

# Flexible Constrained Spectral Clustering

Xiang Wang  
Department of Computer Science  
University of California, Davis  
Davis, CA 95616  
xiang@ucdavis.edu

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

## ABSTRACT

Constrained clustering has been well-studied for algorithms like  $K$ -means and hierarchical agglomerative clustering. However, how to encode constraints into spectral clustering remains a developing area. In this paper, we propose a *flexible* and *generalized* framework for constrained spectral clustering. In contrast to some previous efforts that implicitly encode MUST-LINK and CANNOT-LINK constraints by modifying the graph Laplacian or the resultant eigenspace, we present a more natural and principled formulation, which preserves the original graph Laplacian and explicitly encodes the constraints. Our method offers several practical advantages: it can encode the degree of belief (weight) in MUST-LINK and CANNOT-LINK constraints; it guarantees to lower-bound how well the given constraints are satisfied using a user-specified threshold; and it can be solved deterministically in polynomial time through generalized eigendecomposition. Furthermore, by inheriting the objective function from spectral clustering and explicitly encoding the constraints, much of the existing analysis of spectral clustering techniques is still valid. Consequently our work can be posed as a natural extension to unconstrained spectral clustering and be interpreted as finding the normalized min-cut of a *labeled* graph. We validate the effectiveness of our approach by empirical results on real-world data sets, with applications to constrained image segmentation and clustering benchmark data sets with both binary and degree-of-belief constraints.

## Categories and Subject Descriptors

H.2.8 [Database Applications]: Data Mining

## General Terms

Algorithms, Experimentation

## Keywords

Spectral clustering, constrained clustering

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.

KDD'10, July 25–28, 2010, Washington, DC, USA.

Copyright 2010 ACM 978-1-4503-0055-110/07 ...\$10.00.

## 1. INTRODUCTION

Constrained clustering is a category of techniques that try to incorporate user supervision (side information) into existing clustering algorithms [2]. The typical form of pairwise constraints are MUST-LINK (the pair of points must be assigned into the same cluster, ML for short) and CANNOT-LINK (the pair of points cannot be assigned into the same cluster, CL for short). These types of constraints have been added into many popular clustering algorithms such as  $K$ -means clustering, mixture modeling, hierarchical clustering and density-based clustering [2]. However, constrained spectral clustering remains a developing area.

Spectral clustering is an important clustering technique that has been extensively studied in the image processing, data mining, and machine learning communities [13–15]. It is considered superior to traditional clustering algorithms like  $K$ -means in terms of having deterministic and polynomial-time solution and its equivalence to graph min-cut problems. Its advantage has also been validated by many real-world applications, such as image segmentation [14] and mining social networks [18].

The aim of this paper is to combine spectral clustering and pairwise constraints in a principled and flexible manner. Most of the existing techniques on constrained spectral clustering can be categorized into two different types, based on how they enforce the constraints. The first type of methods [7,8,11,17,19] directly manipulate the graph Laplacian (or equivalently, the affinity matrix) according to the given constraints; then unconstrained spectral clustering is applied on the modified graph Laplacian. The second type of methods use constraints to restrict the feasible solution space. For example, the subspace trick introduced by De Bie et al. [5] alters the resultant eigenspace which the clustering assignment vector will be projected onto, based on given constraints. This technique was later extended in [3] to accommodate inconsistent constraints. Yu and Shi [20,21] encoded partial grouping information as a subspace projection. Li et al. [10] enforced constraints via the regularization of the spectral embedding.

The aforementioned approaches have two limitations:

1. They are designed to handle binary ML and CL constraints. However, there are many applications where the constraints are often provided in the form of real-valued numbers (degree of belief) rather than yes/no verdicts. For example in a collaborative tagging application, constraints are provided by a group of users who do not necessarily agree with each other. So it is more natural to measure the likeliness of two instances

belonging to the same cluster with a real-valued score, according to the ratio of users who agree so.

2. They aim to satisfy as many constraints as possible, which could be unnecessary and difficult in practice. For example, constraints provided by multiple users could be inconsistent, thus they cannot be all satisfied at one time. Also it is reasonable to ignore a small portion of constraints in exchange for a clustering with much lower cost.

In this paper, we go beyond binary ML/CL constraints and propose a more flexible framework to accommodate general-type user supervision. The binary constraints are relaxed to become a degree of belief (real-valued) that two data instances belong to the same class or two different classes. Moreover, instead of trying to satisfy each and every constraint that has been given, we use a user-specified threshold to lower-bound how well the given constraints are satisfied. Therefore, **our method provides maximum flexibility in terms of both representing constraints and satisfying them.**

Specifically, we formulate constrained spectral clustering as a constrained optimization problem by adding a new constraint to the original objective function of spectral clustering (see Section 3.1). Then we show that our objective function can be converted it into a generalized eigenvalue system, which can be solved deterministically in polynomial time (see Section 3.2). This should be considered as a major advantage over constrained  $K$ -means clustering, which produces non-deterministic solutions while being intractable even for  $K = 2$  [4, 6]. Furthermore, our algorithm (see Section 4) guarantees to find the solution if one exists, which is not the case for many constrained clustering algorithms that are sensitive to the ordering of the constraints (See Fig. 1 in [4] for a concrete example). We validate the effectiveness of our approach on several real-world data sets in Section 5. The results of image segmentation (see Fig. 3 and 4) show that our method can produce semantically meaningful clustering that conforms to human intuition and expectation. The results of clustering benchmarks (see Fig. 5) show quantitatively that our method can significantly improve the resultant clustering with given constraints.

Our contributions are:

- We are the first to incorporate user supervision into spectral clustering that allows real-valued degree-of-belief constraints.
- We introduce a user-specified threshold to indicate the importance of user supervision, so that some of the constraints can be ignored in exchange for lower clustering cost (whereas previous approaches tried to satisfy all consistent constraints).
- We inherit the original objective function of spectral clustering while encoding constraints explicitly and creating a novel constrained optimization problem (Eq.(3)).
- Our approach can be viewed as a natural extension to the original spectral clustering formulation: we show how our approach (see Section 3.5) can be interpreted as finding the normalized min-cut of a *labeled* graph.
- We validate the effectiveness of our approach and its advantage over existing methods using standard benchmarks (see Section 5).

**Table 1: Table of notations**

Symbol	Meaning
$A$	The affinity matrix
$D$	The degree matrix
$I$	The identity matrix
$L(\bar{L})$	The (normalized) graph Laplacian
$Q(\bar{Q})$	The (normalized) constraint matrix

## 2. BACKGROUND AND PRELIMINARIES

In this paper we follow the standard graph model that is commonly used in the spectral clustering literature. We reiterate some of the definitions and properties in this section, such as graph Laplacian, normalized min-cut, eigendecomposition and so forth, to make this paper self-contained. Readers who are familiar with the materials can skip to our contributions in Section 3. Important notations used throughout the rest of the paper are listed in Table 1.

