Research case study: Bipartite graph partitioning and data clustering

Vipulanandan Pragatheeswaran

***Abstract*— Data types which are from data wrangling can converted into graphs. A special type of graphs is bipartite graphs. Typical vertex clustering cannot be applied as it is for bipartite graphs. Hence in this case study we will be focusing on data clustering for a bipartite graph. We will computing a partial singular value decomposition (SVD) of the given graph. We will be applying the algorithm for our own dataset and obtaining the clustering results.**

***Keywords*—bipartite graph, clustering, singular value decomposition**

I. Introduction

In exploratory data mining the clustering analysis is used in many applications, especially when we are initially starting out some data analysis or when we don't have much of insights or supervised information about the data or the domain itself. When it comes to implementation of cluster analysis – the methodologies are varying and enormous, many classical [1,3,4] approaches tweaked to fit for the domain specificities. The generic approach for clustering, the algorithms use the underlying assumption that the given dataset consists of covariate attributes for each individual data object, and clustering analysis will group the set of vectors each represent the dataset.

In this case study we will be focusing on a new principal which is proposed in [1], which is used to cluster some bipartite dataset. These datasets were generated using a set of 1400 tweets. The edges in each bipartite represents whether the tweet contains the given TRP or EMP.

In this case study we will be using two datasets, namely EMP and TRP. The dataset was gathered to analyze the ecology of online conspiratorial rhetoric associated with one particular hashtag "white genocide" and the attributes these cultural forms contain.

Within our bipartite graph model, the clustering problem can be solved by constructing vertex graph partitions. Many criteria have been proposed for measuring the quality of graph partitions of undirected graphs [5,6]. In this review, we show how to adapt those criteria for bipartite graph partitioning and therefore solve the bi-clustering problem.

A great variety of objective functions have been proposed for cluster analysis without efficient algorithms for finding the (approximate) optimal solutions. We will show that our bipartite graph formulation naturally leads to partial SVD problems for the underlying edge weight matrix which admit efficient global optimal solutions.

The rest of the paper is organized as follows: We will be discussing the methodology and proving the min cut problem and decomposing it into the singular value decomposition (SVD) problems.

II. Methodology

A. Bipartite graph partitioning

Let's denote a graph by $G(V,E)$, where V is the vertex set and E is the edge set of the graph. A graph $G(V,E)$ is bipartite with two vertex classes X and Y if $V = X \cup Y$ with $X \cap Y = \emptyset$ and each edge in E has one endpoint in X and one endpoint in Y.

Though in the graphs I have tested the methodology of [1] is unweighted, (weights=1), in the methodology section the focus will be to generalize the approach to a weight matrix W.

Hence, consider weighted bipartite graph $G(X,Y,W)$ with $W = \left(w_{ij}\right)$ where $w_{ij} > 0$ denotes the weight of the edge between vertex i and j. We let $w_{ij} = 0$ if there is no edge between vertices i and j.

A vertex partition of $G(X,Y,W)$ denoted by $\Pi(A,B)$ is defined by a partition of the vertex sets X and Y, respectively: $X = A \cup A^c$, and $Y = B \cup B^c$, where for a set S, S^c denotes its compliment. By convention, we pair A with B, and A^c with B^c. We say that a pair of vertices $x \in X$ and $y \in Y$ is a match with respect to a partition $\Pi(A,B)$ if there is an edge between x and y, and either $x \in A$ and $y \in B$ or $x \in A^c$ and $y \in B^c$.

For any two subsets of vertices $S \subset X$ and $T \subset Y$, define

$$W(S,T) = \sum_{i\in S, j\in T} w_{ij}$$

$W(S,T)$ is the sum of the weights of edges with one endpoint in S and one endpoint in T. The quantity $W(S,T)$ can be considered as measuring the association between the vertex sets S and T. In the context of cluster analysis edge weight measures the similarity between data objects. To partition data objects into clusters, we seek a partition of $G(X,Y,W)$ such that the association (similarity) between unmatched vertices is as small as possible. One possibility is to consider for a partition $\Pi(A,B)$ the following quantity

$$\begin{aligned}\text{cut }(A,B) &\equiv W(A,B^c) + W(A^c,B) \\ &= \sum_{i\in A, j\in B^c} w_{ij} + \sum_{i\in A^c, j\in B} w_{ij}\end{aligned}$$

