Constrained Spectral Clustering using L1 Regularization

Jaya Kawale* Daniel Boley*

Abstract

Constrained spectral clustering is a semi-supervised learning problem that aims at incorporating user-defined constraints in spectral clustering. Typically, there are two kinds of constraints: (i) *must-link*, and (ii) *cannot-link*. These constraints represent prior knowledge indicating whether two data objects should be in the same cluster or not; thereby aiding in clustering. In this paper, we propose a novel approach that uses convex subproblems to incorporate constraints in spectral clustering and co-clustering. In comparison to the prior state-of-art approaches, our approach presents a more natural way to incorporate constraints in the spectral methods and allows us to make a trade off between the number of satisfied constraints and the quality of partitions on the original graph. We use an L_1 regularizer analogous to LASSO, often used in literature to induce sparsity, in order to control the number of constraints satisfied. Our approach can handle both *must-link* and *cannot-link* constraints, unlike a large number of previous approaches that mainly work on the former. Further, our formulation is based on the reduction to a convex subproblem which is relatively easy to solve using existing solvers. We test our proposed approach on real world datasets and show its effectiveness for both spectral clustering and co-clustering over the prior state-of-art.

1 Introduction

Constrained spectral clustering is an area of active research within the broad domain of constrained clustering that aims at spectral partitioning of the graph of objects by incorporating user defined constraints in the partitioning process. A user can specify two kinds of constraints: *must-link* and *cannot-link* (see [1, 2] for a general discussion) between the entities. In this paper, we present a general framework to handle constraints in both spectral clustering and co-clustering. Incorporating constraints in spectral clustering has been explored previously by various works [3, 4, 5, 6, 7], however not much research work has been done towards incorporating these constraints in spectral co-clustering setting - apart from the notable exception of [8].

Constrained spectral clustering approaches [3, 4, 5] and similar in spirit constrained co-clustering approaches [8] handle only *must-link* constraints. These approaches overlay the constraint graph on the original graph and modify the graph Laplacian according to the specified *must-link* constraints. The clusters are then computed based on the minimization of the normalized cut [9] of the modified graph Laplacian. There are two main issues with this approach. Firstly, by treating the constraints just like another graph, it becomes harder to ensure that they are satisfied while maintaining a reasonable partition quality on the original graph. In contrast, it would be more desirable to be able to trade the number of satisfied constraints against the quality of the partition on the original graph because in many real applications the constraints can be poorly defined and sometimes even inconsistent. Secondly, it is not clear how to extend the method to handle *cannot-link* constraints.

Alternatively, other approaches to handle constraints in spectral clustering operate by restricting the feasible solution space instead of directly manipulating the graph Laplacian. Wang et al. [6] proposed an approach for constrained clustering that handled both *must-link* and *cannot-link* constraints. Further, their approach could also handle real valued constraints and allowed to control the satisfaction of constraints. They showed performance improvements over existing methods of constrained spectral clustering. However, the approach relies on a quadratically constrained quadratic optimization with an objective function which is not necessarily convex (a quadratic form could be indefinite). By having a quadratic constraint to a quadratic minimization criterion, a large weight on the constraint term would make the effect of the quadratic graph-Laplacian term almost negligible. The solution of their problem involves the computation of all the eigenvalues, and then all the eigenvectors corresponding to the positive eigenvalues. For a large graph, this operation can be quite expensive. The number of eigenvectors could be as large as O(vertices), and the collection of eigenvectors will most certainly be dense. In contrast, we use a 1-norm constraint which makes it easier to enforce

*{kawale, boley}@cs.umn.edu Dept. of Computer Science, University of Minnesota

a large number of constraints while limiting effect of the rest. By using a L_1-norm on the penalty term, we can induce sparsity in the penalty part without using a weight so large as to completely bury the effect of the original graph.

We present a novel approach that uses convex sub-problems to handle *must-link* and *cannot-link* constraints in spectral clustering and co-clustering. To the best of our knowledge this is the first work that handles both the types of constraints in a spectral co-clustering setting. Our proposed approach handles the constraints in a principled manner using a L_1 regularizer modeled after LASSO [10]. The L_1 regularizer has been popularly used in literature as a shrinkage method to drive some of the coefficients of the feature weight vector in regression to zero. We use a generalization of the LASSO idea as a method to satisfy constraints while minimizing the normalized cut of the graph Laplacian. This allows us to make a trade off between the number of satisfied constraints and the quality of the partitions on the original graph. Additionally, we cast our non-convex problem into a sequence of convex sub-problems which are easy to solve. Empirical evaluation over different real datasets shows the effectiveness of our approach.

