Small Network Segmentation with Template Guidance

Kristin Diane Lui
Department of Mathematics
University of California, Davis
Davis, CA 95616
kdlui@math.ucdavis.edu

Ian Davidson
Department of Computer Science
University of California, Davis
Davis, CA 95616
davidson@cs.ucdavis.edu

ABSTRACT
The importance of segmenting social networks has grown along with the usage of social network sites such as Facebook. Social networks provide a wealth of information about people including friendships and interests. Information can be added to segmenting algorithms such as spectral clustering as in the form of a single constraint. Taking advantage of multiple instances of information, such as a template composition, for clustering is not trivial since adding multiple constraints to spectral clustering techniques is difficult. We present an approach to incorporate template constraints to the semi-definite program formulation of spectral clustering. We describe different types of template constraints and their interpretation. Also, we present a randomize rounding scheme to ensure the rounded solution to the SDP formulation satisfies the template segmentation.

Keywords
Spectral Clustering, Semidefinite Programming, Randomized Rounding, Constrained Clustering

1. INTRODUCTION
Social networks have grown in popularity in the last decade where websites such as Facebook, Myspace, and Twitter have over 100 million users each. With growing use, it is important to have useful segmentations of these networks. Segmentations based just on graph topology (edge structure) have their uses but in this paper we shall look at segmentations of ego networks with template constraints that allow domain experts to encode segment expectations as a template structure. This work is well suited to social networks so as to allow users to describe themselves along lines of interests, location, and work affiliation, for example. Extending segmentation of social networks to factor in these descriptions allows for a variety of directions including creating segments of people who are friends and also share (or do not share) similar interests.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MLG '11 San Diego, CA, USA
Copyright 2011 ACM 978-1-4503-0834-2 ...$10.00

A key problem is then how to represent this additional node information, or background information, in a meaningful way so that existing clustering algorithms can take advantage of it. Spectral clustering is an important technique used to find segmentations of a single graph [19][5][23]. The quadratic programming formulation (which spectral clustering is an example of) does not lend itself to easily adding more constraints on this extra information [26]. In this earlier work, we could only add **one preference** for the segmentation. Semi-definite programming (SDP) is an alternative convex programming formulation [27][3] (whose objective function is linear, not quadratic) that easily allows the addition of many constraints.

Using the SDP formulation, we can segment small networks with templates. Templates are naturally modeled as positive semi-definite matrices (kernels) that can be added as linear constraints in the SDP formulation. As a constraint, we can dictate whether the segmentation will have strict or loose adherence to the template composition. For example, we can conduct pragmatic operations such as, "Segment a network so we have in the same clustering: **all** my close friends who have similar music tastes" for strict adherence to the constraint or, "Segment a network so we have in the same clustering: **some** of my close friends who have similar music tastes" for loose adherence to the constraint. More complex operations could involve segmenting a network so that people in the same cluster have similar music tastes but did **not** go to school together. *In this way, we are placing a compositional restriction on who can be in the same cluster together.*

Consider the example shown in Figure (1). The left graph with black nodes shows the segmentation (indicated by the dotted line) from normalized spectral clustering [19]. Spectral clustering is discussed in detail in Section 3.1. The right graph shows the clustering result from template clustering. The red square nodes have different labels from the black circle nodes. The template clustering formulation yields a segmentation which considers both the normalized clustering segmentation and the guidance from the template constraints.

Templates can take any number of forms which include:

1. Topological Structures - preserve $x\%$ of a group affiliation such as friendships or co-workers.
2. Characteristics of People - preserve $x\%$ of people who

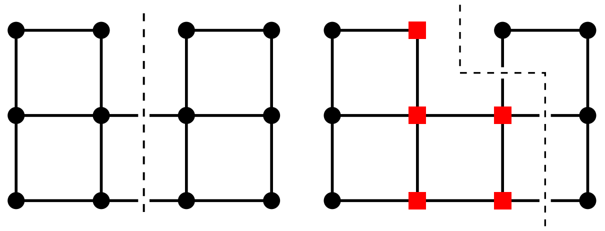

Figure 1: Example of normalized spectral clustering (left) and spectral clustering with a template constraint (right). Normalized spectral clustering seeks a balanced cut of the graph. The template constraint seeks a clustering that also includes known nodes with the same label (shown as red squares).

share a certain characteristic such as people who enjoy rock music.

3. Alternative Segmentations - preserve $x\%$ of a known segmentation of the graph [25][7].

4. Multiple Distance Metrics or Views - Segment a network based on multiple metrics and views

An important contribution of our work is the ability to add multiple templates to the one clustering. Examples of the different types of template compositions and their constraint forms are outlined in Section 4.

