

# On Constrained Spectral Clustering and Its Applications

Xiang Wang · Buyue Qian · Ian Davidson

Received: date / Accepted: date

**Abstract** Constrained clustering has been well-studied for algorithms such as  $K$ -means and hierarchical clustering. However, how to satisfy many constraints in these algorithmic settings has been shown to be intractable. One alternative to encode many constraints is to use spectral clustering, which 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 underlying eigenspace, we present a more natural and principled formulation, which explicitly encodes the constraints as part of a constrained optimization problem. Our method offers several practical advantages: it can encode the degree of belief 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 eigen-decomposition. Furthermore, by inheriting the objective function from spectral clustering and encoding the constraints explicitly, much of the existing analysis of unconstrained spectral clustering techniques remains valid for our formulation. We validate the effectiveness of our approach by empirical results on real-world datasets. We also demonstrate an innovative use of encoding large number of constraints: transfer learning via constraints.

**Keywords** Spectral clustering · Constrained clustering · Active learning · Graph partition

---

X. Wang

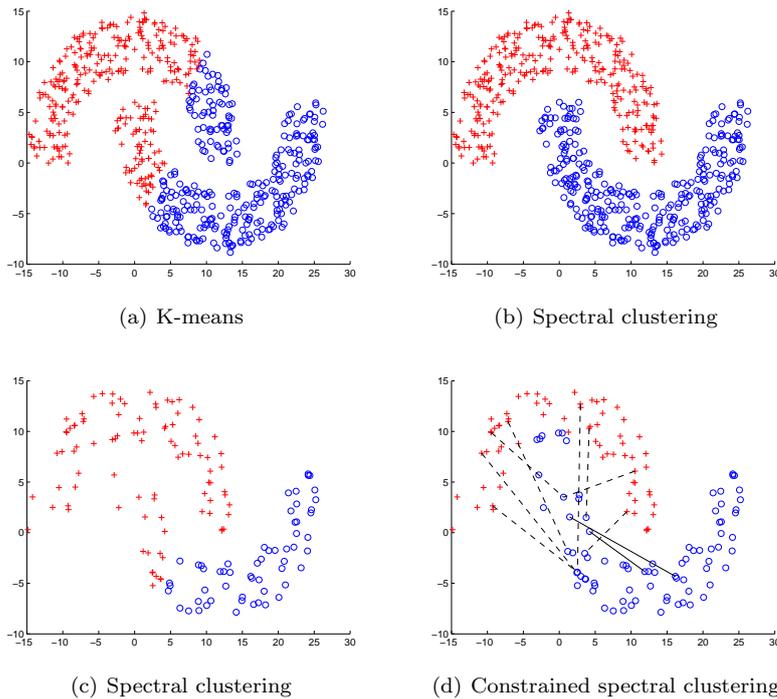
Department of Computer Science, University of California, Davis. Davis, CA 95616, USA.  
E-mail: xiang@ucdavis.edu

B. Qian

Department of Computer Science, University of California, Davis. Davis, CA 95616, USA.  
E-mail: byqian@ucdavis.edu

I. Davidson

Department of Computer Science, University of California, Davis. Davis, CA 95616, USA.  
E-mail: davidson@cs.ucdavis.edu



**Fig. 1** A motivating example for constrained spectral clustering.

## 1 Introduction

### 1.1 Background and Motivation

Spectral clustering is an important clustering technique that has been extensively studied in the image processing, data mining, and machine learning communities Shi and Malik (2000); von Luxburg (2007); Ng et al (2001). It is considered superior to traditional clustering algorithms like  $K$ -means in terms of having deterministic polynomial-time solution, the ability to model arbitrary shaped clusters, and its equivalence to certain graph cut problems. For example, spectral clustering is able to capture the underlying moon-shaped clusters as shown in Fig. 1(b), whereas  $K$ -means would fail (Fig. 1(a)). The advantage of spectral clustering has also been validated by many real-world applications, such as image segmentation Shi and Malik (2000) and mining social networks White and Smyth (2005).

Spectral clustering was originally proposed to address an unsupervised learning problem: the data instances are unlabeled, and all available information is encoded in the graph Laplacian. However, there are cases where spectral clustering in its unsupervised form becomes insufficient. Using the same toy data, as shown in (Fig. 1(c)), when the two moons are under-sampled, the clusters become so sparse that the separation of them becomes difficult. To help spectral clustering recover from an undesirable partition, we can introduce small to extensive amounts of side information in various forms. For example:

1. **Pairwise constraints:** Domain experts may explicitly assign constraints that state a pair of instances must be in the same cluster (Must-Link, ML for short) or that a pair of instance cannot be in the same cluster (Cannot-Link, CL for short). For instance, as shown in Fig. 1(d), we assigned several ML (solid lines) as well as CL (dashed lines) constraints, and applied the constrained spectral clustering algorithm, which we will describe later. As a result, the two moons are successfully recovered.
2. **Partial labeling:** There might be labels on some of the instances, which could be neither complete nor exhaustive. We demonstrate in Fig. 9 that even small amounts of labeled information can greatly improve clustering results when compared against the ground truth partition, as inferred by the labels.
3. **Additional Weak Distance Metrics:** In some situations there may be more than one distance metrics available. For example, in Table 3 and accompanying paragraphs we describe clustering documents using distance functions based on different languages (features).
4. **Transfer of knowledge:** In the context of transfer learning Pan and Yang (2010), if we treat the graph Laplacian as the target domain, we could transfer knowledge from a different but related graph, which serves as the source domain. We discuss this direction in Section 6.3 and 7.5.

Note that all the side information above can be transformed into pairwise ML and CL constraints, which could either be hard (binary) or soft (degree of belief). For example, if the side information comes from a source graph, we can construct pairwise constraints by assuming that the more similar two instance are in the source graph, the more likely they belong to the same cluster in the target graph. Consequently the constraints should naturally be represented by a degree of belief, rather than a binary assertion.

How to make use of these side information to improve clustering falls into the area of constrained clustering Basu et al (2008). In general, constrained clustering is a category of techniques that try to incorporate ML and CL constraints into existing clustering schemes. It has been well studied on algorithms such as  $K$ -means clustering, mixture model, hierarchical clustering, and density-based clustering. Previous studies showed that satisfying all constraints at once Davidson and Ravi (2007a), incrementally Davidson et al (2007), or even pruning constraints Davidson and Ravi (2007b) is intractable. Furthermore, it was shown that algorithms that build set partitions incrementally (such as  $K$ -means and EM) are prone to being over-constrained Davidson and Ravi (2006). As a contrast, incorporating constraints into spectral clustering is a promising direction since, unlike aforementioned algorithms, all points are assigned simultaneously to clusters, even if the given constraints are inconsistent.

Constrained spectral clustering is still a developing area. Previous work on constrained spectral clustering can be divided into two categories, based on how they enforce the constraints. The first category Kamvar et al (2003); Xu et al (2005); Lu and Carreira-Perpiñán (2008); Wang et al (2009); Ji and Xu (2006) 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 category use constraints to restrict the feasible solution space De Bie et al (2004); Coleman et al (2008); Li et al (2009); Yu and Shi (2001, 2004). Existing methods in both categories share several limitations:

1. They are designed to handle only binary constraints. However, as we have stated above, in many real-world applications, constraints are made available in the form of real-valued degree of belief, rather than a yes or no assertion.
2. They aim to satisfy as many constraints as possible, which could lead to inflexibility in practice. For example, the given set of constraints could be noisy, and satisfying some of the constraints could actually hurt the overall performance. Also, it is reasonable to ignore a small portion of constraints in exchange for a clustering with much lower cost.
3. They do not offer a natural interpretation to either the way that constraints are encoded or the implication of enforcing them.

## 1.2 Our Contributions

In this paper, we study how to incorporate **large** amounts of pairwise constraints into spectral clustering, in a flexible manner that addresses the limitations of previous work. Then we go onto show the practical benefits of our approach, including new applications previously not possible.