2 Related Work

Constrained clustering is a class of semi-supervised learning algorithms. It allows users to specify constraints in order to influence the clustering process. There are primarily two kinds of constraints between pairs of objects that are clustered: *must-link* and *cannot-link* [1]. The *must-link* constraint indicates that the two objects should lie in the same cluster, whereas *cannot-link* constraint indicate that the two objects should not be in the same cluster. Previous experience indicates that incorporating prior knowledge as constraints in clustering can improve clustering performance [1, 11, 5]. The seminal work by Wagstaff et al. [1] incorporated the constraints in the KMeans clustering algorithm.

There are two kinds of methods to incorporate constraints in spectral clustering: (a) methods that directly manipulate the graph Laplacian, and (b) methods that restrict the feasible solution space. Kamvar et al. [3] included constraints by modifying the affinity matrix A by assigning $A_{ij} = 1$ for each pair of *must-link* between objects i and j and by assigning $A_{ij} = 0$ for each pair of *cannot-link* between objects i and j. Ji et al. [5] incorporated constraints by adding to the normalized cut [9] objective function a L_2 penalty term which measures the number of constraints that are not satisfied by a partition. Xu et al. [4] proposed a modification of [3] to handle local proximity structure in the graph. Lu at al. [7] proposed a method to propagate the constraints in the affinity matrix using an interpretation of a Gaussian process. Wang et al. [12] propose a method to combine k-means clustering on attribute information and spectral clustering on relational information. The second set of methods involves restricting the feasible solution space using the pair-wise constraints. De Bie et al. [13] restricted the eigenspace on which the cluster assignment is projected with the help of constraints. Coleman at al. [14] presented a framework to include inconsistent advice in spectral clustering. Wang et al. [6] proposed a degree of belief concept to incorporate real valued constraints and create a constrained optimization problem to solve spectral clustering with constraints. Shi et al. [8] proposed a modification to spectral co-clustering to handle *must-link* but not *cannot-link* constraints in the data.

3 Background and Preliminaries

In this section, we give a brief overview of spectral clustering and the existing methods for constrained spectral clustering as a background for our approach.

3.1 Spectral Clustering Spectral clustering is an extensively used graph partitioning algorithm. The most widely used objective function to evaluate the graph partitions in spectral clustering is normalized cut [9]. Let $G = \{V, E, W\}$ be an undirected graph where V be the set of vertices in the graph and $w_{uv} \in W$ be the affinity of the edge $e_{uv} \in E$ between vertex $u \in V$ and $v \in V$. Let S_i and S_j be two cluster partitions, and then the affinity between the two clusters can be defined as follows:

$$W(S_i, S_j) = \sum_{u \in S_i, v \in S_j} w_{uv} \tag{3.1}$$

The "normalized cut" aims at minimizing the affinity between the partitions $S_1, \ldots, S_K$ relative to the size of each partition. It is defined as the following minimization in the case that we measure the size of each partition in terms of edges:

$$NC_{edge} = \min_{S_1,\ldots,S_K} \left\{ \sum_{k=1}^{K} \frac{W(S_k, \bar{S}_k)}{W(S_k, V)} \right\} \tag{3.2}$$

where $\bar{S}_k$ indicate the vertices in V that are not in S_k. We can also define the cut relative to the number of nodes in each partition as follows:

$$NC_{node} = \min_{S_1,\ldots,S_K} \left\{ \sum_{k=1}^{K} \frac{W(S_k, \bar{S}_k)}{|S_k|} \right\} \tag{3.3}$$

The numerators in (3.2) and (3.3) represent the affinity between cluster S_k and all other clusters, whereas

the denominators represent the balance between the partitions in terms of number of edges and vertices, respectively. Suppose $\mathbf{x}_k$ is an indicator vector over the vertices showing their membership in cluster k, where $x_{uk} = 1$ if vertex u is in cluster k, else $x_{uk} = 0$. The degree matrix $D = (d_{uv})$ is the diagonal matrix with diagonal entries $d_{uu} = \sum_{t=1}^{n} w_{ut}$, where $n = |V|$ is the total number of vertices. It can be shown [9, 15] that the normalized cuts can be written in terms of the generalized Rayleigh quotients:

$$NC_{edge} = \min_{\mathbf{x}_1,\ldots,\mathbf{x}_K} \left\{ \sum_{k=1}^{K} \frac{\mathbf{x}_k^T L \mathbf{x}_k}{\mathbf{x}_k^T D \mathbf{x}_k} \right\} \tag{3.4}$$

$$NC_{node} = \min_{\mathbf{x}_1,\ldots,\mathbf{x}_K} \left\{ \sum_{k=1}^{K} \frac{\mathbf{x}_k^T L \mathbf{x}_k}{\mathbf{x}_k^T \mathbf{x}_k} \right\} \tag{3.5}$$