We base our work on the SDP formulation of normalized spectral clustering similar to Xing and Jordan [27] to segment a social network with templates based on background knowledge. Our SDP formulation is shown in Section 4. The SDP is easily implementable in MATLAB by using the CVX package developed by Grant and Boyd [11] for convex optimization. The SDP formulation guarantees an optimal solution to the relaxed objective function and template composition constraints can be added to the optimization problem as additional constraints. Another advantage of the SDP formulation is that we are not limited in the number of constraints that are used so multiple sources of background knowledge can be used for segmentation resulting in a desired template for clustering.

However, using templates in an SDP formulation has a significant challenge. The template constraints can be written so that a certain quantity of the structure must be preserved, such as 80% of people within each segment must be from the same school. While the solution from the SDP formulation will satisfy this constraint, due to the relaxed formulation of the membership assignments (which allows fractional values) the solution obtained with rounding up and down membership assignments to integers may not satisfy the template. We present a randomized rounding scheme in Section 5 to ensure the template constraints are satisfied. For example, let there be a network with four people: $C = \{w, x, y, z\}$ and a desired template composition is to have nodes w, x, y together. A possible segmentation is $c = [0.7, 0.6, 0.08, -0.3]$ but rounding the vector c may result in $\hat{c} = [1, 1, -1, -1]$ and the template constraint is no longer satisfied.

Our contributions are:

- We present a novel approach using a SDP formulation for normalized spectral clustering to segment a social network with template compositions.

- We present linear constraints that represent desired template compositions for segmentation. We can place limits on how strictly or loosely the templates are adhered to.

- We present a randomized rounding scheme to ensure the rounded partition solution still satisfies the template segmentation.

Our experimental results show the usefulness of our work. It should be noted that a well known limitation of SDP formulations is handling large problems. In future work we shall extend our work by considering low rank SDP solvers [16][3].

2. PREVIOUS WORK

Previous work that is closely related to our own includes modifying graph Laplacians with guidance from a priori knowledge by Tolliver et al. [22] and adding must link and cannot link linear constraints to an SDP for graph segmentation by Heiler et al. [12].

Tolliver et al. performed image segmentation by modifying the image's normalized laplacian matrix by incorporating external information of the image in order to perform the segmentation. A priori knowledge is generally pixel-level information of the image such as the desired structure of the segmentation. The Fielder space is iteratively modified by the vector representing the a priori knowledge until the vectors are aligned. While we also utilize a priori knowledge in the forms of labels and alternative cuts of the graph, our work differs in that we do not modify the graph Laplacian.

Heiler et al. also incorporates a priori knowledge, specifically about class membership information, by adding linear constraints to the SDP formulation of spectral clustering. The linear constraints for nodes i and j are of the form: $\text{trace}\big((e_i e_j^T + e_j e_i^T)X\big) = 2$, where $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ is the elementary vector with 1 in the i^{th} position. The constraint encodes a must-link constraint between nodes i and j and can be changed to a cannot-link constraint by setting the constraint to -2 instead. Moreover, multiple constraints can be combined as a single constraint: $\text{trace}\left(\sum_{(i,j)\in P_k} (e_i e_j^T + e_j e_i^T)X\right) = 2|P_k|$ where P_k is the set of nodes that are in the same class. While the constraints we propose are similar in nature, the formulation of the constraints allows for a more flexible interpretation of the types of memberships (known structures, class memberships, and alternative segmentations). Additionally, there is more flexiblity to the adherence to the a priori information. For example, nodes i and j that are known to have the same class membership are not necessarily required to be in the same segmentation. The Heiler et al. constraint forces the cut to put nodes i and j in the same or different clusters.

Another method that seeks to incorporate a priori knowledge is segmentations with multiple graphs. A straightforward approach to clustering with multiple views or representations of a graph is to linearly combine kernels [13][31] or convexly combine graph Laplacians [20][1] and apply existing clustering techniques. Tang et al. proposed Linked Matrix Factorization (LMF) for clustering with multiple views [21]. LMF seeks a common factor matrix P that is the low dimensional embedding of entities characterized by the multiple graphs and matrix Λ that captures the characteristics of each graph. None of the methods are able to utilize the extra information provided by the multiple representations. Instead, they seek to average the multiple views so standard clustering techniques can be used. For template segmentation, we want to use the extra information to find nodes that satisfy a template. Thus, previous work in segmentation with multiple views, or kernels, do not lend themselves to template segmentation.

3. SEGMENTING GRAPHS

3.1 Spectral Formulation

Let $G(V, E)$ be a weighted undirected graph with node set $V(G)$ and edge set $E(G)$. Each edge has an edge weight a_{ij} such that $a_{ij} > 0$ if there is an edge between nodes i and j and $a_{ij} = 0$ if there is no edge between nodes i and j. The affinity matrix A is defined as the non-negative symmetric matrix $A = \{a_{ij}\}$. Let the degree of a node i be the sum of the edge weights of the node: $d_i = \sum_j a_{ij}$. Then let the degree matrix D be a diagonal matrix with the node degrees on the diagonal.