We extend beyond binary ML/CL constraints and propose a more flexible framework to accommodate general-type side information. We allow the binary constraints to be relaxed to real-valued degree of belief that two data instances belong to the same cluster or two different clusters. 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 must be satisfied. Therefore, **our method provides maximum flexibility in terms of both representing constraints and satisfying them**. This in addition to handling large amounts of constraints allows the encoding of new styles of information such as entire graphs and alternative distance metrics in their raw form without considering issues such as constraint inconsistencies and over-constraining.

Our contributions are:

- We propose a principled framework that formulates constrained spectral clustering in a generalized form, which can incorporate large amounts of both hard and soft constraints.
- We show how to enforce constraints in a flexible way: a user-specified threshold is introduced so that a limited amount of constraints can be ignored in exchange for lower clustering cost. This allows incorporating side information in its raw form without considering issues such as inconsistency and over-constraining.
- We extend the objective function of unconstrained spectral clustering by encoding constraints explicitly and creating a novel constrained optimization problem. Thus our formulation naturally covers unconstrained spectral clustering as a special case.
- We show that our objective function can be turned into a generalized eigenvalue problem, which can be solved deterministically in polynomial time. This is a major advantage over constrained  $K$ -means clustering, which produces non-deterministic solutions while being intractable even for  $K = 2$  Drineas et al (2004); Davidson and Ravi (2007b).
- We interpret our formulation from both the graph cut perspective and the Laplacian embedding perspective.

- We validate the effectiveness of our approach and its advantage over existing methods using standard benchmarks and new innovative applications such as transfer learning.

This paper is an extension to our previous work Wang and Davidson (2010) with the following additions: 1) we extended our algorithm from 2-way partition to  $K$ -way partition (Section 6.2); 2) we newly added geometric interpretation to our algorithm (Section 5.2); 3) we showed how to apply our algorithm to a novel application (Section 6.3), transfer learning, and tested it with real-world datasets (Section 7.5); 4) we presented a much more comprehensive experiment section with more tasks conducted on more datasets (Section 7.2 and 7.4).

The rest of the paper is organized as follows: In Section 2 we briefly survey previous work on constrained spectral clustering; Section 3 provides preliminaries for spectral clustering; in Section 4 we formally introduce our formulation for constrained spectral clustering and show how to solve it efficiently; in Section 5 we interpret our objective from two different perspectives; in Section 6 we discuss the implementation of our algorithm and possible extensions; we empirically evaluate our approach in Section 7; Section 8 concludes the paper.

## 2 Related Work

Constrained clustering is a category of methods that extend clustering from unsupervised setting to semi-supervised setting, where side information is available in the form of, or can be converted into, pairwise constraints. A number of algorithms have been proposed on how to incorporate constraints into spectral clustering, which can be grouped into two categories.

The first category manipulates the graph Laplacian directly. Kamvar et al. proposed the spectral learning algorithm Kamvar et al (2003) that sets the  $(i, j)$ -th entry of the affinity matrix to 1 if there is a ML between node  $i$  and  $j$ ; 0 for CL. A new graph Laplacian is then computed based on the modified affinity matrix. In Xu et al (2005), the constraints are encoded in the same way, but a random walk matrix is used instead of the normalized Laplacian. Kulis et al. Kulis et al (2005) proposed to add both positive (for ML) and negative (for CL) penalties to the affinity matrix (they then used kernel  $K$ -means, instead of spectral clustering, to find the partition based on the new kernel). Lu and Carreira-Perpiñán Lu and Carreira-Perpiñán (2008) proposed to propagate the constraints in the affinity matrix. In Ji and Xu (2006); Wang et al (2009), the graph Laplacian is modified by combining the constraint matrix as a regularizer. The limitation of these approaches is that there is no principled way to decide the weights of the constraints, and there is no guarantee that how well the give constraints will be satisfied.

The second category manipulates the eigenspace directly. For example, the subspace trick introduced by De Bie et al. De Bie et al (2004) alters the eigenspace which the cluster indicator vector is projected onto, based on the given constraints. This technique was later extended in Coleman et al (2008) to accommodate inconsistent constraints. Yu and Shi Yu and Shi (2001, 2004) encoded partial grouping information as a subspace projection. Li et al. Li et al (2009) enforced constraints by regularizing of the spectral embedding. This type of approaches usually strictly enforce given constraints. As a result, the results are often over-constrained, which

**Table 1** Table of notations

Symbol	Meaning
$\mathcal{G}$	An undirected (weighted) graph
$A$	The affinity matrix
$D$	The degree matrix
$I$	The identity matrix
$L/\bar{L}$	The unnormalized/normalized graph Laplacian
$Q/\bar{Q}$	The unnormalized/normalized constraint matrix
$vol$	The volume of graph $\mathcal{G}$

makes the algorithms sensitive to noise and inconsistencies in the constraint set. Moreover, it is non-trivial to extend these approaches to incorporate soft constraints.

We would like to stress that the pros and cons of spectral clustering as compared to other clustering schemes, such as  $K$ -means clustering, hierarchical clustering, etc., have been thoroughly studied and well established. We do not claim that constrained spectral clustering is universally superior to other constrained clustering schemes. The goal of this work is to provide a way to incorporate constraints into spectral clustering that is more flexible and principled as compared to existing constrained spectral clustering techniques.

### 3 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 formulation in Section 4. 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 unnormalized 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 von Luxburg (2007))** *Let  $L$  be the graph Laplacian of a connected graph, then:*

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 (2000) showed that the eigenvectors of the graph Laplacian can be related to the normalized cut (Ncut) of  $\mathcal{G}$ . The objective function can be written as:

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

Here

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

is called the *normalized* graph Laplacian von Luxburg (2007);  $\text{vol} = \sum_{i=1}^N D_{ii}$  is the volume of  $\mathcal{G}$ ; the first constraint  $\mathbf{v}^T \mathbf{v} = \text{vol}$  normalizes  $\mathbf{v}$ ; the second constraint  $\mathbf{v} \perp D^{1/2} \mathbf{1}$  rules out the principal eigenvector of  $\bar{L}$  as a trivial solution, because it does not define a meaningful cut on the graph. The relaxed cluster indicator  $\mathbf{u}$  can be recovered from  $\mathbf{v}$  as:

$$\mathbf{u} = D^{-1/2} \mathbf{v}.$$

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.

## 4 A Flexible Framework for Constrained Spectral Clustering

In this section, we show how to incorporate constraints into spectral clustering, whose objective function is shown as in Eq.(1). Our formulation allows both hard and soft constraints. 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.

### 4.1 The Objective Function

We encode user supervision with an  $N \times N$  constraint matrix  $Q$ . Traditionally, constrained clustering only accommodates binary constraints, Must-Link and Cannot-Link, which can be naturally encoded as follows:

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

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 assignment  $\mathbf{u}$ : the measure will increase by 1 if  $Q_{ij} = 1$  and node  $i$  and  $j$  have the same sign in  $\mathbf{u}$ ; decrease by 1 if  $Q_{ij} = -1$  but node  $i$  and  $j$  have different signs in  $\mathbf{u}$ , or,  $Q_{ij} = -1$  but node  $i$  and  $j$  have the same sign in  $\mathbf{u}$ .

To extend above encoding scheme to accommodate soft 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  $Q$  are 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  $\alpha \in \mathbb{R}$ :

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

Following the notations in Eq.(1), we substitute  $\mathbf{u}$  with  $D^{-1/2} \mathbf{v}$ , 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.(1), 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:

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

Here  $\mathbf{v}^T \bar{L} \mathbf{v}$  is the cost of the cut, which we 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  $\mathbf{v}^T \mathbf{v} = \text{vol}$  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.(2), 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.(1) can be covered as a special case of our formulation where  $\bar{Q} = I$  (no constraints are encoded) and  $\alpha = \text{vol}$  ( $\mathbf{v}^T \bar{Q} \mathbf{v} \geq \text{vol}$  is trivially satisfied given  $\bar{Q} = I$  and  $\mathbf{v}^T \mathbf{v} = \text{vol}$ ).