A collection of  $N$  data instances is modeled by an undirected, weighted graph  $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$ , where each data instance corresponds to a vertex (node) in  $\mathcal{V}$ ;  $\mathcal{E}$  is the edge set and  $A$  is the associated affinity matrix.  $A$  is symmetric and non-negative. The diagonal matrix  $D = \text{diag}(D_{11}, \dots, D_{NN})$  is called the degree matrix of graph  $\mathcal{G}$ , where

$$D_{ii} = \sum_{j=1}^N A_{ij}.$$

Then

$$L = D - A$$

is called the graph Laplacian of  $\mathcal{G}$ . Assuming  $\mathcal{G}$  is connected (i.e. any node is reachable from any other node),  $L$  has the following properties:

**PROPERTY 1. (*Properties of graph Laplacian [15]*)**  
*Let  $L$  be the graph Laplacian of a connected graph, then we have:*

1.  $L$  is symmetric and positive semi-definite.
2.  $L$  has one and only one eigenvalue equal to 0, and  $N-1$  positive eigenvalues:  $0 = \lambda_0 < \lambda_1 \leq \dots \leq \lambda_{N-1}$ .
3.  $\mathbf{1}$  is an eigenvector of  $L$  with eigenvalue 0 ( $\mathbf{1}$  is a constant vector whose entries are all 1).

Shi and Malik [14] showed that the eigenvector of  $L$  associated with the second smallest eigenvalue  $\lambda_1$  solves the normalized min-cut (N-CUT) problem of graph  $\mathcal{G}$  (in a relaxed sense). The objective function can be written as:

$$\arg \min_{\mathbf{u} \in \mathbb{R}^N} \mathbf{u}^T L \mathbf{u}, \text{ s.t. } \mathbf{u}^T D \mathbf{u} = \text{vol}(\mathcal{G}), D \mathbf{u} \perp \mathbf{1}, \quad (1)$$

where  $\text{vol}(\mathcal{G}) = \sum_{i=1}^N D_{ii}$ . Note that in Eq.(1),  $\mathbf{u}$  is the relaxed cluster indicator vector;  $\mathbf{u}^T L \mathbf{u}$  is the cost of the cut, which is to minimize; the first constraint  $\mathbf{u}^T D \mathbf{u} = \text{vol}(\mathcal{G})$  normalizes the cluster indicator vector  $\mathbf{u}$ ; the second constraint  $D \mathbf{u} \perp \mathbf{1}$  rules out the principal eigenvector of  $L$  as a trivial solution, because it does not define a meaningful cut on the graph.

In the rest of paper, for the simplicity of notation, we use an equivalent objective function used in [15]. We substitute

$\mathbf{u}$  by  $D^{-1/2}\mathbf{v}$ , then Eq.(1) becomes:

$$\arg \min_{\mathbf{v} \in \mathbb{R}^N} \mathbf{v}^T \bar{L} \mathbf{v}, \text{ s.t. } \mathbf{v}^T \mathbf{v} = \text{vol}(\mathcal{G}), \mathbf{v} \perp D^{1/2} \mathbf{1}. \quad (2)$$

Here

$$\bar{L} = D^{-1/2} L D^{-1/2}$$

is called the *normalized* graph Laplacian [15]. Again, Eq.(2) is equivalent to Eq.(1) since  $\mathbf{v}^*$  is the optimal solution to Eq.(2) if and only if  $\mathbf{u}^* = D^{-1/2} \mathbf{v}^*$  is the optimal solution to Eq.(1).

Note that the result of spectral clustering is solely decided by the affinity structure of graph  $\mathcal{G}$  as encoded in the matrix  $A$  (and thus the graph Laplacian  $L$ ). We will then describe our extensions on how to incorporate additional supervision so that the result of clustering will reflect both the affinity structure of the graph and the structure of the constraint information.

### 3. OUR PROBLEM FORMULATION

In this section, we show how to incorporate user supervision into spectral clustering, whose objective function is shown as in Eq.(2). We encode supervision in such a way that we not only allow binary CL/ML constraints, but also a real-valued degree of belief that two data instances belong to the same cluster or two different ones. We propose a new objective function for constrained spectral clustering, which is formulated as a constrained optimization problem. Then we show how to solve the objective function by converting it into a generalized eigenvalue system. Note that the unconstrained spectral clustering problem can be interpreted as the N-CUT of an unlabeled graph. Similarly, our formulation can be interpreted as the N-CUT of a labeled graph.

#### 3.1 Objective Function

We encode user supervision with an  $N \times N$  constraint matrix  $Q$ . Traditionally, constrained clustering only uses binary constraints: MUST-LINK and CANNOT-LINK, which can be naturally encoded as follows:

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

Let  $\mathbf{u} \in \{-1, +1\}^N$  be a cluster indicator vector, where  $\mathbf{u}_i = +1$  if node  $i$  belongs to cluster  $+$  and  $\mathbf{u}_i = -1$  if node  $i$  belongs to cluster  $-$ , then

$$\mathbf{u}^T Q \mathbf{u} = \sum_{i=1}^N \sum_{j=1}^N \mathbf{u}_i \mathbf{u}_j Q_{ij}$$

is a measure of how well the constraints in  $Q$  are satisfied by the cluster assignment  $\mathbf{u}$ : the measure will increase by 1 if  $Q_{ij} = 1$  and node  $i$  and  $j$  have the same sign in  $\mathbf{u}$ ; the measure will decrease by 1 if 1)  $Q_{ij} = 1$  but node  $i$  and  $j$  have different signs in  $\mathbf{u}$ , or 2)  $Q_{ij} = -1$  but node  $i$  and  $j$  have the same sign in  $\mathbf{u}$ .

Now to accommodate degree-of-belief constraints, we simultaneously relax the cluster indicator vector  $\mathbf{u}$  and the constraint matrix  $Q$  such that:

$$\mathbf{u} \in \mathbb{R}^N, Q \in \mathbb{R}^{N \times N}.$$

$Q_{ij}$  is positive if we believe nodes  $i$  and  $j$  belong to the same class;  $Q_{ij}$  is negative if we believe nodes  $i$  and  $j$  belong to

different classes; the magnitude of  $Q_{ij}$  indicates how strong the belief is.

Consequently,  $\mathbf{u}^T Q \mathbf{u}$  becomes a real-valued measure of how well the constraints in  $Q$  have been satisfied, in the relaxed sense. For example,  $Q_{ij} < 0$  means we believe nodes  $i$  and  $j$  belong to different classes, then in order to improve  $\mathbf{u}^T Q \mathbf{u}$ , we should assign  $\mathbf{u}_i$  and  $\mathbf{u}_j$  with values of different signs; similarly,  $Q_{ij} > 0$  means nodes  $i$  and  $j$  are believed to belong to the same class, then we should assign  $\mathbf{u}_i$  and  $\mathbf{u}_j$  with values of the same sign. The larger  $\mathbf{u}^T Q \mathbf{u}$  is, the better the cluster assignment  $\mathbf{u}$  conforms to the given constraints in  $Q$ .