Intuitively, choosing $\Pi(A,B)$ to minimize cut (A,B) will give rise to a partition that minimizes the sum of all the edge weights between unmatched vertices. Unfortunately, choosing a partition based entirely on cut (A,B) tends to produce unbalanced clusters, i.e., the sizes of A and/or B or their compliments tend to be small. In [1], the authors propose the following normalized variant of the edge cut

$$\text{Ncut }(A,B) \equiv \frac{\text{cut }(A,B)}{W(A,Y)+W(X,B)} + \frac{\text{cut }(A^c,B^c)}{W(A^c,Y)+W(X,B^c)}.$$

The intuition behind this criterion is that not only we want a partition with small edge cut, but we also want the two subgraphs formed between the matched vertices to be as dense as possible.

B. Approximate solutions using singular vectors

Given a bipartite graph G(X, Y, W) and associated partition ∏(A, B). Let X and Y as vertices in A and B.
The weight matrix can be written in a block format,

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22}\end{bmatrix}$$

The vertices in A and B are ordered before vertices in A^c and B^c, respectively. i.e the rows of W_{11} correspond to the vertices in the vertex set A and the columns of W_{11} correspond to those in B.

Additionally we will be defining the sum of all elements in a matrix of m-by-n say $H = (h_{ij})$, define

$$\text{s}(H) = \sum_{i=1}^{m}\sum_{j=1}^{n} h_{ij}$$

Using the above, the Ncut between both partitions will be defined as,

$$\text{Ncut }(A,B) = \frac{\text{s}(W_{12})+\text{s}(W_{21})}{2\text{ s}(W_{11})+\text{s}(W_{12})+\text{s}(W_{21})} + \frac{\text{s}(W_{12})+\text{s}(W_{21})}{2\text{ s}(W_{22})+\text{s}(W_{12})+\text{s}(W_{21})}$$

we first consider the case when W is symmetric. For cases when W is symmetric,

$$\text{NCut(A)} = \text{Ncut}(A,A)$$

$$= \frac{\text{s}(W_{12})}{\text{s}(W_{11})+\text{s}(W_{12})} + \frac{\text{s}(W_{12})}{\text{s}(W_{22})+\text{s}(W_{12})}$$

Let e be the vector with all its elements equal to 1. Let D be the diagonal matrix. Let $x = (x_i)$ be the vector with

$$x_i = \begin{cases} 1, & i\in A, \\ -1, & i\in A^c.\end{cases}$$

Define p,

$$p \equiv \frac{\text{s}(W_{11})+\text{s}(W_{12})}{\text{s}(W_{11})+2\text{ s}(W_{12})+\text{s}(W_{22})} = \frac{\text{s}(W_{11})+\text{s}(W_{12})}{e^T De}$$

Then

$$\begin{aligned}\text{s}(W_{11})+\text{s}(W_{12}) &= pe^T De \\ \text{s}(W_{22})+\text{s}(W_{12}) &= (1-p)e^T De\end{aligned}$$

and

$$\text{Ncut }(A) = \frac{x^T(D-W)x}{4p(1-p)e^T De}.$$

Notice that $(D-W)e = 0$, then for any scalar s, we have

$$(se+x)^T(D-W)(se+x) = x^T(D-W)x.$$

From the Rayleigh quotient, we need to find s such that

$$(se+x)^T D(se+x) = 4p(1-p)e^T De.$$

Since $x^T Dx = e^T De$, it follows from the above equation that $s = 1-2p$. Now let $y = (1-2p)e + x$, it is easy to see that $y^T De = ((1-2p)e+x)^T De = 0$, and

$$y_i = \begin{cases} 2(1-p) > 0, & i\in A, \\ -2p < 0, & i\in A^c.\end{cases}$$

Thus

$$\min_A \text{Ncut }(A) = \text{m } \left\{\frac{y^T(D-W)y}{y^T Dy} \mid y\in S\right\},$$

where

$$S = \{y \mid y^T De = 0, y_i \in \{2(1-p), -2p\}\}.$$

If we drop the constraints $y_i \in \{2(1-p), -2p\}$ and let the elements of y take arbitrary continuous values, then the optimal y can be approximated by the following relaxed continuous minimization problem,

$$\mathrm{m}\ \left\{\frac{y^T(D-W)y}{y^TDy} \mid y^TDe = 0\right\}.$$

This follows from $We = De$,

$$D^{-1/2}WD^{-1/2}\left(D^{1/2}e\right) = D^{-1/2}e$$

Where $D^{1/2}e$ is an eigenvector of $D^{-1/2}WD^{-1/2}$ corresponding to the eigenvalue 1.
Thus the optimal y can be computed as $y = D^{1/2}\hat{y}$, where $\hat{y}$ is the second largest eigenvector of $D^{-1/2}WD^{-1/2}$.