(both subject to $x_{uk} \in \{0, 1\}$). The quantity $L = D - W$ is called the unnormalized graph Laplacian. Minimizing the cut objective functions is known to be NP hard, so it is usually solved by treating a relaxed problem where $\mathbf{x}_k$ can be a vector of any real values, so that (3.4) and (3.5) becomes a generalized and an ordinary eigen problem, respectively.

3.2 Constrained Spectral Clustering In the case of separating into $K = 2$ clusters, one of the main set of approaches to handle constraints in spectral clustering is via overlaying a constraint graph on the original graph and minimizing the normalized cut of the resulting graph Laplacian. The *must-link* graph consists of the same vertices as the original graph, but the set of edges consist of the *must-link* constraints. If L_C is the unnormalized Laplacian for the *must-link* graph, then the problem to solve in [8] is to minimize the weighted sum of the cut and the number of constraint violations:

$$\min \mathbf{x}^T L \mathbf{x} + \delta \mathbf{x}^T (L_C) \mathbf{x} \text{ s.t. } \mathbf{x}^T 1 = 0, \mathbf{x}^T \mathbf{x} = 1. \tag{3.6}$$

This approach has been developed only for *must-link* constraints.

Alternatively, [6] present an approach to handle spectral clustering as a constraint satisfaction problem, forming a matrix Q defined by

$$Q_{ij} = \begin{cases} +1 & \text{if nodes } i, j \text{ must be linked} \\ -1 & \text{if nodes } i, j \text{ must not be linked} \\ 0 & \text{otherwise.} \end{cases}$$

They propose to find a solution that minimizes the following equation:

$$\min \mathbf{x}^T L \mathbf{x} \text{ s.t. } x^T Q \mathbf{x} \geq \alpha, \mathbf{x}^T \mathbf{x} = \text{Vol}(\mathcal{G}), \mathbf{x} \neq \mathbf{1}, \tag{3.7}$$

where $\text{Vol}(\mathcal{G})$ is the volume of original graph (sum of all edge affinities). In order to solve the non-convex problem (3.7), the authors propose to compute an almost complete generalized eigen-decomposition involving the matrices L, Q.

4 Algorithm

Instead of treating the *must-link* constraints as just another graph, our approach is to impose them as a set of linear constraints, and design an objective function so that the violations of the *must-link* constraints are as "sparse" as possible. This approach makes it easy to incorporate *cannot-link* constraints, while still leading to a sequence of convex optimization problems which is easy to solve. We develop the method for the 2-cluster case.

4.1 Unconstrained Optimization Problem As noted above, the usual spectral clustering method finds the eigenvector corresponding to the smallest non-zero eigenvalue of L (for minimizing the node cut NC_{node}) or the smallest generalized eigenvalue of $L\mathbf{x} = \lambda D\mathbf{x}$ (for minimizing edge cut NC_{edge}). The latter is equivalent to solving the following optimization problem

$$\begin{array}{ll} \min_{\mathbf{x}} & \frac{1}{2}\mathbf{x}^T L \mathbf{x} \\ \text{s.t.} & \mathbf{d}^T \mathbf{x} = 0 \\ & \mathbf{x}^T D \mathbf{x} = 1, \end{array} \tag{4.8}$$

where $D, \mathbf{d}$ can be replaced by $I, \mathbf{1}$, respectively, to minimize the node cut. Because of the last constraint, this is not a convex problem, but is easily solved as a [generalized] eigenvalue problem.