Now given this real-valued measure, rather than trying to satisfy all the constraints given in  $Q$ , we can lower-bound this measure with a constant  $\alpha \in \mathbb{R}$ :

$$\mathbf{u}^T Q \mathbf{u} \geq \alpha.$$

By substituting  $\mathbf{u}$  by  $D^{-1/2}\mathbf{v}$ , above inequality becomes

$$\mathbf{v}^T \bar{Q} \mathbf{v} \geq \alpha,$$

where

$$\bar{Q} = D^{-1/2} Q D^{-1/2}$$

is the *normalized* constraint matrix.

We append this lower-bound constraint to the objective function of unconstrained spectral clustering in Eq.(2), and we have:

*Problem 1. (Constrained Spectral Clustering)* Given a normalized graph Laplacian  $\bar{L}$ , a normalized constraint matrix  $\bar{Q}$  and a threshold  $\alpha$ , we want to optimize the following objective function:

$$\arg \min_{\mathbf{v} \in \mathbb{R}^N} \mathbf{v}^T \bar{L} \mathbf{v}, \text{ s.t. } \mathbf{v}^T \bar{Q} \mathbf{v} \geq \alpha, \mathbf{v}^T \mathbf{v} = \text{vol}(\mathcal{G}), \mathbf{v} \neq D^{1/2} \mathbf{1}. \quad (3)$$

Here  $\mathbf{v}^T \bar{L} \mathbf{v}$  is the cost of the cut, which is to minimize; the first constraint  $\mathbf{v}^T \bar{Q} \mathbf{v} \geq \alpha$  is to lower-bound how well the constraints in  $Q$  are satisfied; the second constraint  $\mathbf{v}^T \mathbf{v} = \text{vol}(\mathcal{G})$  normalizes  $\mathbf{v}$ ; the third constraint  $\mathbf{v} \neq D^{1/2} \mathbf{1}$  rules out the trivial solution  $D^{1/2} \mathbf{1}$ . Suppose  $\mathbf{v}^*$  is the optimal solution to Eq.(3), then  $\mathbf{u}^* = D^{-1/2} \mathbf{v}^*$  is the optimal cluster indicator vector.

It is easy to see that the unconstrained spectral clustering in Eq.(2) can be covered as a special case of our formulation where  $\bar{Q} = I$  and  $\alpha = \text{vol}(\mathcal{G})$ .

#### 3.2 Solving the Objective Function

To solve a constrained optimization problem, we normally use the Karush-Kuhn-Tucker Theorem [9], which describes the necessary conditions for the optimal solution to the problem. We can derive a set of candidates, which are called *feasible solutions*, that satisfy all the necessary conditions. Then we can find the optimal solution among the feasible solutions using brute-force method, given the size of the feasible set is small.

For our objective function in Eq.(3), we introduce Lagrange multipliers as follows:

$$\Lambda(\mathbf{v}, \lambda, \mu) = \mathbf{v}^T \bar{L} \mathbf{v} - \lambda(\mathbf{v}^T \bar{Q} \mathbf{v} - \alpha) - \mu(\mathbf{v}^T \mathbf{v} - \text{vol}(\mathcal{G})). \quad (4)$$

Then according to the KKT Theorem, any feasible solution

to Eq.(3) must satisfy the following conditions:

$$\text{(Stationarity)} \quad \bar{L}\mathbf{v} - \lambda\bar{Q}\mathbf{v} - \mu\mathbf{v} = 0, \quad (5)$$

$$\text{(Primal feasibility)} \quad \mathbf{v}^T\bar{Q}\mathbf{v} \geq \alpha, \mathbf{v}^T\mathbf{v} = \text{vol}(\mathcal{G}), \quad (6)$$

$$\text{(Dual feasibility)} \quad \lambda \geq 0, \quad (7)$$

$$\text{(Complementary slackness)} \quad \lambda(\mathbf{v}^T\bar{Q}\mathbf{v} - \alpha) = 0. \quad (8)$$

Note that Eq.(5) comes from taking the derivative of Eq.(4) with respect to  $\mathbf{v}$ . Also note that we dismiss the constraint  $\mathbf{v} \neq D^{1/2}\mathbf{1}$  at this moment, because it can be checked independently, after we find the feasible solutions.

To solve Eq.(5)-(8), we start with looking at the complementary slackness requirement in Eq.(8), which can be broken down into two mutual-exclusive cases:

**Case 1:**  $\lambda = 0$ : In this case, the KKT conditions become:

$$\begin{aligned} \bar{L}\mathbf{v} - \mu\mathbf{v} = 0 &\Rightarrow \bar{L}\mathbf{v} = \mu\mathbf{v} \\ \mathbf{v}^T\bar{Q}\mathbf{v} \geq \alpha, \mathbf{v}^T\mathbf{v} = \text{vol}(\mathcal{G}). \end{aligned} \quad (9)$$

This case is easy to check because the feasible solutions generated in this case are still the eigenvectors of  $\bar{L}$ . All we need to do is to remove the ones that fail to satisfy the constraint  $\mathbf{v}^T\bar{Q}\mathbf{v} \geq \alpha$ .

**Case 2:**  $\lambda \neq 0$ : In this case, for Eq.(8) to hold  $\mathbf{v}^T\bar{Q}\mathbf{v} - \alpha$  must be 0. Consequently the KKT conditions become:

$$\bar{L}\mathbf{v} - \lambda\bar{Q}\mathbf{v} - \mu\mathbf{v} = 0, \quad (10)$$

$$\mathbf{v}^T\mathbf{v} = \text{vol}(\mathcal{G}), \quad (11)$$

$$\mathbf{v}^T\bar{Q}\mathbf{v} = \alpha, \quad (12)$$

$$\lambda > 0, \quad (13)$$

Unfortunately, under the assumption that  $\alpha$  is arbitrarily given by user and  $\lambda$  and  $\mu$  are independent variables, Eq.(10-13) cannot be solved explicitly, and it may produce infinite number of feasible solutions, if a solution exists.

Thus we introduce an additional variable,  $\beta$ , which is defined as the ratio between  $\mu$  and  $\lambda$ . Formally:

$$\beta = -\frac{\mu}{\lambda}\text{vol}(\mathcal{G}). \quad (14)$$

The introduction of  $\beta$  brings two computational benefits:

1. It helps convert our problem into a generalized eigenvalue system, which can be solved efficiently and produces up to  $N$  feasible solutions.
2. As we will show below,  $\beta$  always lower-bounds  $\alpha$ . Thus we can let user specify the value for  $\beta$ , and our original constraint will be satisfied automatically.