## 4.2 Solving the Objective Function

To solve a constrained optimization problem, we normally use the Karush-Kuhn-Tucker Theorem Kuhn and Tucker (1982), which describes the necessary conditions for the optimal solution to the problem. We can derive a set of candidates, or *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.(2), we introduce Lagrange multipliers as follows:

$$A(\mathbf{v}, \lambda, \mu) = \mathbf{v}^T \bar{L} \mathbf{v} - \lambda(\mathbf{v}^T \bar{Q} \mathbf{v} - \alpha) - \mu(\mathbf{v}^T \mathbf{v} - vol). \quad (3)$$

Then according to the KKT Theorem, any feasible solution to Eq.(2) must satisfy the following conditions:

$$\text{(Stationarity)} \quad \bar{L} \mathbf{v} - \lambda \bar{Q} \mathbf{v} - \mu \mathbf{v} = 0, \quad (4)$$

$$\text{(Primal feasibility)} \quad \mathbf{v}^T \bar{Q} \mathbf{v} \geq \alpha, \mathbf{v}^T \mathbf{v} = vol, \quad (5)$$

$$\text{(Dual feasibility)} \quad \lambda \geq 0, \quad (6)$$

$$\text{(Complementary slackness)} \quad \lambda(\mathbf{v}^T \bar{Q} \mathbf{v} - \alpha) = 0. \quad (7)$$

Note that Eq.(4) comes from taking the derivative of Eq.(3) 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.(4)-(7), we start with looking at the complementary slackness requirement in Eq.(7), which implies we must have either  $\lambda = 0$  or  $\mathbf{v}^T \bar{Q} \mathbf{v} - \alpha = 0$ . If  $\lambda = 0$ , we will have a trivial problem because the second term from Eq.(4) would be eliminated and the problem would be reduced to unconstrained spectral clustering. Therefore in the following we focus on the case where  $\lambda \neq 0$ . In this case, for Eq.(7) to hold  $\mathbf{v}^T \bar{Q} \mathbf{v} - \alpha$  must be 0. Consequently the KKT conditions become:

$$\bar{L} \mathbf{v} - \lambda \bar{Q} \mathbf{v} - \mu \mathbf{v} = 0, \quad (8)$$

$$\mathbf{v}^T \mathbf{v} = vol, \quad (9)$$

$$\mathbf{v}^T \bar{Q} \mathbf{v} = \alpha, \quad (10)$$

$$\lambda > 0, . \quad (11)$$

Under the assumption that  $\alpha$  is arbitrarily given by user and  $\lambda$  and  $\mu$  are independent variables, Eq.(8-11) cannot be solved explicitly, and it may produce infinite number of feasible solutions, if a solution exists. As a workaround, we temporarily drop the assumption that  $\alpha$  is an arbitrary value assigned by the user. Instead, we assume  $\alpha \triangleq \mathbf{v}^T \bar{Q} \mathbf{v}$ , i.e.  $\alpha$  is defined as such that Eq.(10) holds. Then we introduce an auxiliary variable,  $\beta$ , which is defined as the ratio between  $\mu$  and  $\lambda$ :

$$\beta \triangleq -\frac{\mu}{\lambda} vol. \quad (12)$$

Now we substitute Eq.(12) into Eq.(8) we obtain:

$$\bar{L} \mathbf{v} - \lambda \bar{Q} \mathbf{v} + \frac{\lambda \beta}{vol} \mathbf{v} = 0,$$

or equivalently:

$$\bar{L}\mathbf{v} = \lambda(\bar{Q} - \frac{\beta}{vol}I)\mathbf{v} \quad (13)$$

We immediately notice that Eq.(13) is a generalized eigenvalue problem for a given  $\beta$ .

Next we show that the substitution of  $\alpha$  with  $\beta$  does not compromise our original intention of lower bounding  $\mathbf{v}^T\bar{Q}\mathbf{v}$  in Eq.(2).

**Lemma 1**  $\beta < \mathbf{v}^T\bar{Q}\mathbf{v}$ .

*Proof* Let  $\gamma \triangleq \mathbf{v}^T\bar{L}\mathbf{v}$ , by left-hand multiplying  $\mathbf{v}^T$  to both sides of Eq.(13) we have

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

Then incorporating Eq.(9) and  $\alpha \triangleq \mathbf{v}^T\bar{Q}\mathbf{v}$  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

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

In summary, our algorithm 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.(13). Note that both  $\bar{L}$  and  $\bar{Q} - \beta/volI$  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  $\mathbf{v}^T\mathbf{v} = vol$ . 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.(13), 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 choose from the feasible solutions 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.(2) for any given  $\beta$  and  $\mathbf{v}^{*T}\bar{Q}\mathbf{v}^* \triangleq \alpha > \beta$ .

**Algorithm 1:** Constrained Spectral Clustering

---

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

- 1  $vol \leftarrow \sum_{i=1}^N \sum_{j=1}^N A_{ij}$ ,  $D \leftarrow \text{diag}(\sum_{j=1}^N A_{ij})$ ;
- 2  $\bar{L} \leftarrow I - D^{-1/2} A D^{-1/2}$ ,  $\bar{Q} \leftarrow D^{-1/2} Q D^{-1/2}$ ;
- 3  $\lambda_{max}(\bar{Q}) \leftarrow$  the largest eigenvalue of  $\bar{Q}$ ;
- 4 **if**  $\beta \geq \lambda_{max}(\bar{Q}) \cdot vol$  **then**
- 5 |   **return**  $\mathbf{u}^* = \emptyset$ ;
- 6 **end**
- 7 **else**
- 8 |   Solve the generalized eigenvalue system in Eq.(13);
- 9 |   Remove eigenvectors associated with non-positive eigenvalues and normalize the rest by  $\mathbf{v} \leftarrow \frac{\mathbf{v}}{\|\mathbf{v}\|} vol$ ;
- 10 |    $\mathbf{v}^* \leftarrow \text{argmin}_{\mathbf{v}} \mathbf{v}^T \bar{L} \mathbf{v}$ , 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**

---

## 4.3 A Sufficient Condition for the Existence of 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.(13) combined with the KKT conditions in Eq.(8-11) will give rise to at least one feasible solution. In this section, we discuss such a sufficient condition:

$$\beta < \lambda_{max}(\bar{Q}) \cdot vol,$$

where  $\lambda_{max}(\bar{Q})$  is the largest eigenvalue of  $\bar{Q}$ . In this case, the matrix on the right hand side of Eq.(13), namely  $\bar{Q} - \beta/vol \cdot I$ , will have at least one positive eigenvalue. Consequently, the generalized eigenvalue system in Eq.(13) 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  $\beta < \lambda_{min}(\bar{Q})vol$ ,  $\lambda_{min}(\bar{Q})$  to be the smallest eigenvalue of  $\bar{Q}$ , then  $\bar{Q} - \beta/vol \cdot I$  becomes positive definite. Then the generalized eigenvalue system in Eq.(13) will generate  $N - 1$  feasible eigenvectors (the trivial solution  $D^{1/2} \mathbf{1}$  with eigenvalue 0 is dropped).

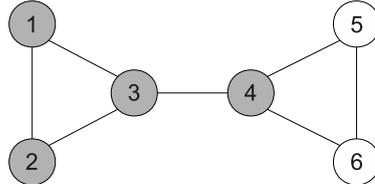
In practice, we normally choose the value of  $\beta$  within the range

$$(\lambda_{min}(\bar{Q}) \cdot vol, \lambda_{max}(\bar{Q}) \cdot vol).$$

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  $\beta < \lambda_{max}(\bar{Q}) \cdot vol$ , the value of  $\alpha$  will always be bounded by

$$\beta < \alpha \leq \lambda_{max} vol.$$

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



**Fig. 2** 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  $\{5, 6\}$ .

#### 4.4 An Illustrative Example

To illustrate how our algorithm works, we present a toy example as follows. In Fig. 2, we have a graph associated with the following affinity matrix:

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

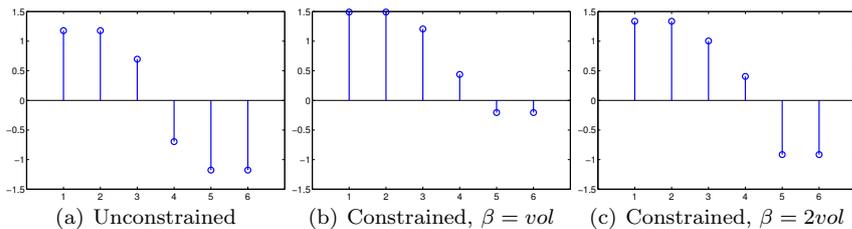
Unconstrained spectral clustering will cut the graph at edge  $(3, 4)$  and split it into two symmetric parts  $\{1, 2, 3\}$  and  $\{4, 5, 6\}$  (Fig. 3(a)).