The graph partition problem seeks to partition graph G into k disjoint sets $(S_1, S_2, \ldots, S_k)$ by minimizing the total sum of the edge weights removed to create the partitions. This total sum is also known as the cut of the graph which for two sets is defined as: $cut(S_1, S_2) = \sum_{i \in S_1, j \in S_2} a_{ij}$. The drawback of the graph partition problem is that found partitions tend to favor separating a single node from the graph.

To prevent cuts of the graph which are unfavorable, Shi and Malik developed the normalized cut, or Ncut [19] which is defined as

$$Ncut(S_1, \ldots, S_k) = \sum_{i=1}^{k} \frac{cut(S_i, \bar{S}_i)}{vol(S_i)}, \tag{1}$$

where $vol(S)$ is the sum of the edge weights in set S. The definition aims to find balanced clusters based on the edge weights in the cluster. Solving this normalized partition problem is known to be NP hard. However, the problem can be relaxed to a normalized spectral clustering problem. Let the Laplacian matrix be defined as $L = D - A$ and the normalized Laplacian as $\bar{L} = D^{-1/2} L D^{-1/2}$. Now equation (1) can be solved by solving the relaxed optimization problem:

$$\begin{aligned} \underset{z}{argmin} \quad & \frac{z^T \bar{L} z}{z^T z} \\ \text{such that} \quad & z^T z = 1 \end{aligned}$$

which is known as normalized spectral clustering. This problem is easily solved by finding the k top eigenvectors of $\bar{L}$. These k eigenvectors are then used to find the k clusters of the graph.

It is difficult to add constraints to the spectral clustering formulation. To add constraints, some methods alter the affinity matrix or the graph Laplacian [14][24][28] such as the method by Lu et al. [17] which incorporates pair-wise constraints into the affinity matrix. Other methods use the constraints to restrict the feasible solution space [6][8][29][30]. Wang and Davidson provide a formulation for flexible constrained spectral clustering in which a constraint is added to the objective function for spectral clustering to create a novel constrained optimization problem. However, the formulation is only able to incorporate a single set of labels as a single constraint.

3.2 SDP Formulation

A semi-definite program (SDP) is a convex optimization problem that seeks to minimize (or maximize) a linear objective function over the convex cone of symmetric and positive semidefinite matrices subject to linear constraints. The canonical form of a SDP is:

$$\begin{aligned} \underset{X}{argmin} \quad & \text{trace}(KX) \\ \text{such that} \quad & \text{trace}(A_i X) = b_i \quad \text{for } i = 1, \ldots, m \\ & X \succeq 0 \end{aligned}$$

where the constraint $X \succeq 0$ restricts X to positive semidefinite matrices.

The advantage of formulating spectral clustering problems as a SDP is that the relaxed solution of the SDP formulation is tighter than the relaxed solution provided by spectral clustering. Also, additional linear constraints can be added to the SDP formulation. Previous work using SDP to approximate graph optimization problems includes Max-Cut and graph partitioning [9][10][15]. Peng reformulates the spectral clustering problem as a 0-1 SDP [18] which, as an integer program, is NP hard but can also be relaxed. Normalized spectral clustering is still more desirable because clusters are balanced and partitions of single nodes are avoided. Below is the relaxed SDP formulation for normalized spectral clustering given by Xing and Jordan [27]:

$$\begin{aligned} \underset{Z}{argmin} \quad & \text{trace}(\bar{L}Z) \\ \text{such that} \quad & Z\text{diag}(D^{1/2}) = \text{diag}(D^{1/2}) & (2) \\ & Z \geq 0 \text{ (elementwise)} & (3) \\ & \text{trace } (Z) = k & (4) \\ & Z \succeq 0, Z = Z^t & (5) \\ & I - Z \succeq 0 & (6) \end{aligned}$$

$\bar{L}$ is the normalized Laplacian defined earlier. $Z \in \mathbb{R}^{n \times n}$ is a positive semidefinite co-occurrence matrix and the solution to the relaxed SDP formulation. Constraint (2) ensures that each node is assigned to one cluster; (3) and (5) are from relaxing the SDP formulation; (4) corresponds to the k clusters desired for segmentation; and (6) is from the defintion of Y.

Z is decomposed into $Z = YY^T$ where $Y \in \mathbb{R}^{n \times k}$ yields the k clusters of the graph. The factor Y has the following properties: $Y^T Y = I$ and $Y = D^{1/2} X (X^T D X)^{-1/2}$. Y is obtained by applying a singular value decomposition to Z and using the top k singular vectors. We then apply the randomized rounding scheme with restarts to obtain the segmentation.