Now we substitute Eq.(14) into Eq.(10) we obtain:

$$\bar{L}\mathbf{v} - \lambda\bar{Q}\mathbf{v} + \frac{\lambda\beta}{\text{vol}(\mathcal{G})}\mathbf{v} = 0,$$

or equivalently:

$$\bar{L}\mathbf{v} = \lambda(\bar{Q} - \frac{\beta}{\text{vol}(\mathcal{G})}I)\mathbf{v} \quad (15)$$

We immediately notice that Eq.(15) is a generalized eigenvalue problem once  $\beta$  is given.

We denote  $\gamma = \mathbf{v}^T\bar{L}\mathbf{v}$ , by left-hand multiplying  $\mathbf{v}^T$  to both sides of Eq.(15) we have

$$\mathbf{v}^T\bar{L}\mathbf{v} = \lambda\mathbf{v}^T(\bar{Q} - \frac{\beta}{\text{vol}(\mathcal{G})}I)\mathbf{v}.$$

Then combining Eq.(11) and (12) we have

$$\gamma = \lambda(\alpha - \beta).$$

Now recall that  $L$  is positive semi-definite (Property 1), and so is  $\bar{L}$ , which means

$$\gamma = \mathbf{v}^T\bar{L}\mathbf{v} > 0, \forall \mathbf{v} \neq D^{1/2}\mathbf{1}.$$

Consequently, we have

$$\alpha - \beta = \frac{\gamma}{\lambda} > 0 \Rightarrow \alpha > \beta.$$

Recall that  $\alpha$  is the lower-bound of how well the given constraints are satisfied. And now we show that  $\beta$  is a lower-bound of  $\alpha$ . Therefore, instead of letting user assign the value of  $\alpha$  explicitly, we let user assign the value of  $\beta$ , and the output of algorithm will guarantee  $\mathbf{v}^T\bar{Q}\mathbf{v} = \alpha > \beta$ .

In summary, our method works as follows (the exact implementation is shown in Algorithm 1):

1. **Generating candidates:** The user specifies a value for  $\beta$ , and we solve the generalized eigenvalue system given in Eq.(15). Note that both  $\bar{L}$  and  $\bar{Q} - \beta/\text{vol}(\mathcal{G})I$  are Hermitian matrices, thus the generalized eigenvalues are guaranteed to be real numbers.
2. **Finding the feasible set:** Removing generalized eigenvectors associated with non-positive eigenvalues, and normalize the rest such that  $\mathbf{v}^T\mathbf{v} = \text{vol}(\mathcal{G})$ . Note that the trivial solution  $D^{1/2}\mathbf{1}$  is automatically removed in this step because if it is a generalized eigenvector in Eq.(15), the associated eigenvalue would be 0. Since we have at most  $N$  generalized eigenvectors, the number of feasible eigenvectors is at most  $N$ .
3. **Choosing the optimal solution:** We combine the feasible solutions from Case 1 and 2, and choose from them the one that minimizes  $\mathbf{v}^T\bar{L}\mathbf{v}$ , say  $\mathbf{v}^*$ .

Then in retrospect, we can claim that  $\mathbf{v}^*$  is the optimal solution to the objective function in Eq.(3) for  $\beta$  as given and  $\alpha = \mathbf{v}^{*T}\bar{Q}\mathbf{v}^*$ .

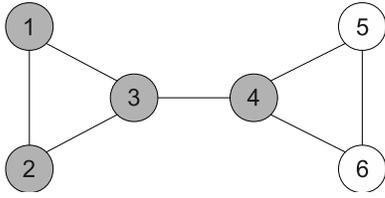
### 3.3 How To Set $\beta$ : Sufficient Condition for the Existence of Feasible Solutions

On one hand, our method described above is guaranteed to generate a finite number of feasible solutions. On the other hand, we need to set  $\beta$  appropriately so that the generalized eigenvalue system in Eq.(15) combined with the KKT conditions in Eq.(10-13) will give rise to at least one feasible solution. In this section, we discuss such a sufficient condition:

$$\beta < \lambda_{max}\text{vol}(\mathcal{G}),$$

where  $\lambda_{max}$  is the largest eigenvalue of  $\bar{Q}$ .

In this case, the matrix on the right hand side of Eq.(15),  $\bar{Q} - \beta/\text{vol}(\mathcal{G})I$ , will have at least one positive eigenvalue. Consequently, the generalized eigenvalue system in Eq.(15) will have at least one positive eigenvalue. Moreover, the number of feasible eigenvectors will increase if we make  $\beta$  smaller. For example, if we set  $\beta < \lambda_{min}\text{vol}(\mathcal{G})$ ,  $\lambda_{min}$  to be the smallest eigenvalue of  $\bar{Q}$ , then  $\bar{Q} - \beta/\text{vol}(\mathcal{G})I$  becomes positive definite. Then the generalized eigenvalue system in Eq.(15) will generate  $N - 1$  feasible eigenvectors (the trivial solution  $D^{1/2}\mathbf{1}$  with eigenvalue 0 is dropped).



**Figure 1: An illustrative example: the affinity structure says  $\{1, 2, 3\}$  and  $\{4, 5, 6\}$  while the node labeling (coloring) says  $\{1, 2, 3, 4\}$  and  $\{5, 6\}$ .**

In practice, we normally choose the value of  $\beta$  within the range

$$(\lambda_{\min} \text{vol}(\mathcal{G}), \lambda_{\max} \text{vol}(\mathcal{G})).$$

In that range, the greater  $\beta$  is, the more the solution will be biased towards satisfying the constraints in  $Q$ . Again, note that whenever we have  $\beta < \lambda_{\max} \text{vol}(\mathcal{G})$ , the value of  $\alpha$  will always be bounded by

$$\beta < \alpha \leq \lambda_{\max} \text{vol}(\mathcal{G}).$$

Therefore we do not need to take care of  $\alpha$  explicitly.

### 3.4 Generating Constraints from Labels

In practice, the pairwise constraints are often generated from known data labels, i.e. a ML is added when two instances have the same label and a CL is added when two instances have different labels. Similarly, the constraint matrix  $Q$  in our formulation can be conveniently generated from a (partially) labeled data set. Let  $X$  be an  $N \times K$  matrix, where  $X_{ij}$  is positive if we believe data instance  $i$  belongs to class  $j$ ; and negative otherwise; the magnitude of  $X_{ij}$  indicates the degree of that belief. Then  $Q$  can be generated by

$$Q = XX^T. \quad (16)$$

Note that our model’s capacity of incorporating real-valued constraints makes it possible to better handle multi-labeled data set: if node  $i$  and  $j$  share two labels whereas node  $k$  and  $l$  only share one label, then our formulation is able to generate constraints in such a way that node  $i$  and  $j$  are *more strongly* advised to be assigned into the same cluster than node  $k$  and  $l$  are.

### 3.5 A Graph Cut Interpretation

Unconstrained spectral clustering can be interpreted as finding the N-CUT of an *unlabeled* graph. Similarly, our formulation of constrained spectral clustering in Eq.(3) can also be interpreted as finding the N-CUT in a *labeled/colored* graph.