Then we introduce constraints as encoded in the following constraint matrix:

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

$Q$  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 “complete information” constraint matrix does not happen in practice, we use it here only to make the result more explicit and intuitive.

$\bar{Q}$  has two distinct eigenvalues: 0 and 2.6667. As analyzed above,  $\beta$  must be smaller than  $2.6667 \times vol$  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  $vol$  and  $2vol$  respectively, and see how the results will be affected by different values of  $\beta$ . We solve the generalized eigenvalue system in Eq.(13), and plot the cluster indicator vector  $\mathbf{u}^*$  in Fig. 3(b) and 3(c), respectively. 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 better constraint satisfaction.



**Fig. 3** The solutions to the illustrative example in Fig. 2 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.

## 5 Interpretations of Our Formulation

### 5.1 A Graph Cut Interpretation

Unconstrained spectral clustering can be interpreted as finding the Ncut of an *unlabeled* graph. Similarly, our formulation of constrained spectral clustering in Eq.(2) can be interpreted as finding the Ncut of 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. 2). Let  $\mathbf{v}^*$  be the solution to the constrained optimization problem in Eq.(2). 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 we minimize. On the other hand,

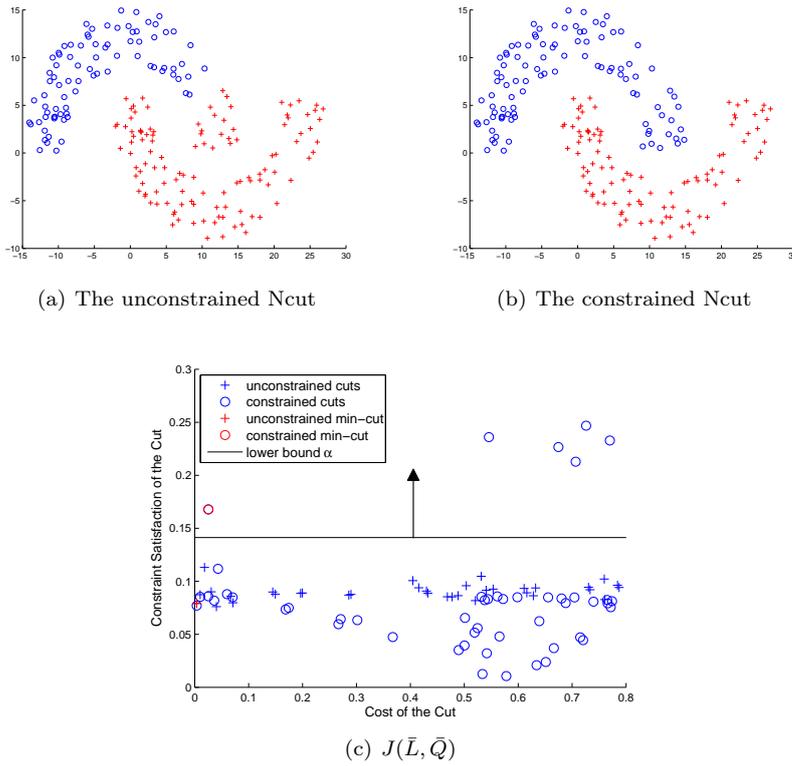
$$\alpha = \mathbf{v}^{*T}\bar{Q}\mathbf{v}^* = \mathbf{u}^{*T}Q\mathbf{u}^*$$

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  $Q \in \{-1, 0, 1\}^{N \times N}$  and  $\mathbf{u}^* \in \{-1, 1\}^N$ , then  $\alpha$  equals to the number of constraints in  $Q$  that are satisfied by  $\mathbf{u}^*$  minus the number of constraints violated. More generally, if  $Q_{ij}$  is a positive number, then  $\mathbf{u}_i^*$  and  $\mathbf{u}_j^*$  having the same sign will contribute positively to the purity of the cut, whereas different signs will contribute negatively to 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 (0 violations), which is the case where constraints in  $Q$  are completely satisfied.

### 5.2 A Geometric Interpretation

We can also interpret our formulation as constraining the joint numerical range Horn and Johnson (1990) of the graph Laplacian and the constraint matrix. Specifically, we consider:

$$J(\bar{L}, \bar{Q}) \triangleq \{(\mathbf{v}^T\bar{L}\mathbf{v}, \mathbf{v}^T\bar{Q}\mathbf{v}) : \mathbf{v}^T\mathbf{v} = 1\}. \quad (14)$$



**Fig. 4** The joint numerical range of the normalized graph Laplacian  $\bar{L}$  and the normalized constraint matrix  $\bar{Q}$ , as well as the optimal solutions to unconstrained/constrained spectral clustering.

$J(\bar{L}, \bar{Q})$  essentially maps all possible cuts  $\mathbf{v}$  to a 2-D plane, where the  $x$ -coordinate corresponds to the cost of the cut, and the  $y$ -axis corresponds to the constraint satisfaction of the cut. According to our objective in Eq.(2), we want to minimize the first term while lower-bounding the second term. Therefore, we are looking for the leftmost point among those that are above the horizontal line  $y = \alpha$ .

In Fig. 4(c), we visualize  $J(\bar{L}, \bar{Q})$  by plotting all the unconstrained cuts given by spectral clustering and all the constrained cuts given by our algorithm in the joint numerical range, based on the graph Laplacian of a Two-Moon dataset with a randomly generated constraint matrix. The horizontal line and the arrow indicate the constrained area from which we can select feasible solutions. We can see that most of the unconstrained cuts proposed by spectral clustering are well below the threshold, which suggests spectral clustering could not lead to the groundtruth partition (as shown in Fig. 4(b)) without the help of constraints.

## 6 Implementation and Extensions

In this section, we discuss the implementation issues of our method. Then we show how to generalize it to  $K$ -way partition where  $K \geq 2$ .

### 6.1 Constrained Spectral Clustering for 2-Way Partition

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  $Q$ , and a threshold  $\beta$ . Then we solve the generalized eigenvalue problem in Eq.(13) and find all the feasible generalized eigenvectors. The output is the optimal (relaxed) cluster assignment indicator  $\mathbf{u}^*$ . In practice, a partition is often derived from  $\mathbf{u}^*$  by assigning nodes corresponding to the positive entries in  $\mathbf{u}^*$  to one side of the cut, and negative entries to the other side. Our algorithm is summarized in Algorithm 1.

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

The runtime of our algorithm is dominated by that of the generalized eigen-decomposition. 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.

### 6.2 Extension to $K$ -Way Partition

Our algorithm can be naturally extended to  $K$ -way partition for  $K > 2$ , following what we usually do for unconstrained spectral clustering von Luxburg (2007): instead of only using the optimal feasible eigenvector  $\mathbf{u}^*$ , we preserve top- $(K - 1)$  eigenvectors associated with positive eigenvalues, and perform  $K$ -means algorithm based on that embedding.

Specifically, the constraint matrix  $Q$  follows the same encoding scheme:  $Q_{ij} > 0$  if node  $i$  and  $j$  are believed to belong to the same cluster,  $Q_{ij} < 0$  otherwise. To guarantee we can find  $K - 1$  feasible eigenvectors, we set the threshold  $\beta$  such that

$$\beta < \lambda_{K-1} vol,$$

where  $\lambda_{K-1}$  is the  $(K - 1)$ -th largest eigenvalue of  $\bar{Q}$ . Given all the feasible eigenvectors, we pick the top  $K - 1$  in terms of minimizing  $\mathbf{v}^T \bar{L} \mathbf{v}$ <sup>1</sup>. Let the  $K - 1$  eigenvectors form the columns of  $V \in \mathbb{R}^{N \times (K-1)}$ . We perform  $K$ -means clustering on the rows of  $V$  and get the final clustering. Algorithm 2 shows the complete routine.