A disadvantage of using the SDP formulations is that the formulation cannot handle large matrices. However, low-rank SDP formulations do exist [16][3]. Another disadvantage is that the solution Z from the relaxed formulation takes on real values in the interval [-1,1] instead of integers $\{0, 1\}$. We present a rounding scheme in Section 5 that allows us to round the real values of the solutions to more interpretable integers for segmentation.

4. ADDING TEMPLATES TO SEGMENTING

Background knowledge can take the form of pair-wise similarities via a Laplacian or a co-occurrence matrix. We can use the background knowledge we have to create templates for desired clusters. For example, we may want to find a segmentation of the graph that includes the template of co-workers who enjoy the same outdoor activities. Background knowledge can be added to the SDP formulation for normalized spectral clustering as additional linear constraints. The template constraints can simply be added to the SDP formulation:

$$\begin{array}{ll} \underset{Z}{argmin} & \text{trace}(\bar{L}Z) \\ \text{such that} & \text{normalized spectral clustering constraints} \\ & \text{template constraint 1} \\ & \text{template constraint 2} \end{array}$$

The template constraints are of the form $\text{trace}(KZ) \geq \beta$. The value of the constant β determines whether the segmentation follows strict or loose adherence to the template constraint. Examples of different types of matrix K for template constraints and the effect of strict and loose adherence to the constraints on segmentations are shown Section 4.1. Explanation and interpretation of the constant β for template constraints is in Section 4.2.

4.1 Types of Templates

We present four types of template constraints: topological groups, characteristics of people, alternative segmentations, and multiple metrics. In each section, we offer an illustrative example to explain the affect of each template constraint on the segmentation. While the SDP formulation for spectral clustering can find a segmentation of a graph for k-clusters, our examples will be for $k = 2$. The same graph is used for all examples and the image of the graph and its normalize spectral clustering segmentation is shown in Figure (1).

4.1.1 Topological Groups/Structures

It may be desirable to preserve known relationships in a social network such as family members, co-workers, or rival sports teams. Such relationships can be encoded in a matrix. Let the matrix $M = \{m_{ij}\}$, where $m_{ij} \in \{-1, 1\}$, be the $n \times n$ matrix that holds the together and apart relations. The template constraint is then $\text{trace}(MZ) \geq \beta$, where $\beta > 0$ is some constant (interpretation of $\text{trace}(MZ)$ and β are in Section 4.2). Consider the networks in Figure (2). The red square nodes indicate nodes which are known to have a relationship that the user would like to preserve. The graph on the left shows strong adherence to the template constraint since all the red square nodes are in the same cluster. The graph on the right shows a looser adherence to the template constraint since one red square node is in a different cluster. Note that the right graph (with the looser adherence) has a segmentation that is close to the normalized segmentation.

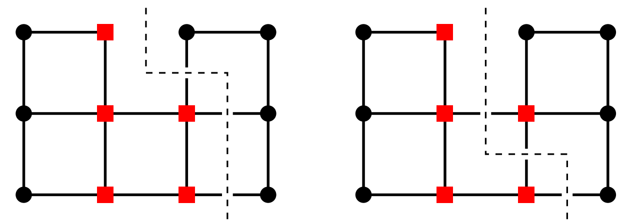

Figure 2: Example of template clustering using information from known relationships between a few nodes. (Section 4.1.1). The left graph shows a strong adherence to the template constraint. The right graph shows a looser adherence. The red square nodes are associated with the template constraint for groups.

4.1.2 Characteristics of People

We can encode k binary labels in k indicator vectors. For indicator vector v, $v_i = 1$ if node i has the label and $v_i = -1$ if the node does not have the label. The labels may encode the likes and dislikes of people in a social network such as which people enjoy mountain biking, rock music, or cooking. We can implement the labels as a rank-1 matrix $M = \{m_{ij}\}$, where $m_{ij} \in \{0, 1\}$, which is created from the outer product of the indicator matrix. The template constraint is then $\text{trace}(MZ) \geq \beta$, where $\beta > 0$ is some constant. Consider the two graphs in Figure (3). The red square nodes are nodes with labels on them while the labels for the black circle nodes are unknown. The graph on the left shows the resulting segmentation from strong adherence to the template constraint with all the red square nodes in the same cluster. The right network shows a looser adherence to the template constraint as red square and black circle nodes are clustered together instead of separately.

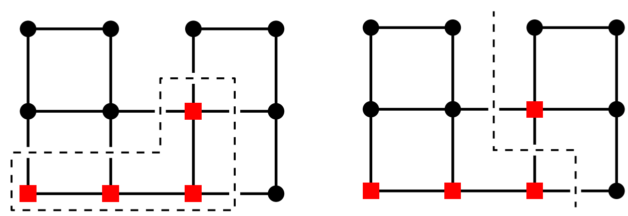

Figure 3: Example of template clustering using information from labels on nodes (Section 4.1.2). The left graph shows a strong adherence to the constraint. The right graph shows a looser adherence. The red square nodes are associated with the template constraint for labels.

