \min_{a \in [k], b \in [m]} min\{R_{ab}, \frac{1}{R_{ab}}\} \quad (2.1)$$

2. **Social Fairness**

   This notion is for K means clustering algorithm.This notion needs to be minimized. Let O denotes the k means clustering cost then social fairness is given by the following equation

   $$Social Fairness = \max_{a \in m} \frac{O(U, X_a)}{|X_a|} \quad (2.2)$$

3. **Bounded Representation**

   We define constraint with two parameters alpha and beta which all the clusters should satisfy in order to be called as fair clustering. We generally take alpha and beta as 1/m where m is the number of protected groups.

   $$\beta <= P_{ab} <= \alpha, \forall a \in [k], b \in [m] \quad (2.3)$$

4. **Max Fairness cost**

   The lower the value of Max Fairness Cost the better the clusters will be in terms of fairness. Here we need an additional parameter Ideal proportion(I=1/m) . It is a group level notion.

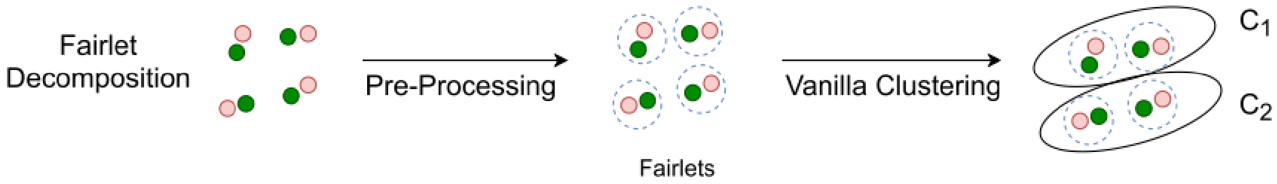
$$MaxFairnessCost = \max_{a \in k} \sum_{b \in [m]} |P - I_{ab}| \tag{2.4}$$

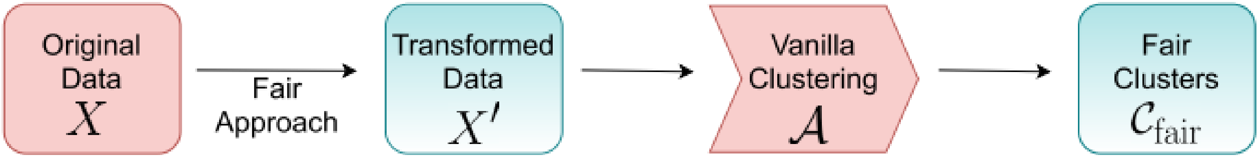
2.3 Approaches of Fair clustering

Three approaches are followed

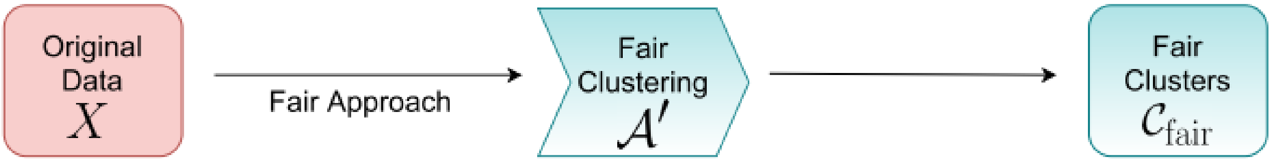
- Pre-processing
- In-processing
- Post-processing

1. **Pre-Processing Approach** Before clustering the data set is passed through a pre-process stage and then vanilla clustering algorithm is applied in it. As a result the fair clusters are formed.

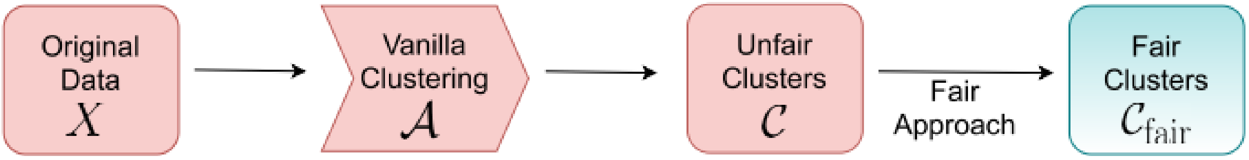



Original Data X
Fair Approach
Transformed Data X'
Vanilla Clustering $\mathcal{A}$
Fair Clusters $\mathcal{C}_{\text{fair}}$

2. **In-Processing Approach** In this approach the clustering algorithm itself is modified such that at the end fair clusters are produced.