Note that  $K$ -means is only one of many possible discretization techniques that can derive a  $K$ -way partition from the relaxed indicator matrix  $D^{-1/2} V^*$ . Due to the orthogonality of the eigenvectors, they can be independently discretized first, e.g. we can replace Step 11 of Algorithm 2 with:

$$\mathbf{u}^* \leftarrow \text{kmeans}(\text{sign}(D^{-1/2} V^*), K). \quad (15)$$

This additional step can help alleviate the influence of possible outliers on the  $K$ -means step in some cases.

<sup>1</sup> Note that here we assume the trivial solution, the eigenvector with all 1's, has been excluded.

**Algorithm 2:** Constrained Spectral Clustering for  $K$ -way Partition

---

**Input:** Affinity matrix  $A$ , constraint matrix  $Q$ ,  $\beta$ ,  $K$ ;  
**Output:** The cluster assignment indicator  $\mathbf{u}^*$ ;

- 1  $vol \leftarrow \sum_{i=1}^N \sum_{j=1}^N A_{ij}$ ,  $D \leftarrow \text{diag}(\sum_{j=1}^N A_{ij})$ ;
- 2  $\bar{L} \leftarrow I - D^{-1/2}AD^{-1/2}$ ,  $\bar{Q} \leftarrow D^{-1/2}QD^{-1/2}$ ;
- 3  $\lambda_{max} \leftarrow$  the largest eigenvalue of  $\bar{Q}$ ;
- 4 **if**  $\beta \geq \lambda_{K-1} vol$  **then**
- 5 |   **return**  $\mathbf{u}^* = \emptyset$ ;
- 6 **end**
- 7 **else**
- 8 |   Solve the generalized eigenvalue system in Eq.(13);
- 9 |   Remove eigenvectors associated with non-positive eigenvalues and normalize the rest by  $\mathbf{v} \leftarrow \frac{\mathbf{v}}{\|\mathbf{v}\|} vol$ ;
- 10 |    $V^* \leftarrow \text{argmin}_{V \in \mathbb{R}^{N \times (K-1)}} \text{trace}(V^T \bar{L} V)$ , where the columns of  $V$  are a subset of the feasible eigenvectors generated in the previous step;
- 11 |   **return**  $\mathbf{u}^* \leftarrow \text{kmeans}(D^{-1/2}V^*, K)$ ;
- 12 **end**

---

Moreover, notice that the feasible eigenvectors, which are the columns of  $V^*$ , are treated equally in Eq.(15). This may not be ideal in practice because these candidate cuts are not equally favored by graph  $\mathcal{G}$ , i.e. some of them have higher costs than the other. Therefore, we can weight the columns of  $V^*$  with the inverse of their respective costs:

$$\mathbf{u}^* \leftarrow \text{kmeans}(\text{sign}(D^{-1/2}V^*(V^{*T}\bar{L}V^*)^{-1}), K). \quad (16)$$

### 6.3 Using Constrained Spectral Clustering for Transfer Learning

The constrained spectral clustering framework naturally fits into the scenario of transfer learning between two graphs. Assume we have two graphs, a source graph and a target graph, which share the same set of nodes but have different sets of edges (or edge weights). The goal is to transfer knowledge from the source graph so that we can improve the cut on the target graph. The knowledge to transfer is derived from the source graph in the form of soft constraints.

Specifically, let  $\mathcal{G}_S(V, E_S)$  be the source graph,  $\mathcal{G}_T(V, E_T)$  the target graph.  $A_S$  and  $A_T$  are their respective affinity matrices. Then  $A_S$  can be considered as a constraint matrix with only ML constraints. It carries the complete knowledge from the source graph, and we can transfer it to the target graph using our constrained spectral clustering formulation:

$$\text{argmin}_{\mathbf{v} \in \mathbb{R}^N} \mathbf{v}^T \bar{L}_T \mathbf{v}, \text{ s.t. } \mathbf{v}^T A_S \mathbf{v} \geq \alpha, \mathbf{v}^T \mathbf{v} = vol, \mathbf{v} \neq D_T^{1/2} \mathbf{1}. \quad (17)$$

$\alpha$  is now the lower bound of how well the knowledge from the source graph must be enforced on the target graph. To solution to this is similar to Eq.(13):

$$\bar{L}_T \mathbf{v} = \lambda (\bar{A}_S - \frac{\beta}{\text{vol}(\mathcal{G}_T)} I) \mathbf{v} \quad (18)$$

Note that since the largest eigenvalue of  $\bar{A}_S$  corresponds to a trivial cut, in practice we should set the threshold such that  $\beta < \lambda_1 vol$ ,  $\lambda_1$  to be the second largest eigenvalue of  $\bar{A}_S$ . This will guarantee a feasible eigenvector that is non-trivial.

## 7 Testing and Innovative Uses of Our Work

We begin with three sets of experiments to test our approach on standard spectral clustering data sets. We then show that since our approach can handle large amounts of soft constraints in a flexible fashion, this opens up two innovative uses of our work: encoding multiple metrics for translated document clustering and transfer learning for fMRI analysis.

We aim to answer the following questions with the empirical study:

- Can our algorithm effectively incorporate side information and generate semantically meaningful partitions?
- Does our algorithm converge to the underlying groundtruth partition as more constraints are provided?
- How does our algorithm perform on real-world datasets, as evaluated against groundtruth labeling, with comparison to existing techniques?
- How well does our algorithm handle soft constraints?
- How well does our algorithm handle large amounts of constraints?

Recall that in Section 1 we listed four different types of side information: explicit pairwise constraints, partial labeling, additional metrics, and transfer of knowledge. The empirical results presented in this section are arranged accordingly.

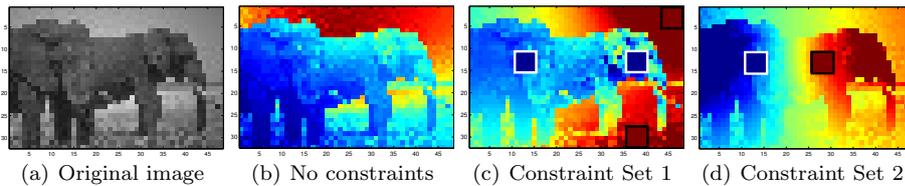
All but one (the fMRI scans) datasets used in our experiments are publicly available online. We implemented our algorithm in MATLAB, which is also online at <http://bayou.cs.ucdavis.edu/>.

### 7.1 Explicit Pairwise Constraints: Image Segmentation

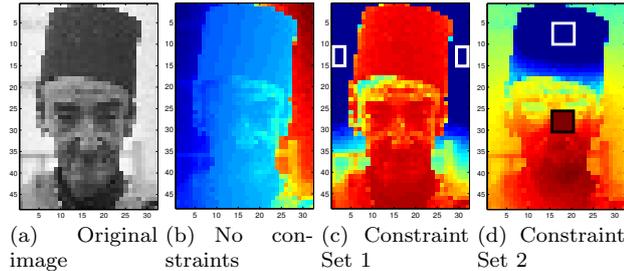
We demonstrate the effectiveness of our algorithm in the context of image segmentation using explicit pairwise constraints assigned by users.

We choose image segmentation 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 users; 3) instead of generating random constraints, we can provide semantically meaningful constraints to see if the constrained partition conforms to our expectation.

The images we used were chosen from the *Berkeley Segmentation Dataset and Benchmark* Martin et al (2001). 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. 5(a) and 6(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 Shi and Malik (2000) 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



**Fig. 5** Segmentation results of the elephant image. The images are reconstructed based on the relaxed cluster indicator  $\mathbf{u}^*$ . 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.



**Fig. 6** Segmentation results of the face image. The images are reconstructed based on the relaxed cluster indicator  $\mathbf{u}^*$ . 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.

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. 5 and 6. 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. 5(b), the unconstrained spectral clustering partitioned the elephant image into two parts: the sky (red 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. 5(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. 5(c)). 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 to

$$\beta = \lambda_{max} \times vol \times \left( 0.5 + 0.4 \times \frac{\# \text{ of constraints}}{N^2} \right), \quad (19)$$

where  $\lambda_{max}$  is the largest eigenvalue of  $\bar{Q}$ . In this way,  $\beta$  is always between  $0.5\lambda_{max}vol$  and  $0.9\lambda_{max}vol$ , and it will gradually increase as the number of constraints increases. From Fig. 5(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. 5(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. 5(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.