4.1.3 Alternative Segmentations

There may be cases where we have a cut of a graph based on another kernel. We can then use the information of the cut as a template constraint. Let $M = \{m_{ij}\}$, where $m_{ij} \in \{-1, 1\}$, be the co-occurrence matrix that encodes the other segmentation. The template constraint is then $\text{trace}(MZ) \geq \beta$, where $\beta > 0$ is some constant. The higher value of the β, the stronger the solution Z will adhere to the constraint. Consider the graphs in Figure (4). The blue triangle nodes represent an alternative clustering. The left graph shows strong adherence to the template constraint: the blue triangle nodes are clustered together, separate from the rest of the graph. The right graph shows a looser adherence to the template constraint: the blue nodes are clustered together but a black circle node is allowed as part of the cluster.

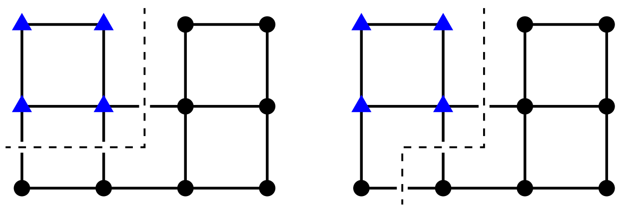

Figure 4: Example of template clustering using information from another segmentation (Section 4.1.3). The left graph shows strong adherence to the template constraint. The right graph shows a looser adherence. The blue triangle nodes are associated with the template constraint for an alternative segmentation.

4.1.4 Multiple Distance Metrics or Views

The affiliations of a social network can also be measured with different metrics. For example, one metric may represent work affiliations while another metric represents friendships. We may wish to find a segmentation that partially adheres to the secondary metric. The template constraint is trace$(MZ) \geq \beta$, where $\beta > 0$ is some constant. Consider the graphs in Figure (5). The graph on the left represents the same set of nodes as the graph on the right but the edges are determined by a some other metric. The graph on the right shows the resulting segmentation when the secondary metric is incorporated as a template constraint.

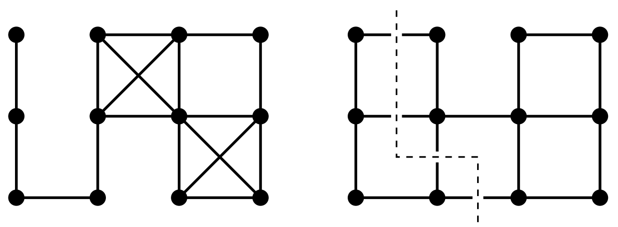

Figure 5: Example of template clustering using information from another metric for the same graph (Section 4.1.4). The left graph represents the same graph but with a different metric. The right graph is the resulting cut.

4.2 Interpreting constraints/constants

The four types of template constraints we have presented take the form of trace$(KZ) \geq \beta$ where K is a matrix representing the additional information of the graph, such as known relationships or labels, and $\beta > 0$ is a constant. An interpretation of this constraint is that we want the segmentation suggested by Z to match the “assignments” given by K at least β times. That is, at least β nodes of the graph are clustered together and match the information given by K. Note that $\beta \in [0, m]$ where $m = \max_i |\sum_i^n K_{ij}|$.

Note that for small values of β, the segmentation will favor the normalized spectral clustering. That is, the template constraint will be loosely adhered to. In fact, $\beta = 0$ will produce the normalize cut of the graph. Conversely, large values of β will favor the template constraint. For example, the graphs in Figure (3) have nodes with known labels (shown as red square nodes). The graph on the left shows strict adherence to the template constraint by clustering all the red square nodes together. The graph on the right shows the same formulation but with a smaller β value for the template constraint. A majority of the red square nodes are clustered together and the normalized cut is more closely followed. However, the cut deviates from the normalized cut to cluster three of the red square nodes together.

4.3 Combining Templates

The SDP formulation allows for the addition of more than one constraint. Thus, a user could construct a template composition of several different instances of each type of template constraint presented in Section 4.1. However, one can do more than add multiple instances of one type of constraint such as using a characteristic constraint with a grouping constraint.

Consider the following example. Two template constraints are used: Constraint 1 for labels on nodes represented by red square nodes (Section 4.1.2) with bound β_1 and Constraint 2 for alternative segmentations represented by blue triangle nodes (Section 4.1.3) with bound β_2. The affect of changing the values for β_1 and β_2 are shown in Figure (6). The graph on the left is the result of enforcing a stronger adherence to the alternative clustering template constraint via a larger β_2 value. The graph on the right is the result of enforcing a stronger adherence to the labels on nodes template constraint via a larger β_1 value.