Specifically, suppose we have an undirected, weighted graph. The nodes of the graph are colored in such a way that nodes of the same color are advised to be assigned into the same cluster while nodes of different colors are advised to be assigned into different clusters (e.g. Fig. 1). Let  $\mathbf{v}^*$  be the solution to the constrained optimization problem in Eq.(3). We cut the graph into two parts based on the values of the entries of  $\mathbf{u}^* = D^{-1/2} \mathbf{v}^*$ . Then  $\mathbf{v}^{*T} \bar{L} \mathbf{v}^*$  can be interpreted as the **cost** of the cut (in a relaxed sense), which is to minimize. On the other hand,

$$\alpha = \mathbf{v}^{*T} \bar{Q} \mathbf{v}^* = \mathbf{u}^{*T} Q \mathbf{u}^*$$

can be interpreted as the **purity** of the cut (also in a relaxed sense), according to the color of the nodes in respective sides. For example, if  $Q_{ij}$  is a positive number, then  $\mathbf{u}_i^*$  and  $\mathbf{u}_j^*$  having the same sign will help increase the purity of the cut, whereas their having different signs will decrease the purity of the cut. It is not difficult to see that the purity can be maximized when there is no pair of nodes with different colors that are assigned to the same side of the cut, which is the case where constraints in  $Q$  are completely satisfied.

### 3.6 An Illustrative Example

To illustrate how our approach works, we present a toy example as follows. In Fig. 1, we have a graph associated with the following affinity matrix:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Unconstrained spectral clustering will cut the graph at edge (3,4) and split it into two symmetric parts  $\{1, 2, 3\}$  and  $\{4, 5, 6\}$  (Fig. 2(a)). Now we introduce constraints as follows:

$$Q = \begin{bmatrix} +1 & +1 & +1 & +1 & -1 & -1 \\ +1 & +1 & +1 & +1 & -1 & -1 \\ +1 & +1 & +1 & +1 & -1 & -1 \\ +1 & +1 & +1 & +1 & -1 & -1 \\ -1 & -1 & -1 & -1 & +1 & +1 \\ -1 & -1 & -1 & -1 & +1 & +1 \end{bmatrix}.$$

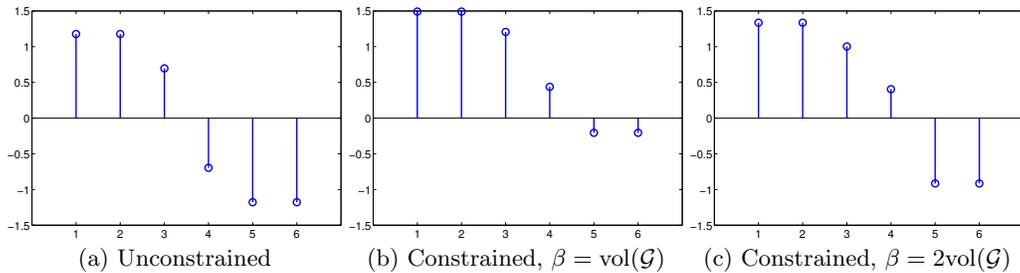
$Q$  is essentially saying that we want to group nodes  $\{1, 2, 3, 4\}$  into one cluster and  $\{5, 6\}$  the other. Although this kind of “full supervision” does not make sense in practice, it is used here just to make the result more obvious and intuitive.

$\bar{Q}$  has two distinct eigenvalues: 0 and 2.6667. As analyzed above,  $\beta$  must be smaller than  $2.6667 \times \text{vol}(\mathcal{G})$  to guarantee the existence of a feasible solution, and larger  $\beta$  means we want more constraints in  $Q$  to be satisfied (in a relaxed sense). Thus we set  $\beta$  to  $\text{vol}(\mathcal{G})$  and  $2\text{vol}(\mathcal{G})$  respectively, and see how the results will be affected by different values of  $\beta$ . We solve the generalized eigenvalue system in Eq.(15), and plot the resultant cluster indicator vector  $\mathbf{u}^*$  in Fig. 2(b) and 2(c). We can see that as  $\beta$  increases, node 4 is dragged from the group of nodes  $\{5, 6\}$  to the group of nodes  $\{1, 2, 3\}$ , which conforms to our expectation that greater  $\beta$  value implies higher level of constraint satisfaction.

## 4. ALGORITHM

In this section, we discuss the implementation issues of our method.

The routine of our method is similar to that of unconstrained spectral clustering. The input of the algorithm is an affinity matrix  $A$ , the constraint matrix  $Q$  (or alternatively the label matrix  $X$ ), and the threshold  $\beta$ . Then we solve the generalized eigenvalue problem in Eq.(15) and find all the feasible generalized eigenvectors. The output is the optimal (relaxed) cluster assignment indicator  $\mathbf{u}^*$ . The algorithm is summarized in Algorithm 1. Note that it only considers the solutions generated from Case 2 in Section 3.2. Those from Case 1 (the trivial case) can be examined separately.



**Figure 2: The solutions to the illustrative example in Fig. 1 with different  $\beta$ . The x-axis is the indices of the instances and the y-axis is the corresponding entry values in the optimal (relaxed) cluster indicator  $\mathbf{u}^*$ . Notice that node 4 is biased toward nodes  $\{1, 2, 3\}$  as  $\beta$  increases.**

---

**Algorithm 1: Constrained Spectral Clustering**

---

**Input:** Affinity matrix  $A$ , constraint matrix  $Q$ ,  $\beta$ ;  
**Output:** The optimal (relaxed) cluster indicator  $\mathbf{u}^*$ ;

- 1  $\text{vol}(\mathcal{G}) \leftarrow \sum_{i=1}^N \sum_{j=1}^N A_{ij}$ ,  $D \leftarrow \text{diag}(\sum_{j=1}^N A_{ij})$ ;
- 2  $\bar{L} \leftarrow I - D^{-1/2} A D^{-1/2}$ ,  $\bar{Q} \leftarrow D^{-1/2} Q D^{-1/2}$ ;
- 3  $\lambda_{max} \leftarrow$  the largest eigenvalue of  $\bar{Q}$ ;
- 4 **if**  $\beta \geq \lambda_{max} \text{vol}(\mathcal{G})$  **then**
- 5     **return**  $\mathbf{u}^* = \emptyset$ ;
- 6 **end**
- 7 **else**
- 8     Solve the generalized eigenvalue system in Eq.(15);
- 9     Remove eigenvectors associated with non-positive eigenvalues and normalize the rest by  $\mathbf{v} \leftarrow \frac{\mathbf{v}}{\|\mathbf{v}\|} \text{vol}(\mathcal{G})$ ;
- 10     $\mathbf{v}^* \leftarrow \arg \min_{\mathbf{v}} \mathbf{v}^T L \mathbf{v}$ , where  $\mathbf{v}$  is among the feasible eigenvectors generated in the previous step;
- 11    **return**  $\mathbf{u}^* \leftarrow D^{-1/2} \mathbf{v}^*$ ;
- 12 **end**

---

Our algorithm can be naturally extended to  $K$ -way partition for  $K > 2$ , following what we usually do for unconstrained spectral clustering [15]: instead of using the optimal generalized eigenvector  $\mathbf{u}^*$ , we preserve top- $K$  generalized eigenvectors corresponding to positive generalized eigenvalues, and perform  $K$ -means algorithm in the eigenspace.