Now we return to the rectangular case for the weight matrix W, and let D_X and D_Y be diagonal matrices such that

$$We = D_Xe,\ W^Te = D_Ye.$$

Consider a partition $\Pi(A,B)$, and define

$$u_i = \begin{cases} 1, & i \in A \\ -1, & i \in A^c \end{cases},\ v_i = \begin{cases} 1, & i \in B \\ -1, & i \in B^c \end{cases}$$

Let W have the block form as in (2), and consider the augmented symmetric matrix [4]

$$\hat{W} = \begin{bmatrix} 0 & W \\ W^T & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & W_{11} & W_{12} \\ 0 & 0 & W_{21} & W_{22} \\ W_{11}^T & W_{21}^T & 0 & 0 \\ W_{12}^T & W_{22}^T & 0 & 0 \end{bmatrix}$$

If we interchange the second and third block rows and columns of the above matrix, we obtain

$$\begin{bmatrix} 0 & W_{11} & 0 & W_{12} \\ W_{11}^T & 0 & W_{21}^T & 0 \\ 0 & W_{21} & 0 & W_{22} \\ W_{12}^T & 0 & W_{22}^T & 0 \end{bmatrix} \equiv \begin{bmatrix} \hat{W}_{11} & \hat{W}_{12} \\ \hat{W}_{12}^T & \hat{W}_{22} \end{bmatrix}$$

and the normalized cut can be written as

$$\mathrm{Ncut}\ (A,B) = \frac{\mathrm{s}(\hat{W}_{12})}{\mathrm{s}(\hat{W}_{11}) + \mathrm{s}(\hat{W}_{12})} + \frac{\mathrm{s}(\hat{W}_{12})}{\mathrm{s}(\hat{W}_{22}) + \mathrm{s}(\hat{W}_{12})},$$

a form that resembles the symmetric case.
Define

$$q = \frac{2\ \mathrm{s}(W_{11}) + \mathrm{s}(W_{12}) + \mathrm{s}(W_{21})}{e^TD_Xe + e^TD_Ye}.$$

Then we have

$$\begin{aligned} \mathrm{Ncut}\ (A,B) &= \frac{-2x^TWy + x^TD_Xx + y^TD_Yy}{x^TD_Xx + y^TD_Yy} \\ &= 1 - \frac{2x^TWy}{x^TD_Xx + y^TD_Yy}, \end{aligned}$$

where $x = (1-2p)e + u, y = (1-2p)e + v$. It is also easy to see that

$$x^TD_Xe + y^TD_Ye = 0,\ x_i, y_i \in \{2(1-q), -2q\}$$

Therefore,

$$\begin{aligned} &\min_{\Pi(A,B)} \mathrm{Ncut}\ (A,B) \\ &= 1 - \max_{x\neq 0, y\neq 0} \left\{\frac{2x^TWy}{x^TD_Xx + y^TD_Yy} \mid x, y \text{ satisfy (7)}\right\}. \end{aligned}$$

In [2], the Laplacian of $\hat{W}$ is used for partitioning a rectangular matrix in the context of designing load-balanced matrix-vector multiplication algorithms for parallel computation. However, the eigenvalue problem of the Laplacian of $\hat{W}$ does not lead to a simpler singular value problem. Ignoring the discrete constraints on the elements of x and y, we have the following continuous maximization problem,

$$\max_{x\neq 0, y\neq 0} \left\{\frac{2x^TWy}{x^TD_Xx + y^TD_Yy} \mid x^TD_Xe + y^TD_Ye = 0\right\}$$

Without the constraints $x^TD_Xe + y^TD_Ye = 0$, the above problem is equivalent to computing the largest singular triplet of $D_X^{-1/2}WD_Y^{-1/2}$ [1]. From this the authors in [1], simply the normalization to following,

$$D_X^{-1/2}WD_Y^{-1/2}\left(D_Y^{1/2}e\right) = D_X^{1/2}e$$
$$\left(D_X^{-1/2}WD_Y^{-1/2}\right)^T\left(D_X^{1/2}e\right) = D_Y^{1/2}e$$