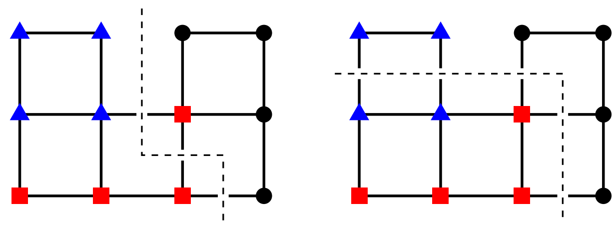

Figure 6: Example of template clustering using information from two types of constraints: alternative segmentation and labels. The left graph shows a stronger adherence to the template constraint for the blue triangle nodes. The right graph shows a stronger adherence to the template constraint for red square nodes.

5. MAKING TEMPLATES INTERPRETABLE

5.1 Rounding Schemes

The solution to the relaxed SDP formulation for spectral clustering must be rounded from real values to integers for cluster assignments. Rounding schemes are used to recover the feasible solution partition matrix $X = \{x_{ij}\}$ where $x_{ij} \in \{0, 1\}$ A drawback of rounding is that the rounded solution may no longer satisfy the template constraints. Some rounding schemes include:

1. Directional cosine method [4]:
   Orthogonal projection from n-dimensions to k dimensions

2. Randomized Projection Heuristic Method [9]:
   Take rows of Z and project them to a random lower dimensional space.

3. Rounding by Clustering:
   While treating each row of Z as a vertex, use a clustering method, such as k-mean clustering, to obtain X.

However, none of these rounding schemes provide bounds on the quality of the rounded solution. We now present, to our knowledge, some of the only work that attempts to do this. We present a randomized rounding scheme for recovering the solution X that satisfies the un-relaxed normalized spectral clustering problem with template constraints. Let $u_i : u_{ij} \in [-1, 1]$ be the solution to the relaxed SDP for normalized spectral clustering. We can apply the following randomized rounding scheme to vector u to get solution v, $v_i \in \{-1, 1\}$ where v is determined probabilistically as a function of u_i:

$$v_i = \begin{cases} 1 & \text{with probability } (1 + u_i)/2 \\ -1 & \text{with probability } (1 - u_i)/2 \end{cases}$$

We note the following well known results:

$$E[v_i] = u_i$$
$$Var[v_i] = 1 - u_i^2$$

We now wish to determine bounds to answer two questions:

1. If we apply the rounding scheme how will the quality of the solution be affected?

2. How many times should the rounding scheme be applied in order to satisfy the bounds.

One insight we shall empirically validate is that for question (1) the mean quality of the solution after applying the rounding scheme will be the same as the quality of the solution which is found, as shown below.

$$\begin{aligned} E[\text{trace}(KV)] &= E\left[\sum_i l_i v_i^T\right] = \sum_i l_i E\left[v_i^T\right] = \sum_i l_i u_i^T \\ &= \text{trace}(KU) \qquad \text{where } K \text{ is some kernel} \end{aligned}$$

However, an answer to question (2) is not so straight forward. As mentioned before, $\text{trace}(KV) = \sum_{i=1}^n K_{i*}V_{*i}$ is essentially interpreted as the agreement between the labeling (kernel K) and segmentation (solution V) schemes. This can be broken down into n summations whose total value is greater than β. Consider the instance giving the worst such agreement which we know will have numerical value less than β/n, that is $argmin_j K_{j*}V_{*j} < \beta/n$. Let q be the index that minimizes this expression. We wish to determine how many times we should apply our randomized rounding scheme so that its value is greater than β/n.

Let $\epsilon = |K_{q*}V_{*q} - \beta/n|$. Note that ϵ cannot exceed one since β is at most the number of nodes having the label which is always less than n and the least that $K_{q*}V_{*q}$ can be is 0. We wish our randomized rounding scheme, when applied to each point, to have a sum greater than ϵ. Given that our data set consists of n points, we require that $\frac{1}{n}\sum_i K_{i*}V_{*i} > \epsilon$: that is, the rounding scheme will on average need to exceed the worst possible case.

We can now define the Chernoff inequality to determine the chance our rounding scheme will satisfy the template constraint inequality:

$$P\left[\theta - \frac{1}{n}\sum_i u_i > \epsilon\right] \leq e^{-2n\epsilon^2}$$

$$P\left[K_{q*}V_{*q} - \frac{1}{n}\sum_i \text{tr}\,(LU_i) > \epsilon\right] \leq e^{-2n\epsilon^2} \qquad (7)$$

Where n is the number of instances in the data set.