Since our model encodes constraints as a degree of belief, inconsistent constraints in  $Q$  will not corrupt our algorithm. Instead, they are enforced implicitly by the effort of improving  $\mathbf{u}^T Q \mathbf{u}$ . Note that if the constraint matrix  $Q$  is generated from the partial label matrix  $X$ , then the constraints in  $Q$  will always be consistent.

The runtime of our algorithm is dominated by that of the generalized eigendecomposition. In other words, the complexity of our algorithm is on a par with that of unconstrained spectral clustering in big-O notation, which is  $\mathcal{O}(kN^2)$ ,  $N$  to be the number of data instances and  $k$  to be the number of eigenpairs we need to compute.

## 5. EMPIRICAL STUDY

In this section, we applied our method on several real-world data sets, with comparison to existing techniques. Our goal is to answer the following questions:

1. Does our method generate semantically meaningful re-

sults that conform to user supervision and expectation?

2. Does our method produce entirely novel clustering when given different sets of constraints?
3. Does our method outperform unconstrained spectral clustering and existing constrained spectral clustering techniques?
4. Can our method effectively utilize degree-of-belief constraints, which may carry information that binary constraints cannot encode?

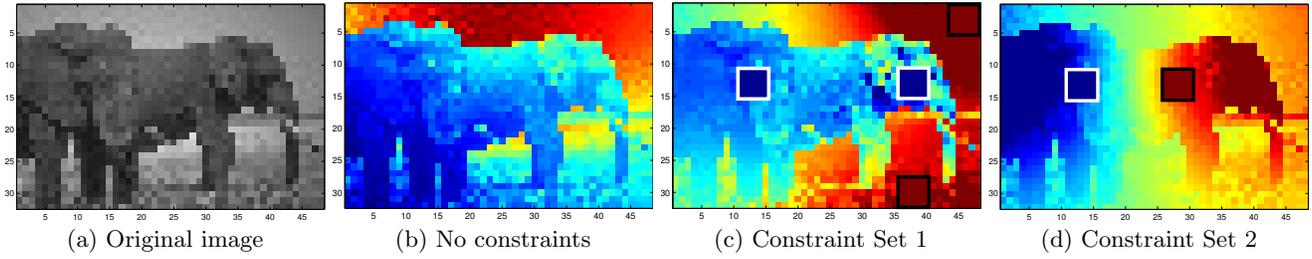
All data sets and source codes (in Matlab) used in the experiments are publicly available. Please contact the first author for further information.

### 5.1 Constrained Image Segmentation

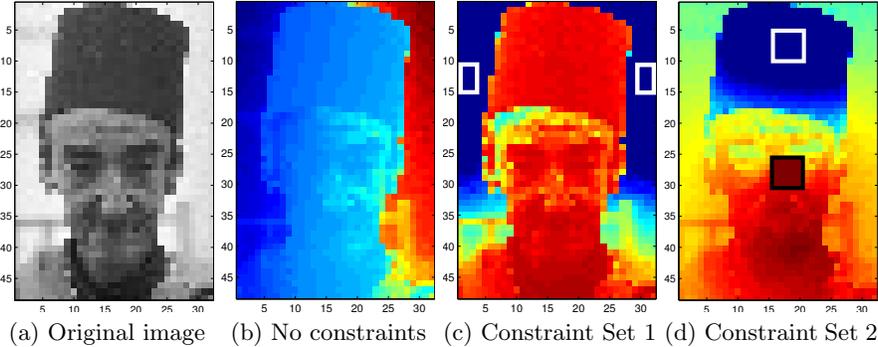
First we validate the effectiveness of our approach in the context of image segmentation. We choose image segmentation as a demonstration for several reasons: 1) it is one of the applications where spectral clustering significantly outperforms other clustering techniques, e.g.  $K$ -means; 2) the results of image segmentation can be easily interpreted and evaluated by human; 3) instead of generating random constraints, we can add in semantically meaningful constraints to see if the results of constrained clustering conform to our expectation.

The images we used were chosen from the *Berkeley Segmentation Dataset and Benchmark* [12]. The original images are  $480 \times 320$  grayscale images in jpeg format. For efficiency consideration, we compressed them to 10% of the original size, which is  $48 \times 32$  pixels, as shown in Fig. 3(a) and 4(a). Then affinity matrix of the image was computed using RBF kernel, based on both the positions and the grayscale values of the pixels. As a baseline, we used unconstrained spectral clustering [14] to generate a 2-segmentation of the image. Then we introduced different sets of constraints to see if they can generate expected segmentation. Note that the results of image segmentation vary with the number of segments. To save us from the complications of parameter tuning, which is related to the contribution of this work, we always set the number of segments to be 2.

The results are shown in Fig. 3 and 4. Note that to visualize the resultant segmentation, we reconstructed the image using the entry values in the relaxed cluster indicator vector  $\mathbf{u}^*$ . In Fig. 3(b), the unconstrained spectral clustering partitioned the elephant image into two parts: the sky (red



**Figure 3: Segmentation results of the elephant image (best viewed in color).** The images are reconstructed based on the relaxed cluster indicator  $u^*$ . Pixels that are closer to the red end of the spectrum belong to one segment and blue the other. The labeled pixels are as bounded by the black and white rectangles.



**Figure 4: Segmentation results of the face image (best viewed in color).** The images are reconstructed based on the relaxed cluster indicator  $u^*$ . Pixels that are closer to the red end of the spectrum belong to one segment and blue the other. The labeled pixels are as bounded by the black and white rectangles.

pixels) and the two elephants and the ground (blue pixels). This is not satisfying in the sense that it failed to isolate the elephants from the background (the sky and the ground). To correct this, we introduced constraints by labeling two  $5 \times 5$  blocks to be 1 (as bounded by the black rectangles in Fig. 3(c)): one at the upper-right corner of the image (the sky) and the other at the lower-right corner (the ground); we also labeled two  $5 \times 5$  blocks on the heads of the two elephants to be  $-1$  (as bounded by the white rectangles in Fig. 3(c)). Then we used Eq.(16) to generate the constraint matrix  $Q$ : a ML was added between every pair of pixels with the same label and a CL was added between every pair of pixels with different labels. The parameter  $\beta$  was set as:

$$\beta = \lambda_{max} \times \text{vol}(\mathcal{G}) \times \left(0.5 + 0.4 \times \frac{\# \text{ of constraints}}{N^2}\right), \quad (17)$$

where  $\lambda_{max}$  is the maximum eigenvalue of  $\bar{Q}$ . In this way,  $\beta$  is always between  $0.5\lambda_{max}\text{vol}(\mathcal{G})$  and  $0.9\lambda_{max}\text{vol}(\mathcal{G})$ , and it will gradually increase as the number of constraints increases. From Fig. 3(c) we can see that with the help of user supervision, our method successfully isolated the two elephants (blue) from the background, which is the sky and the ground (red). Note that our method achieved this with very simple labeling: four squared blocks.