Since singular values of $D_X^{-1/2}WD_Y^{-1/2}$ are at most 1. Therefore, an optimal pair $\{x, y\}$ can be computed as $x = D_X^{-1/2}\hat{x}$ and $y = D_Y^{-1/2}\hat{y}$, where $\hat{x}$ and $\hat{y}$ are the second largest left and right singular vectors of $D_X^{-1/2}WD_Y^{-1/2}$

III. Algorithm

Spectral Recursive Embedding (SRE) Given a weighted bipartite graph $G = (X, Y, E)$ with its edge weight matrix W

1 Compute D_X and D_Y and form the scaled weight matrix $\hat{W} = D_X^{-1/2}WD_Y^{-1/2}$.

2 Compute the second largest left and right singular vectors of $\hat{W}$, $\hat{x}$ and $\hat{y}$.

3 Find cut points c_x and c_y for $x = D_X^{-1/2}\hat{x}$ and $y = D_Y^{-1/2}\hat{y}$, respectively.

4 Form partitions $A = \{i \mid x_i \geq c_x\}$ and $A^c = \{i \mid x_i < c_x\}$ for vertex set X, and $B = \{j \mid y_j \geq c_y\}$ and $B^c = \{j \mid y_j < c_y\}$ for vertex set Y.

5 Recursively partition the sub-graphs $G(A, B)$ and $G(A^c, B^c)$ if necessary.

IV. RESULTS

The results section is segmented into two, namely the analysis for EMP edge dataset and TRP edge dataset.
The following contains two sets of tests which were carried out,

- Start with a random partition and then apply the discussed algorithm
- Apply the discussed algorithm directly

In both cases it was observed that neither way the clustering results did not vary.

One interesting observation which was observed is that during each recursive partition only 1 or 2 nodes from the user-id was dropped via the min cut. Which leads to the conclusion that majority of the data set is not partitionable/ very minimal partition between them. Yet another school of thought is that the threshold determines how quickly one may achieve ideal partition, like a learning rate in neural networks.

Also during the partition, it was observed that only the user-ids were partitioned erratically but not the EMPs or TRP classes.

A. EMP edge dataset

The following results were obtained for the EMP edge dataset.