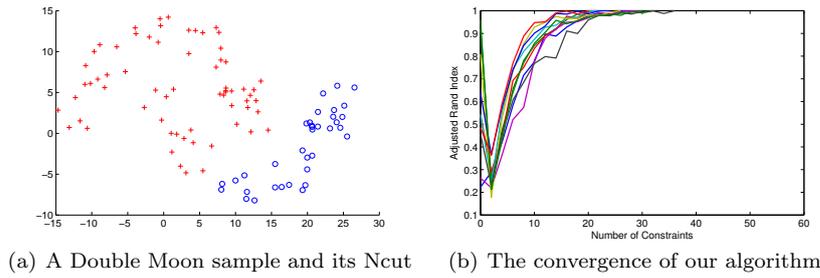
Similarly, we applied our method on a human face image as shown in Fig. 6(a). The unconstrained spectral clustering failed to isolate the human face from the background (Fig. 6(b)). This is because the tall hat breaks the spatial continuity between the left side of the background and the right side. 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. 6(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. 6(d)).

## 7.2 Explicit Pairwise Constraints: The Double Moon Dataset

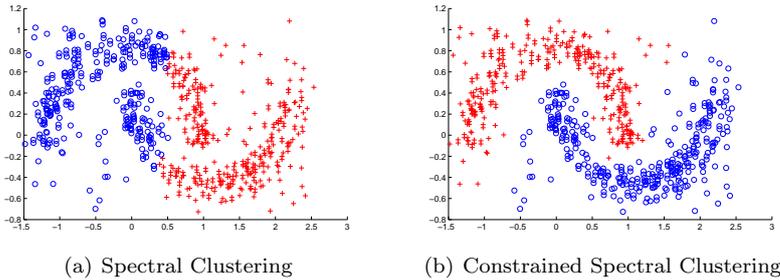
We further examine the behavior of our algorithm on a synthetic dataset using explicit constraints that are derived from underlying groundtruth.

We claim that our formulation is a natural extension to spectral clustering. The question to ask is then whether the output of our algorithm converge to that of spectral clustering. More specifically, consider the groundtruth partition defined by performing spectral clustering on an ideal distribution. We draw an imperfect sample from the distribution, on which spectral clustering would not be able to find the groundtruth partition. Then if we perform our algorithm on this imperfect sample, as more constraints are provided, we want to know whether or not the partition found by our algorithm would converge to the groundtruth partition.

To answer the question, we used the Double Moon distribution. As we illustrated in Fig. 1, spectral clustering is able to find the two moons when the sample is dense enough. In Fig. 7(a), we generated an under-sampled instance of the distribution with 100 data points, on which unconstrained spectral clustering could no longer find the groundtruth partition. Then we performed our algorithm on this imperfect sample, and compared the partition found by our algorithm to the groundtruth partition, in terms of adjusted Rand index Hubert and Arabie (1985). Higher value means the two partitions agree better; 1 means an exact match between the two partitions. For each sample, we generated 50 different sets of random constraints and recorded the average adjusted Rand index. We repeated the process on 10 different random samples of the same size and reported the results in Fig. 7(b). We can see that our algorithm consistently converge to the groundtruth



**Fig. 7** The convergence of our algorithm on 10 random samples from the Double Moon distribution.



**Fig. 8** The partition of a noisy Double Moon sample.

result as more constraints are provided. Notice that there is performance drop when an extreme small number of constraints are provided (less than 10), which is expected because such small number of constraints are insufficient to hint a different partition, and consequentially lead to random perturbation to the results. However, the results were soon stabilized when more constraints came in.

To illustrate the robustness of our approach, we created a Double Moon sample with uniform background noise. As shown in Fig. 8, although the sample is dense enough (600 data points in total), spectral clustering fails to find the correctly identify the two moons, due to the influence of background noise (100 data points). However, with 20 constraints, our algorithm successfully recovers the two moons, and reduces the influence of background to minimum.

### 7.3 Constraints from Partial Labeling: UCI Benchmarks

Next we evaluate the performance of our algorithm by clustering UCI benchmark datasets Asuncion and Newman (2007), using constraints derived from groundtruth labeling.

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: 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

**Table 2** The UCI benchmarks

Identifier	#Instances	#Attributes
Hepatitis	80	19
Iris	100	4
Wine	130	13
Glass	214	9
Ionosphere	351	34
WDBC	569	30

to be well-separated from the other two. For the same reason we removed Class 3 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. To generate constraints, we randomly selected pairs of nodes that the unconstrained spectral clustering wrongly partitioned, and fill in the correct relation in  $Q$  according to groundtruth labels. The quality of the clustering results was measured by adjusted Rand index. Since the constraints are guaranteed to be correct, we set  $\beta$  such that there would be only one feasible eigenvector, i.e. the one that best conforms to the constraint matrix  $Q$ .

In addition to comparing our algorithm (CSP) to unconstrained spectral clustering, we implemented two state-of-the-art techniques:

- Spectral Learning (SL) Kamvar et al (2003) modifies the affinity matrix of the original graph directly:  $A_{ij}$  is set to 1 if there is a ML between node  $i$  and  $j$ , 0 for CL.
- Semi-Supervised Kernel  $K$ -means (SSKK) Kulis et al (2005) adds penalties to the affinity matrix based on the given constraints, and then performs kernel  $K$ -means on the new kernel to find the partition.

We also tried the algorithm from Yu and Shi (2001, 2004), which encodes partial grouping information as a projection matrix, the subspace trick from De Bie et al (2004), and the affinity propagation algorithm from Lu and Carreira-Perpiñán (2008). However, we were not able to achieve better results using these algorithms as compared to SL and SSKK, thus their results are not reported. Xu et al. Xu et al (2005) proposed a modification to SL, where the constraints are encoded in the same way, but instead of the normalized graph Laplacian, they suggested to use the random walk matrix for partition. We performed their algorithm on the datasets we used, which produced largely identical results to that of SL.

We report the adjusted Rand index against the number of constraints used (ranging from 50 to 500) so that we can see how the quality of clustering varies when more constraints are added in. At each stop, we randomly generate 100 sets of constraints and reported the mean, maximum and minimum adjusted Rand index of the 100 random trials, as shown in Fig. 9. We also reported the ratio of the constraints that were actually satisfied by the constrained partition in Fig. 10. From the results we can tell:

- Across all six datasets, our algorithm is able to effectively utilize the constraints and improve over unconstrained spectral clustering (Baseline). On the one hand, our algorithm can quickly improve the results with a small amount

**Table 3** The Reuters Multilingual dataset

Language	#Documents	#Words
English	2000	21,531
French	2000	24,893
German	2000	34,279
Italian	2000	15,506
Spanish	2000	11,547

of constraints. On the other hand, as more constraints are provided, the performance of our algorithm consistently increases and tends to converge to the groundtruth partition.