To show the flexibility of our method, we tried a different set of constraints on the same elephant image with the same parameter settings. This time we aimed to separate the two elephants from each other, which is impossible in the unconstrained case because the two elephants are not only similar in color (grayscale value) but also adjacent in space.

Again we used two  $5 \times 5$  blocks (as bounded by the black and white rectangles in Fig. 3(d)), one on the head of the elephant on the left, labeled to be 1, and the other on the body of the elephant on the right, labeled to be  $-1$ . As shown in Fig. 3(d), our method cut the image into two parts with one elephant on the left (blue) and the other on the right (red), just like what a human user would do.

Similarly, we applied our method on a human face image as shown in Fig. 4(a). The unconstrained spectral clustering failed to isolate the human face from the background (Fig. 4(b)). This is because the tall hat breaks the spatial continuity between the left side of the background and the right side. Then we labeled two  $5 \times 3$  blocks to be in the same class, one on each side of the background. As we intended, our method assigned the background of both sides into the same cluster and thus isolated the human face with his tall hat from the background (Fig. 4(c)). Again, this was achieved simply by labeling two blocks in the image, which covered about 3% of all pixels. Alternatively, if we labeled a  $5 \times 5$  block in the hat to be 1, and a  $5 \times 5$  block in the face to be  $-1$ , the resultant clustering will isolate the hat from the rest of the image (Fig. 4(d)).

## 5.2 Clustering UCI Benchmarks

Next we evaluated our method (CSP) by clustering benchmark data sets from the UCI Archive [1]. We chose six different data sets with class label information, namely Hepatitis, Iris, Wine, Glass, Ionosphere and Breast Cancer Wisconsin (Diagnostic). We performed 2-way clustering simply by partitioning the optimal cluster indicator according to sign:

**Table 2: The UCI benchmarks**

Identifier	#Instances	#Attributes
Hepatitis	80	19
Iris	100	4
Wine	119	13
Glass	214	9
Ionosphere	351	34
WDBC	569	30

**Table 3: The newsgroup data**

Group	Label	#Instances
3	comp.os.ms-windows.misc	53
4	comp.sys.ibm.pc.hardware	60
5	comp.sys.mac.hardware	59
9	rec.motorcycles	65
10	rec.sport.baseball	64
11	rec.sport.hockey	51

positive entries to one cluster and negative the other. We removed the SETOSA class from the Iris data set, which is the class that is known to be well-separated from the other two. For the same reason we removed Class 1 from the Wine data set, which is well-separated from the other two. We also removed data instances with missing values. The statistics of the data sets after preprocessing are listed in Table 2.

For each data set, we computed the affinity matrix using the RBF kernel. We randomly generated constraints using the groundtruth label information. For each round we randomly chose a certain percentage of data instances and assumed that their labels are known; then we generated the constraint matrix  $Q$  following Eq.(16). Note that when no constraint was generated, the unconstrained spectral clustering was performed. The quality of the clustering results was measured by Rand index [16], which tells how similar our clustering is as compared to the groundtruth class labels. The only parameter in our method,  $\beta$ , was set according to Eq.(17) throughout the experiments.

We compared our method to two existing techniques. The first one (**ModAff**) is from [8], which modifies the affinity matrix directly: when a ML constraint is given, it changes the corresponding entries in  $A$  to 1; and 0 for CL constraint. Then unconstrained spectral clustering is performed on the new graph Laplacian. The second one (**GrBias**) is from [20,21], which encodes partial grouping information as a projection matrix. Note that the **GrBias** method can only accommodate ML constraints. We also implemented the subspace trick in [5] and the affinity propagation algorithm in [11]. We did not present results from those two techniques because overall they did not perform as well as the two we presented.

We report the Rand index of all methods against the percentage of known labels (from 0 to 100%, by 10% increment) so that we can see how the quality of clustering varies when more constraints are added in. Note that at each step, we randomly generate 100 sets of constraints and reported the mean, maximum and minimum Rand index of the 100 random trials, as shown in Fig. 5. From the results we can tell:

- Our method consistently and significantly outperforms unconstrained spectral clustering (whose Rand index is

the one reported at  $x = 0$ ). We can also notice the performance boost from  $x = 0$  to  $x = 10\%$ , which means that our method can effectively improve the clustering result with a small number of constraints.

- Our method outperforms the two competitors in most cases. And more importantly, the performance of our method is much more stable over different data sets, different number of constraints, and different sets of randomly generated constraints. As a contrast, the **GrBias** produced good results on certain data sets with certain configurations, but bad results in other cases. One possible reason, among many others, might be that since it only uses ML constraints, it will not help much in cases where CL constraints are necessary to improve the results.
- Unlike the **GrBias** method, the performance of our method increases consistently with the number of given constraints, which indicates that our method can effectively use constraints to improve the clustering results. (Similar observation was made in [8] that the performance of **GrBias** may drop when more constraints are provided.)

### 5.3 Results with Degree-of-Belief Constraints

Lastly, we show that our method can effectively incorporate degree-of-belief constraints, which may carry richer information that binary constraints cannot accommodate. To make our case, we encode hierarchical (multiple) labels into degree-of-belief constraints, and see if our method can recover the hierarchical structure of the data set (only) based on given constraints. We chose a subset of the 20 Newsgroup data set<sup>1</sup>, as shown in Table 3. We randomly sampled about 350 documents from 6 groups. At the highest level, those groups belong to two topics: computer (comp) and recreation (rec). We used these two topics as groundtruth labels when computing Rand index.

To generate binary constraints, we added a ML link between two articles within the same group; CL between two articles from different groups. To generate degree-of-belief constraints, we converted the group titles into multiple labels, e.g. rec.sport.baseball became rec, sport, and baseball. Then given two articles, we compared the number of common labels they shared. For example, if those two articles came from the same group, we set the corresponding entry in  $Q$  to +3; if one was from rec.sport.baseball and the other from rec.sport.hockey, we set the corresponding entry to +2; if they did not share any label at all, we set the entry to -1 (recall that in our model, 0 does not mean CL but no constraints given).