This gives us the chance that **one application** of the rounding scheme will satisfy the bound. We wish to obtain at least one such success (we can easily empirically verify the success) which is given by the binomial distribution $1 - f(0; n, p) = \binom{n}{0} p^0 (1-p)^{n-0} = (1-p)^n$. We can then determine the number of times one must use the randomize rounding scheme so that V satisfies template constraint with a certain confidence.

6. EXPERIMENTAL RESULTS

The purpose of this experimental section is two fold:

- To verify the correctness of our randomized rounding scheme.
- Empirically verify the bound in Equation (7).

The first purpose is to show that the composition of the clustering does indeed satisfy the right hand side of the bound. For example, for the constraint $\text{trace}(MV) \geq \beta$, where matrix M encodes the characteristics of people (Section 4.1.2), we are stating our clustering must contain a cluster with at least β people with the characteristic represented by M. The second purpose is required to determine the tightness of our bound in Equation (7). Since this is an upper bound we wish to test how tight the bound is.

In this paper, we use a subset of a Facebook data set[1] (analysis of data set in [2]) consisting of 92 individuals. Each individual has a number of possible labels associated with him/her: current location, undergraduate university, graduate university, and current employer.

For our first experiment, we label those people who live in Washington, DC. We wish to cluster the network according to the friendship links but also require all nine people who live in Washington, DC to be in the same cluster. To achieve this we created a indicator label vector l of 0's and 1's to identify those people who live in Washington, DC. We then take the outer product $M = ll^T$ to obtain a matrix that is positive semi-definite. Since we want all nine people to be in the same cluser we enforce the template constraint: $\text{trace}(MV) \geq \beta \geq 9$. According to the bound in Equation (7), we can generate the probability of the randomized rounding scheme satisfying the requirement that all 9 people

[1]Obtained from the Davis Social Links project `http://dsl.ucdavis.edu/lab_website/`)

Label set	From Wash., DC	Went to UVA
Upper bound of rounding scheme success from Eqn (7)	82.81%	88.63%
Empirically how often the rounding scheme satisfied trace$(MV) \geq \beta$	65%	43%
Empirically how often rounding scheme clustered labeled nodes together	3%	25%

Table 1: Results of using our randomize rounding scheme on a Facebook data set with labels. 100 trials of the rounded scheme were used and successes were measured by whether the rounded solution satisfied trace$(MB) \geq \beta$ and if the labeled nodes were clustered together.

from Virginia are in the same cluster. We can then empirically compare what proportion of times the rounding scheme when applied satisfied the constraint to verify the correctness of the bound. For each time the rounding scheme is successful we can then verify the composition of the clustering does indeed have the nine people from Virginia in the same cluster. Results are shown in Table 1.

We repeat this experiment for another label, people who went to the University of Virginia (UVA) of which there are 10 such people in our data set. trace$(MV) \geq \beta \geq 10$. The template constraint is Experimental results are shown in Table 1.

Based on our experiments, we can conclude that for small β, the bound is loose. For the first case where the number of labeled nodes is nine, the predicted upper bound of success is 82.81% and was empirically verified to be 65%. For the second case where the number of labeled nodes is 10, the predited upper bound of success is 88.63% but was verified empirically to be 43%. Furthermore, we empirically verified than when the bound was satisfied after randomized rounding that indeed all 9 or 10 people from Washington, D.C. and who went to UVA respectively were in the same cluster.

7. CONCLUSION AND FUTURE WORK

Templates allows us to impose our background expectations on the clustering of a social network. It allows us to specify the composition of the segmentations with respect to the properties of the people in the network not just on their friendship structure. However, adding multiple constraints to spectral clustering is difficult. Methods which utilize the information of other kernels of a network can only do so by taking a linear combination of those kernels and then applying spectral clustering techniques to the new kernel.

We presented a novel approach to clustering small social networks with template compositions using a relaxed SDP formulation for normalized spectral clustering. The SDP formulation allows the addition of multiple constraints and provides solutions that satisfy template compositions. We showed the possible types of template constraints, the ways in which they are implemented in the SDP, and the ways in which the user can enforce strict or loose adherence to the constraint. We also addressed the drawback of the SDP formulation, which is that the rounded solution may no longer satisfy the template constraints, by presenting a randomized rounding scheme. We showed how the SDP for spectral clustering with template constraints and the randomized rounding scheme can be applied to a small social network.

Future work includes implementing other types of template constraints such as grouping constraints and using alternative cut information and also implementing combinations of two or more of different template constraint types such as a constraint for characteristics of people and for alternative cut information. Finally, applying the template constraints with a fast SDP or low-rank SDP formulations to test constraints for larger data sets.