Original Data X
Fair Approach
Fair Clustering $\mathcal{A}'$
Fair Clusters $\mathcal{C}_{\text{fair}}$

3. **Post Processing Approach** In this approach vanilla clustering algorithm is applied over the original data set. The clusters formed are then processed to make the clusters fair


Original Data X
Vanilla Clustering $\mathcal{A}$
Unfair Clusters $\mathcal{C}$
Fair Approach
Fair Clusters $\mathcal{C}_{\text{fair}}$

Chapter 3

Fair Spectral clustering

Spectral clustering sometimes shows unsavory behaviour against different demographic group. Several settings have been made to make supervised learning fair. In this chapter we will see how to incorporate fairness in spectral clustering. Inorder to make spectral clustering fair we want make the proportion of each group same as the proportion of these groups in the entire data set

3.1 Unnormalized spectral clustering

- Make similarity graph
- Find laplacian matrix
- Find eigen value and eigen vectors
- Place k smallest eigen vector as the column of a matrix H
- Apply k means algorithm to cluster the data points

3.2 Adding fairness constraints

For clustering to be fair every clusters should hold the following

$$\forall s \in [h-1] : \sum_{i=} n(f_i^{(s)} - \frac{|V_s|}{n})Hil = 0 \implies \forall s \in [h] : \frac{V_s \cap C_l}{|C_l|} = \frac{V_s}{n} \tag{3.1}$$

Here h is the demographic group After adding this into the spectral clustering we can make spectral clusters fair

We may relax the fairness constraints to

$$\min_{HinR^{nxk}} Tr(H^T LH) \tag{3.2}$$

Subject to

$$1.H^T H = I_k \tag{3.3}$$

$$2.F^T H = 0_{(h-1)xk} \tag{3.4}$$

Let Z be the orthonormal nullspace of Transpose of F. We can substitute H=ZY.

Equation 3.2 can be decomposed into

$$\min_{YinR^{(n-h+1)xk}} (Y^T Z^T LZY) \tag{3.5}$$

subject to

$$Y^T Y = I_k \tag{3.6}$$

Solution to problem 3.5 can be given by a matrix Y thats contains orthonormal eigen vectors corresponding to k smallest eigen values of

$$Z^T LH \tag{3.7}$$

3.3 Unnormalized spectral clustering with fairness constraints

- Make similarity graph
- Find laplacian matrix
- Let F be a matrix with columns

$$f_s - \frac{V_s}{n}.1_s, s \in [h-1] \tag{3.8}$$

- Compute Z whose columns form the orthonormal basis of transpose of F

- compute the k smallest eigenvalues of

$$Z_T L Z \tag{3.9}$$

  and the corresponding orthonormal eigenvectors
- apply k-means clustering to the rows of H = ZY