After the constraints were generated, we performed our method with degree-of-belief (**CSP-DoB**) and binary constraints (**CSP-Binary**), respectively. Our task was to cluster the data set into 6 clusters, and then compute the Rand index with respect to the two-topic groundtruth (comp and rec). As shown in Fig. 6, since the binary constraints only captured the group-level relationship, the top-level class information (comp vs. rec) cannot be recovered by using binary constraints. As a contrast, the degree-of-belief constraints preserved the hierarchical structure of the data set. Hence, although the number of clusters were assigned to be 6, our

<sup>1</sup><http://people.csail.mit.edu/jrennie/20Newsgroups/>

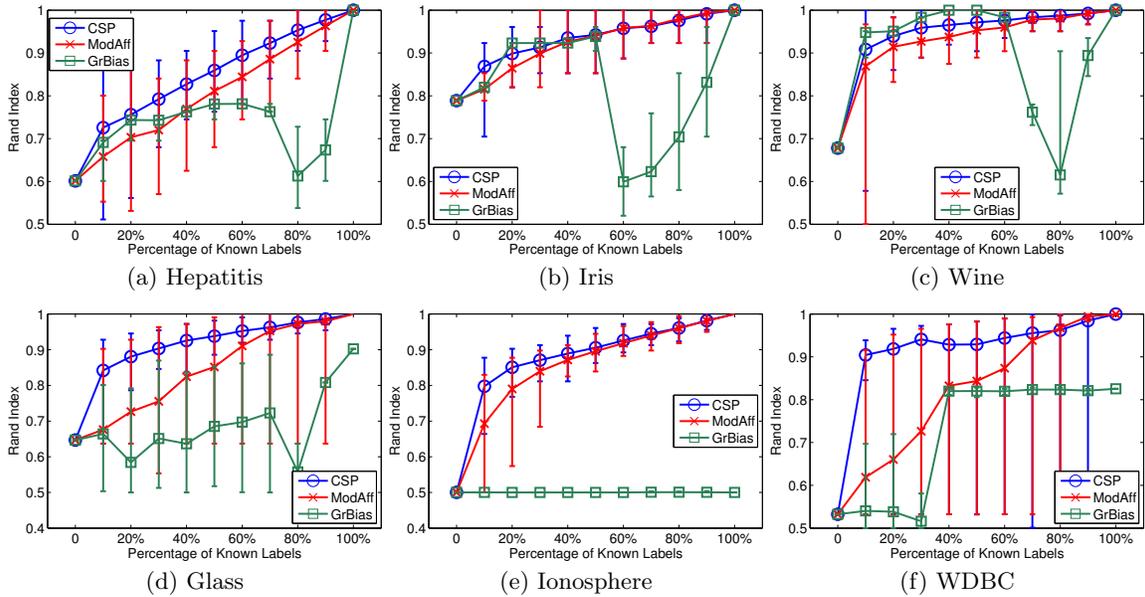


Figure 5: Clustering results on UCI benchmarks. The Rand index over different percentage of known labels is reported (mean, maximum and minimum over 100 random constraint sets).

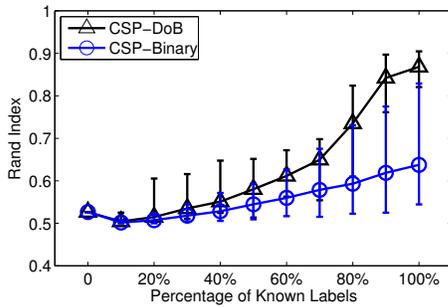


Figure 6: Clustering results on the newsgroup data set, with binary and degree-of-belief constraints.

method with degree-of-belief constraints tended to merge the articles into 2 clusters: one corresponding to comp and the other rec. The resultant clustering successfully recovered the original hierarchical information, as indicated by the steadily increasing Rand index (CSP-DoB) in Fig. 6.

## 6. CONCLUSION

This paper addresses the problem of constrained spectral clustering. While constrained  $K$ -means clustering has been well-studied, existing techniques on constrained spectral clustering are limited because they are primarily focused on MUST-LINK and CANNOT-LINK constraints, which could be both insufficient and inflexible in practice. To overcome this, we propose a generalized framework for constrained spectral clustering. Our approach is more flexible in the sense that we can deal with both binary constraints and real-valued degree-of-belief constraints. Our approach is also more principled since it can be considered a natural extension to the original objective function of unconstrained spectral clustering, as a cut on a labeled graph. Our objective function is formulated as a constrained optimization prob-

lem and can be solved in closed form with polynomial time, through generalized eigenvalue decomposition. Our method introduces a user-specified parameter  $\beta$ , which serves as a tradeoff factor between the structure as defined by the original graph Laplacian and that by the constraint matrix.

Empirical results justified the effectiveness of our method. We used image segmentation to demonstrate that our method can produce meaningful and intuitive clustering with various sets of constraints. We also evaluated our method on benchmark data sets. We showed that our method can improve the quality of clustering by taking in both binary and real-valued constraints over unconstrained spectral clustering. We also showed the advantage of our approach over existing techniques.

## 7. ACKNOWLEDGMENTS

The authors gratefully acknowledge support of this research from the NSF (IIS-0801528) and ONR (N000140910712 P00001).

## 8. REFERENCES

- [1] A. Asuncion and D. Newman. UCI machine learning repository, 2007.
- [2] S. Basu, I. Davidson, and K. Wagstaff, editors. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 2008.
- [3] T. Coleman, J. Saunderson, and A. Wirth. Spectral clustering with inconsistent advice. In *ICML*, pages 152–159, 2008.
- [4] I. Davidson and S. S. Ravi. Intractability and clustering with constraints. In *ICML*, pages 201–208, 2007.
- [5] T. De Bie, J. A. K. Suykens, and B. De Moor. Learning from general label constraints. In *SSPR/SPR*, pages 671–679, 2004.

- [6] P. Drineas, A. M. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33, 2004.
- [7] X. Ji and W. Xu. Document clustering with prior knowledge. In *SIGIR*, pages 405–412, 2006.
- [8] S. D. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *IJCAI*, pages 561–566, 2003.
- [9] H. Kuhn and A. Tucker. Nonlinear programming. *ACM SIGMAP Bulletin*, pages 6–18, 1982.
- [10] Z. Li, J. Liu, and X. Tang. Constrained clustering via spectral regularization. In *CVPR*, pages 421–428, 2009.
- [11] Z. Lu and M. Á. Carreira-Perpiñán. Constrained spectral clustering through affinity propagation. In *CVPR*, 2008.
- [12] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int’l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
- [13] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.
- [14] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.
- [15] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.
- [16] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *ICML*, pages 1103–1110, 2000.
- [17] F. Wang, C. H. Q. Ding, and T. Li. Integrated KL (K-means - Laplacian) clustering: A new clustering approach by combining attribute data and pairwise relations. In *SDM*, pages 38–48, 2009.
- [18] S. White and P. Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, 2005.
- [19] Q. Xu, M. desJardins, and K. Wagstaff. Constrained spectral clustering under a local proximity structure assumption. In *FLAIRS Conference*, pages 866–867, 2005.
- [20] S. X. Yu and J. Shi. Grouping with bias. In *NIPS*, pages 1327–1334, 2001.
- [21] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):173–183, 2004.