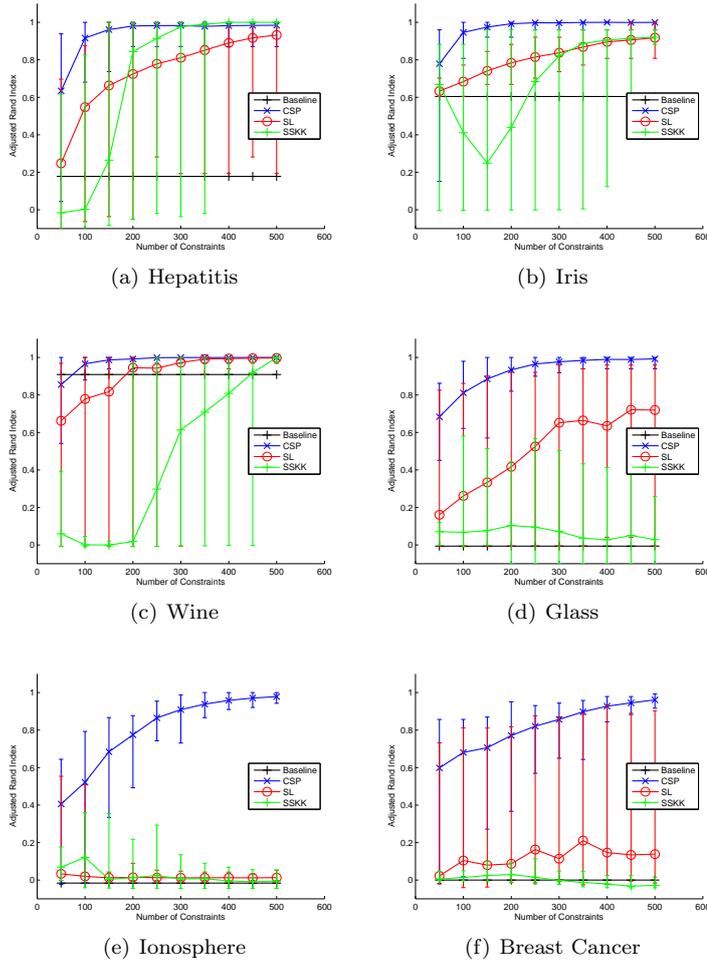
- Our algorithm outperforms the competitors by a large margin (Fig. 9). Since we have control over the lower-bounding threshold  $\alpha$ , our algorithm is able to satisfy almost all the given constraints (Fig. 10).
- The performance of our results has significantly smaller variance over different random constraint sets than those of its competitors (Fig. 9 and 10), and the variance quickly decreases as more constraints are provided. This suggests that our algorithm would performance more consistently in practice.
- Our algorithm performs especially well on sparse graphs, i.e. Fig. 9(e)(f), where the competitors suffer. The reason is that when the graph is too sparse, it implies many “free” cuts that are equally good to unconstrained spectral clustering. Even after introducing a small number of constraints, the modified graph remains too sparse for SL and SSKK, which are not able to identify the groundtruth cut. However, these free cuts are not equivalent when judged by the constraint matrix  $Q$  in our algorithm, which could easily identify the one cut that minimizes  $\mathbf{v}^T \tilde{Q} \mathbf{v}$ . As a result, our algorithm outperforms SL and SSKK significantly.

#### 7.4 Constraints from Additional Metrics: The Reuters Multilingual Dataset

We test our algorithm with soft constraints derived from additional metrics of a same dataset.

We used the Reuters Multilingual dataset Amini et al (2009). Each time we randomly sampled 1000 documents which were originally written in one language and then translated into four others, respectively. The statistics of the dataset is listed in Table 3. These documents came with groundtruth labels that categorize them into six topics ( $K = 6$ ). We constructed one graph based on the English version, and another graph based on a translation. Then we used one of the two graphs as the constraint matrix  $Q$ , whose entries can then be considered as soft ML constraints. We enforce this constraint matrix to the other graph to see if it can help improve the clustering. We did not compare our algorithm to existing techniques because they are unable to incorporate soft constraints.

As shown in Fig. 11, spectral clustering performs better on the original version than the translated versions, which is expected. If we use the original version as the constraints and enforce that onto a translated version using our algorithm, we yield a constrained clustering that is not only better than the unconstrained clustering on the translated version, but also even better than the unconstrained



**Fig. 9** The performance of our algorithm (CSP) on six UCI datasets, with comparison to unconstrained spectral clustering (Baseline) and the Spectral Learning algorithm (SL). Adjusted Rand index over 100 random trials is reported (mean, min, and max).

clustering on the original version. This indicates that our algorithm is not merely a tradeoff between the original graph and the given constraints. Instead it is able to integrate the knowledge from the constraints into the original graph and achieve a better partition.

### 7.5 Transfer of Knowledge: fMRI Analysis

Finally we apply our algorithm to transfer learning on the resting state fMRI data.

An fMRI scan of a person consists of a sequence of 3D images over time. We can construct a graph from a given scan, such that a node in the graph corresponds to a voxel in the image, and the edge weight between two nodes is (the absolute

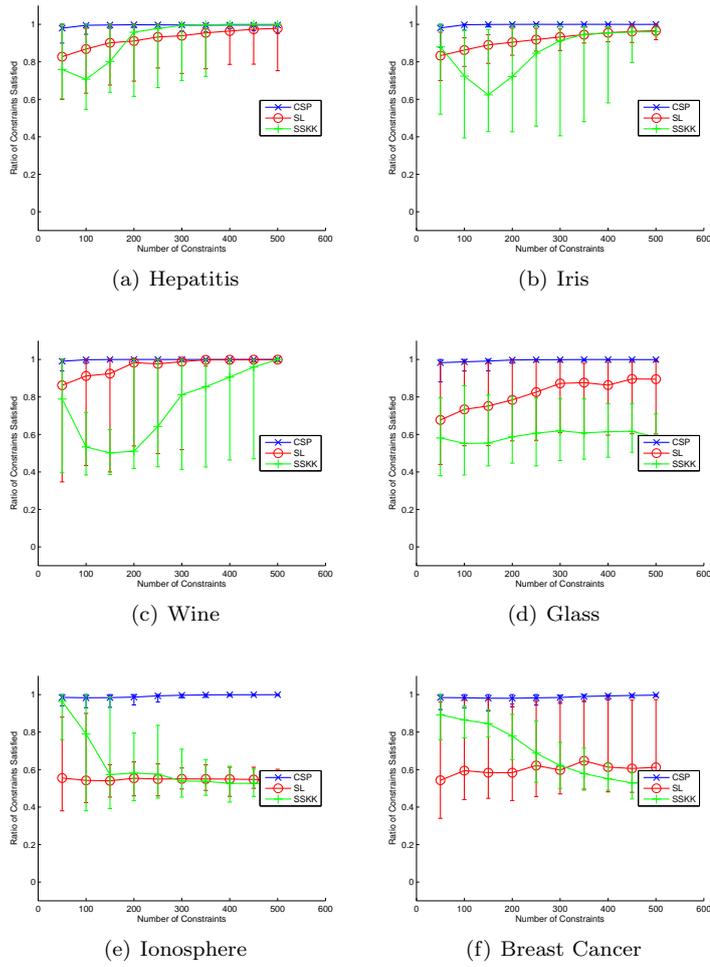


Fig. 10 The ratio of constraints that are actually satisfied.

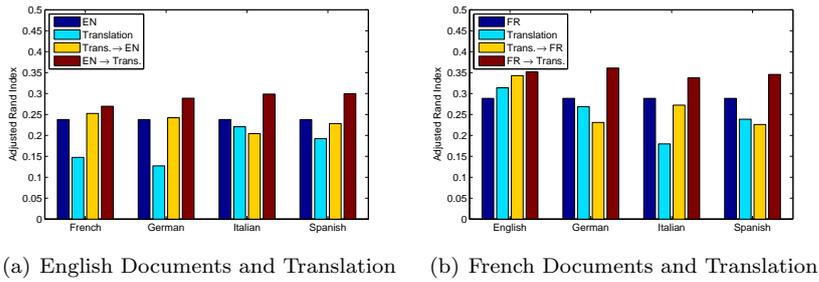
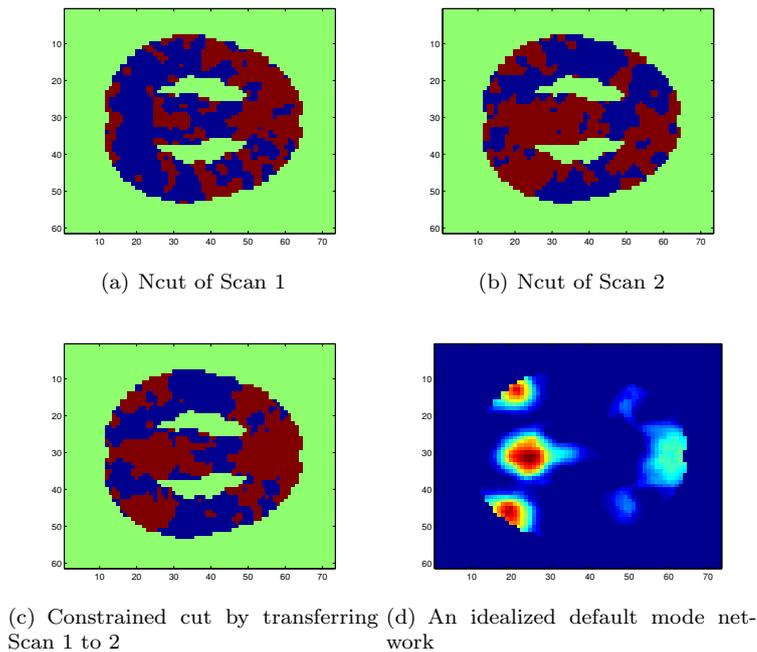


Fig. 11 The performance of our algorithm on Reuters Multilingual dataset.