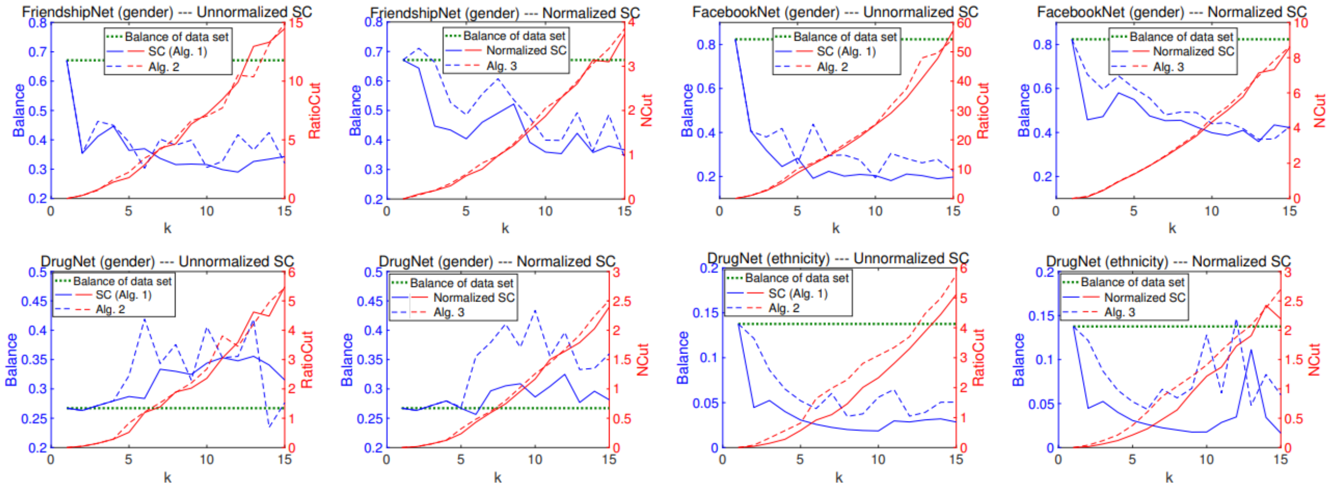


Fig. 3.1 Guarantees for Spectral Clustering with Fairness Constraints

3.4 Computational complexity

Fair spectral clustering has the same time and space complexity as spectral clustering in worst case ie;

Space complexity: O(n^2)

Time Complexity: O(n^3)

3.5 Normalized spectral clustering with fairness constraints

- Make similarity graph
- Find laplacian matrix
- Let F be a matrix with columns

$$f_s - \frac{V_s}{n}.1_s, s \in [h-1] \tag{3.10}$$

- Compute Z whose columns form the orthonormal basis of transpose of F

- Compute square root Q of
$$Z_T LZ \tag{3.11}$$
- Compute orthonormal eigen vector correspoinding to k smallest eigen values of
$$Q^{-1}Z^T LZQ^{-1} \tag{3.12}$$
- X is the matrix containing these eigen vectors
- Apply k-means clustering to the rows of
$$H = ZQ^{-1}X \tag{3.13}$$

3.6 Computational complexity

Fair spectral clustering has the same time and space complexity as spectral clustering in worst case ie; **Space complexity**: O(n^2)

Time Complexity: O(n^3)

Chapter 4

Constrained spectral clustering

Introduction to domain knowledge may greatly improve the performance of any clustering algorithm. Domain knowledge can be given in the form of class label (like supervised learning) or in the form of constraints. However giving constraints is generally followed. Performance of any clustering algorithms strongly depends on how these constraints are imposed, there are two ways

1. Imposing constraints in the affinity matrix

2. Integration of constraints with the optimization criteria

There are different methods to reduce the complexity of tasks like **PCA** and **LPP**. They aim to reduce project the data points in a space where they are linearly separable.

4.1 Constrained Principal component analysis

Let us define M and C as the set of must-link and cannot-link in the data set Then the criteria is defined by

$$J_{cPCA} = J_{PCA} + \frac{1}{|C|} \sum_{(x_i,x_j)\in C} (a^T x_i - a^T x_j)^2 - \frac{1}{|M|} \sum_{x_i,x_j)\in M} (a^T x_i - a^T x_j)^2 \tag{4.1}$$

This equation leads to

$$J_{cPCA} = \sum_{i,j} (a^T x_i w_{ij} x_i^T a - a^T x_i w_{ij} x_j^T a) \tag{4.2}$$

$$J_{cPCA} = a^T X(D - W) X^T a \tag{4.3}$$

The solution for this is obtained from top eigen vectors of

$$XLX^T a = \lambda a \tag{4.4}$$

4.2 Constrained locality preserving projection

It constructs an affinity matrix as

$$W_{ij} = e^{\frac{-|x_i - x_j|^2}{2\sigma^2}}, if x_i isknearestneighborsof x_j \tag{4.5}$$

$$= 0, otherwise \tag{4.6}$$

The objective function can be given as

$$J_{cLPP} = J_{LPP} + \frac{1}{|C|} \sum_{(x_i, x_j) \in C} (a^T x_i - a^T x_j)^2 - \frac{1}{|M|} \sum_{x_i, x_j) \in M} (a^T x_i - a^T x_j)^2 \tag{4.7}$$