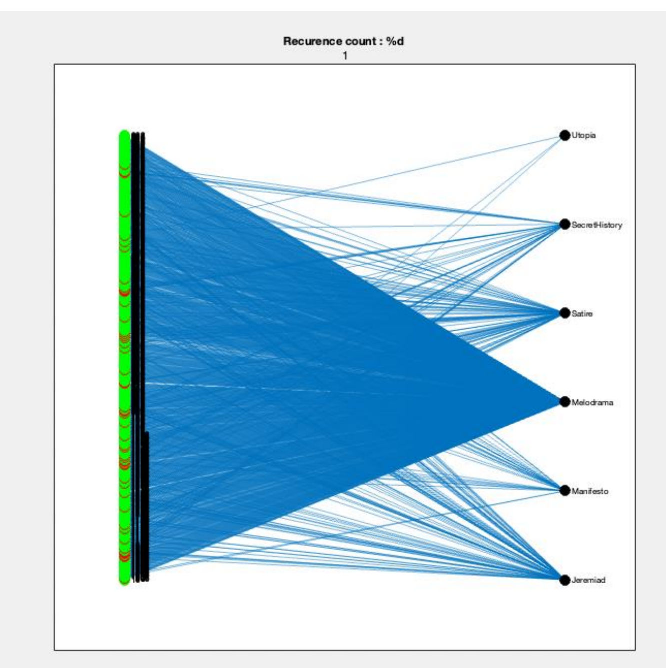

Figure 1: Random partition split in both set A and B

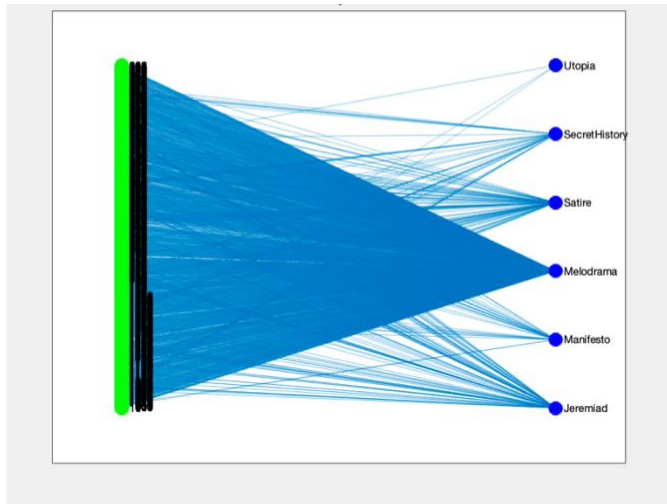

Figure 2: Set partition after recursion of 2 iteration

From figure 2, it is visible that the partitions don't technically partition but rather group all of them together.

B. TRP edge dataset

The following results were obtained for the TRP edge dataset.

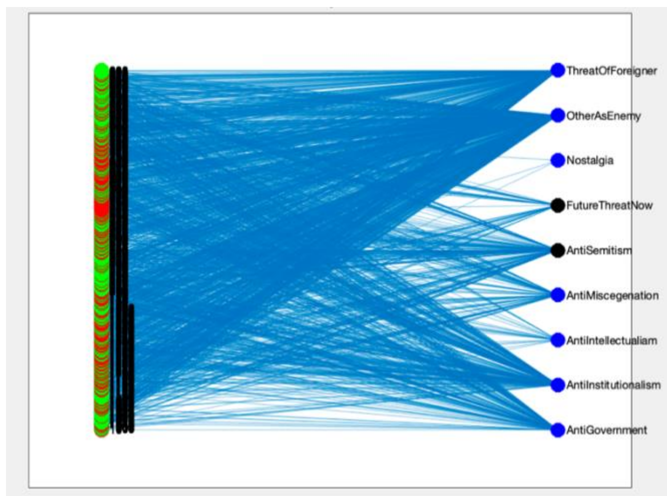

Figure 3:Random partition split in both set A and B

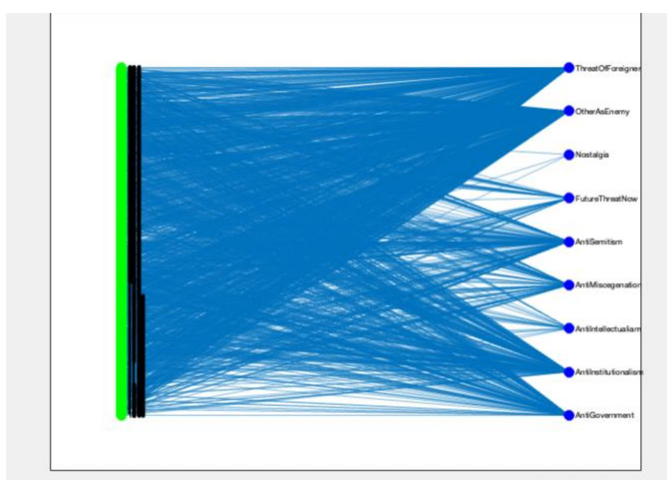

Figure 4: Clustering results from bipartites

From figure 4, it is visible that the partitions don't technically partition but rather group all of them together.

V. CONCLUSION AND FUTURE WORK

Though significant partition was not observable in either case, one key factor to consider is that the min-cut threshold c_x and c_y were maintained as zero. Varying the threshold to get better partition would be an approach which should be applied. Additionally varying thresholds for each recursion is also a possible approach for better partitions of bipartite.

REFERENCES

[1] Zha, Hongyuan & He, Xiaofeng & Ding, Chris & Simon, Horst & Gu, Ming. (2001). Bipartite Graph Partitioning and Data Clustering. International Conference on Information and Knowledge Management, Proceedings. 10.2172/816202..

[2] B. Hendrickson and T.G. Kolda. Partitioning sparse rectangular and structurally nonsymmetric matrices for parallel computation. SIAM J. Scientific Computing, 21:2048-2072, 2000

[3] B. Everitt. Cluster Analysis. Edward Arnold, 1993.

[4] A. D. Gordon. Classification. Second Edition, Chapman and Hall, 2000.

[5] F.R.K. Chung. Spectral Graph Theory. American Mathematical Society, 1997.

[6] J. Shi and J. Malik. Normalized cuts and image segmentation. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, June 1997.