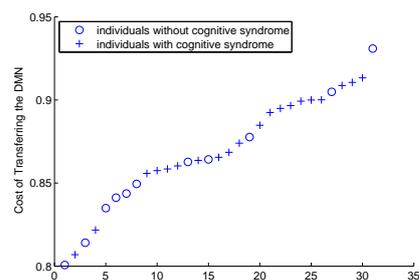


**Fig. 12** Transfer learning on fMRI scans.

value of) the correlation between the two time sequences associated with those two voxels. Previous work has shown that by applying spectral clustering to the resting state fMRI scan graph we can find the substructures in the brain that are periodically and simultaneously activated over time in the resting state, which may indicate a network associated with certain functions van den Heuvel et al (2008).

One of the challenges of resting state fMRI analysis is instability. Noise can be easily introduced into the scan result, e.g. the person moved his/her head during the scan, the person was not at resting state (actively thinking about things during the scan), etc. Consequently, the result of spectral clustering becomes unreliable. For instance, we apply spectral clustering to two fMRI scans of *the same* person on two different days. As shown in Fig. 12(a) and (b), the normalized min-cuts on the two different scans are so different that they provide little insight into the brain activity of that person. To overcome this problem, we use our formulation to transfer knowledge from Scan 1 to Scan 2 and get a constrained cut, as shown in Fig. 12(c). This cut represents what the two scans both agree on. The pattern captured by Fig. 12(c) is actually the *default mode network* (DMN) of people at resting state, which is the network that is periodically activated at resting state (Fig. 12(d) shows the idealized DMN as specified by domain experts).

To further illustrate the practicability of our approach, we transfer the idealized DMN in Fig. 12(d) to a set of fMRI scans of elderly individuals. The dataset was collected and processed within the research program of the University of California at Davis Alzheimer’s Disease Center (UCD ADC). The individuals can be categorized into two groups: those diagnosed with cognitive syndrome (20 individ-



**Fig. 13** The costs of transferring the idealized default mode network to the fMRI scans of two groups of elderly individuals.

uals) and those without cognitive syndrome (11 individuals). For each individual scan, we encoded the idealized DMN into a constraint matrix (using RBF kernel), and enforce the constraints onto the original fMRI scan graph. We then computed the cost of the constrained cut that corresponded to the DMN. Note that if the constrained cut has a higher cost, it means there is greater disagreement between the original graph and the given constraints, which is the idealized DMN, and vice versa.

In Fig. 13, we plot the costs of transferring the DMN to both groups of individuals. We can clearly see that the costs of transferring the DMN to people without cognitive syndrome tend to be lower than to people with cognitive syndrome. This conforms well to the observation made in a recent study that the DMN is often disrupted for people with Alzheimer’s disease Buckner et al (2008).

## 8 Conclusion

In this work we proposed a principled and flexible framework for constrained spectral clustering that can incorporate large amounts of both hard and soft constraints. The flexibility of our framework lends itself to the use of all types of side information: pairwise constraints, partial labeling, additional metrics, and transfer learning. Our formulation is a natural extension to unconstrained spectral clustering and can be solved efficiently using generalized eigendecomposition. We demonstrated the effectiveness of our approach on a variety of datasets: the synthetic Two-Moon dataset, image segmentation, the UCI benchmarks, the multilingual Reuters texts, and resting state fMRI scans. The comparison to existing techniques validated the advantage of our approach.

## References

- Amini MR, Usunier N, Goutte C (2009) Learning from multiple partially observed views - an application to multilingual text categorization. In: *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pp 28–36
- Asuncion A, Newman D (2007) UCI machine learning repository. URL <http://www.ics.uci.edu/~mlern/{MLR}epository.html>

- Basu S, Davidson I, Wagstaff K (eds) (2008) *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC
- Buckner RL, Andrews-Hanna JR, Schacter DL (2008) The brain's default network. *Annals of the New York Academy of Sciences* 1124(1):1–38
- Coleman T, Saunderson J, Wirth A (2008) Spectral clustering with inconsistent advice. In: *ICML*, pp 152–159
- Davidson I, Ravi SS (2006) Identifying and generating easy sets of constraints for clustering. In: *AAAI*
- Davidson I, Ravi SS (2007a) The complexity of non-hierarchical clustering with instance and cluster level constraints. *Data Min Knowl Discov* 14(1):25–61
- Davidson I, Ravi SS (2007b) Intractability and clustering with constraints. In: *ICML*, pp 201–208
- Davidson I, Ravi SS, Ester M (2007) Efficient incremental constrained clustering. In: *KDD*, pp 240–249
- De Bie T, Suykens JAK, De Moor B (2004) Learning from general label constraints. In: *SSPR/SPR*, pp 671–679
- Drineas P, Frieze AM, Kannan R, Vempala S, Vinay V (2004) Clustering large graphs via the singular value decomposition. *Machine Learning* 56(1-3):9–33
- van den Heuvel M, Mandl R, Hulshoff Pol H (2008) Normalized cut group clustering of resting-state fmri data. *PLoS ONE* 3(4):e2001
- Horn R, Johnson C (1990) *Matrix analysis*. Cambridge Univ. Press
- Hubert L, Arabie P (1985) Comparing partitions. *Journal of Classification* 2:193–218
- Ji X, Xu W (2006) Document clustering with prior knowledge. In: *SIGIR*, pp 405–412
- Kamvar SD, Klein D, Manning CD (2003) Spectral learning. In: *IJCAI*, pp 561–566
- Kuhn H, Tucker A (1982) Nonlinear programming. *ACM SIGMAP Bulletin* pp 6–18
- Kulis B, Basu S, Dhillon IS, Mooney RJ (2005) Semi-supervised graph clustering: a kernel approach. In: *ICML*, pp 457–464
- Li Z, Liu J, Tang X (2009) Constrained clustering via spectral regularization. In: *CVPR*, pp 421–428
- Lu Z, Carreira-Perpiñán MÁ (2008) Constrained spectral clustering through affinity propagation. In: *CVPR*
- von Luxburg U (2007) A tutorial on spectral clustering. *Statistics and Computing* 17(4):395–416
- Martin D, Fowlkes C, Tal D, Malik J (2001) 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*, vol 2, pp 416–423
- Ng AY, Jordan MI, Weiss Y (2001) On spectral clustering: Analysis and an algorithm. In: *NIPS*, pp 849–856
- Pan SJ, Yang Q (2010) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22(10):1345–1359
- Shi J, Malik J (2000) Normalized cuts and image segmentation. *IEEE Trans Pattern Anal Mach Intell* 22(8):888–905
- Wang F, Ding CHQ, Li T (2009) Integrated KL (K-means - Laplacian) clustering: A new clustering approach by combining attribute data and pairwise relations. In: *SDM*, pp 38–48

- 
- Wang X, Davidson I (2010) Flexible constrained spectral clustering. In: KDD '10: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 563–572
- White S, Smyth P (2005) A spectral clustering approach to finding communities in graph. In: SDM
- Xu Q, desJardins M, Wagstaff K (2005) Constrained spectral clustering under a local proximity structure assumption. In: FLAIRS Conference, pp 866–867
- Yu SX, Shi J (2001) Grouping with bias. In: NIPS, pp 1327–1334
- Yu SX, Shi J (2004) Segmentation given partial grouping constraints. *IEEE Trans Pattern Anal Mach Intell* 26(2):173–183