It can be rewritten using Laplacian matrix as

$$J_{cLPP} = a^T XLX^T a (s.t. a^T a = 1) \tag{4.8}$$

4.3 Constrained spectral clustering approaches

1. Integration of constraints in affinity matrix They incorporate pairwise constraints in the affinity matrix as

$$\tilde{w}_{ij} = \begin{cases} 0 & \text{if } (x_i, x_j) \in \mathcal{C}, \\ +1 & \text{if } (x_i, x_j) \in \mathcal{M}, \\ w_{ij} & \text{otherwise.} \end{cases}$$

2. Integration of constraints as optimization criteria Beside laplacian term it explicitly encodes some conatraints in the optimization problem. The constraint matrix is given by

$$q_{ij} = \begin{cases} -1 & \text{if } (x_i, x_j) \in \mathcal{C}, \\ +1 & \text{if } (x_i, x_j) \in \mathcal{M}, \\ 0 & \text{else.} \end{cases}$$

The constained spectral clustering is obtained from the eigen vectors of

$$L_{cSC} = \gamma L + (1 - \gamma) L_Q \tag{4.9}$$

Chapter 5

Constrained Flexible spectral Clustering

After getting to know about the incorporation of constraints in spectral clustering, we needed a way to incorporate these constraints effectively and also needed to solve these objective function in polynomial amount of time. In this chapter we will be looking at how to incorporate constraints in the spectral clustering in an efficient manner such that it can be solved deterministically in polynomial time. We are trying to incorporate user supervision into spectral clustering with a real valued degree of belief as a result of which we end up having a multi objective optimisation

5.1 Objective function

$$Q_{ij} = Q_{ji} = \begin{cases} +1 & \text{if ML}(i,j) \\ -1 & \text{if CL}(i,j) \\ 0 & \text{no supervision available} \end{cases} .$$

The larger the value of

$$u^T Q u = \sum_{i=1} N \sum_{j=1} N u_i u_j Q_{ij} \tag{5.1}$$

the better the clustering algorithm satisfies the constraint imposed.

$$u^T Q u >= \alpha \tag{5.2}$$

lets take

$$u = D^{-\frac{1}{2}} v \tag{5.3}$$

$$v^T Q * v >= \alpha \tag{5.4}$$

where

$$Q* = D^{-\frac{1}{2}} Q D^{-\frac{1}{2}} \tag{5.5}$$

is normalized constraint matrix. We want to optimize this which is a multi objective

$$\arg\min_{\mathbf{v}\in\mathbb{R}^N} \mathbf{v}^T \bar{L} \mathbf{v}, \;\; \text{s.t. } \mathbf{v}^T \bar{Q} \mathbf{v} \geq \alpha, \;\; \mathbf{v}^T \mathbf{v} = \mathrm{vol}(\mathcal{G}), \;\; \mathbf{v} \neq D^{1/2} \mathbf{1}.$$

optimisation.

5.2 Problem Statement

After incorporating constraints into fair spectral clustering, our aim is to find the solution of the following multi-objective function deterministically in polynomial time.

$$\arg\min_{\mathbf{v}\in\mathbb{R}^N} \mathbf{v}^T \bar{L} \mathbf{v}, \;\; \text{s.t. } \mathbf{v}^T \bar{Q} \mathbf{v} \geq \alpha, \;\; \mathbf{v}^T \mathbf{v} = \mathrm{vol}(\mathcal{G}), \;\; \mathbf{v} \neq D^{1/2} \mathbf{1}.$$

5.3 Future work and conclusion

We are looking forward to solve the multi optimization function in order to find fair clusters. Many of the constrained spectral clustering algorithms are present which focus on MUST-LINK and CANNOT-LINK constraints which is very inefficient and infelxible at the same time. We will be dealing with both binary constraints and real-valued degree of belief constraints which is flexible. Our approach will be an extension of original objective function and can be solved in polynomial time complexity.