8. REFERENCES

[1] A. Argyriou, M. Herbster, M. Herbster, M. Pontil, and M. Pontil. Combining graph laplacians for semi-supervised learning. In *Advances in Neural Information Processing Systems 18*, pages 67–74. MIT Press, 2005.

[2] P. Bhattacharyya, A. Garg, and S. F. Wu. Analysis of user keyword in online social networks. In *In Social Networks Analysis and Mining Journal (by Springer)*, 2010.

[3] T. D. Bie, N. Cristianini, P. Bennett, and E. Parrado-hernández. Fast sdp relaxations of graph cut clustering, transduction, and other combinatorial problems. *JMLR*, 7:1409–1436, 2006.

[4] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *Proceedings of the 30th international Design Automation Conference*, DAC '93, pages 749–754, New York, NY, USA, 1993. ACM.

[5] F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, Feb. 1997.

[6] T. Coleman, J. Saunderson, and A. Wirth. Spectral clustering with inconsistent advice. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 152–159. ACM, 2008.

[7] I. Davidson and Z. Qi. Finding alternative clusterings using constraints. In *ICDM'08*, pages 773–778, 2008.

[8] T. De Bie, J. Suykens, and B. D. Moor. Learning from general label constraints. In *SSPR/SPR*, pages 671–679, 2004.

[9] A. M. Frieze and M. Jerrum. Improved approximation algorithms for max k-cut and max bisection. In *Proceedings of the 4th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 1–13, London, UK, 1995. Springer-Verlag.

[10] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *JACM*, 42:1115–1145, November 1995.

[11] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, Apr. 2011.

[12] M. Heiler, J. Keuchel, and C. Schnörr. Semidefinite clustering for image segmentation with a-priori knowledge. In *DAGM-Symposium*, pages 309–317,

2005.
[13] T. Joachims, T. J. De, N. Cristianini, and N. R. A. Uk. Composite kernels for hypertext categorisation. In *In Proceedings of the International Conference on Machine Learning (ICML*, pages 250–257. Morgan Kaufmann Publishers, 2001.
[14] S. Kamvar, D. Klein, and C. Manning. Spectral learning. In *In IJCAI*, pages 561–566, 2003.
[15] S. E. Karisch and F. Rendl. Semidefinite programming and graph equipartition. In *In Topics in Semidefinite and Interior-Point Methods*, pages 77–95. AMS, 1998.
[16] B. Kulis. Fast low-rank semidefinite programming for embedding and clustering. In *in Eleventh International Conference on Artifical Intelligence and Statistics, AISTATS 2007*, page 2007, 2007.
[17] Z. Lu and M. Carreira-Perpinan. Constrained spectral clustering through affinity propagation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, june 2008.
[18] J. Peng and Y. Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM J. on Optimization*, 18:186–205, February 2007.
[19] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 731–, Washington, DC, USA, 1997. IEEE Computer Society.
[20] V. Sindhwani and P. Niyogi. A co-regularized approach to semi-supervised learning with multiple views. In *Proceedings of the ICML Workshop on Learning with Multiple Views*, 2005.
[21] W. Tang, Z. Lu, and I. S. Dhillon. Clustering with multiple graphs. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 1016–1021, Washington, DC, USA, 2009. IEEE Computer Society.
[22] D. Tolliver, G. L. Miller, and R. Collins. Corrected laplacians: closer cuts and segmentation with shape priors. In *Computer Vision and Pattern Recognition, CVPR*, volume 2, pages 92–98. IEEE, June 2005.
[23] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, December 2007.
[24] F. Wang, C. Ding, and T. Liz. Integrated kl (k-means - laplacian) clustering: A new clustering approach by combining attribute data and pairwise relations. *SDM*, pages 38–48, 2009.
[25] X. Wang and I. Davidson. Discovering contexts and contextual outliers using random walks in graphs. In *ICDM'09*, pages 1034–1039, 2009.
[26] X. Wang and I. Davidson. Flexible constrained spectral clustering. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 563–572, New York, NY, USA, 2010. ACM.
[27] E. P. Xing and M. I. Jordan. On semidefinite relaxation for normalized k-cut and connections to spectral clustering. Technical Report UCB/CSD-03-1265, EECS Department, University of California, Berkeley, Jun 2003.
[28] Q. Xu, M. desJardins, and K. Wagstaff. Constrained spectral clustering under a local proximity structure assumption. In *FLAIRS Conference*, pages 866–867, 2005.
[29] S. X. Yu and J. Shi. Grouping with bias. In *In Advances in Neural Information Processing Systems*. MIT Press, 2001.
[30] S. X. Yu, I. C. Society, J. Shi, and I. C. Society. Segmentation given partial grouping constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:173–183, 2004.
[31] T. Zhang, A. Popescul, and B. Dom. Linear prediction models with graph regularization for web-page categorization. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 821–826, New York, NY, USA, 2006. ACM.