4.2 Adding Must-Link and Cannot-Link Constraints We wish to add a constraint to the above problem of the form $C\mathbf{x} = 0$, where C encodes *must-link* or *cannot-link* constraints. Each *must-link* constraint and cannot-link constraint are encoded in rows of C of the form

$$\begin{array}{ll} (0, \ldots, 0, -1, 0, \ldots, 0, +1, 0, \ldots, 0) & (\textit{must-link}) \\ (0, \ldots, 0, +1, 0, \ldots, 0, +1, 0, \ldots, 0) & (\textit{cannot-link}). \end{array}$$

C can be thought of as the incidence matrix (edges by vertices matrix) for the graph of *must-link* constraints, plus similar non-negative rows for the cannot-link constraints. C represent the $N_{constraints} \times |V|$ incidence matrix capturing the constraints in the graph.

In principle, if the constraints are all enforced completely, then we have to solve the problem

$$\begin{array}{ll} \min_{\mathbf{x}} & \frac{1}{2}\mathbf{x}^T L \mathbf{x} \\ \text{s.t.} & \mathbf{d}^T \mathbf{x} = 0 \\ & C\mathbf{x} = 0 \\ & \mathbf{x}^T D \mathbf{x} = 1. \end{array}$$

This could be solved as a generalized eigenvalue problem (as in [13]), but at considerable expense if the graph or number of constraints are large. However, we wish to handle the case where many of the constraints are somewhat uncertain or speculative due to noise or other factors, or where all the constraints together would overly distort the clustering. So we wish for a problem setup which would minimize the number of violated constraints while minimizing the usual relaxed normalized cut. This would lead to the following ideal optimization problem, where we have added a penalty term in the objective function to penalize violations of the constraints:

$$\begin{array}{ll} \min_{\mathbf{x},\mathbf{z}} & \frac{1}{2}\mathbf{x}^T L\mathbf{x} + \lambda\|\mathbf{z}\|_p \\ \text{s.t.} & \mathbf{d}^T\mathbf{x} = 0 \\ & C\mathbf{x} = \mathbf{z} \\ & \mathbf{x}^T D\mathbf{x} = 1, \end{array} \tag{4.9}$$

with $p = 0$ representing the "0-norm", the number of non-zero elements (a count). Instead, we relax this to a more tractable problem by using $p = 1$. Unlike [6], we choose $p = 1$ as opposed to $p = 2$ as a way to encourage sparsity in the penalty vector $\mathbf{z}$. Here λ is a user-selected weighting parameter. The parameter λ controls the degree to which the constraints are to be satisfied.

4.3 Convex Subproblem The optimization problem (4.9) is not convex, but we can formulate a convex subproblem that yields an approximate solution that can be improved by iterating the subproblem. Convex optimization is relatively easy to solve. There are a number of off-the-shelf convex optimization toolboxes available for such problems. We use the convex optimization toolbox CVX [16], which sufficed for our purposes. The subproblem is obtained by replacing the last (quadratic) constraint above with a linear local approximation and adding a penalty against large deviations:

$$\begin{array}{ll} \min_{\hat{\mathbf{x}},\hat{\mathbf{z}}} & \frac{1}{2}\hat{\mathbf{x}}^T L\hat{\mathbf{x}} + \mu\|\hat{\mathbf{x}} - \mathbf{x}\|_D^2 + \lambda\|\hat{\mathbf{z}}\|_1 \\ \text{s.t.} & \mathbf{d}^T\hat{\mathbf{x}} = 0 \\ & C\hat{\mathbf{x}} - \hat{\mathbf{z}} = 0 \\ & \mathbf{x}^T D\hat{\mathbf{x}} = 1, \end{array} \tag{4.10}$$

where $\mathbf{x}$ is the starting point for the subproblem, and μ is a damping factor, discouraging large movement away from the starting vector $\mathbf{x}$. Here $\|\mathbf{x}\|_D^2 = \mathbf{x}^T D\mathbf{x}$ is a weighted 2-norm.

The claim is that solving the subproblem (4.10) yields a descent step for (4.9), even after renormalizing the resulting $\hat{\mathbf{x}}_{\min}$ to satisfy the original quadratic constraint $\mathbf{x}^T D\mathbf{x} = 1$. The resulting iteration is given as Algorithm 1.

Theorem. Each pass through steps 2–5 of Algorithm 1 is a descent step for optimization (4.9).

Algorithm 1 CONSTRAINED CLUSTERING

Require: The graph Laplacian L, constraint incidence matrix C, scalars λ, μ.
Let x denote the indicator vector containing the cluster assignment.

Start with an initial $\mathbf{x}^{[0]}$

1. For $k = 0, 1, 2, \ldots$ until convergence
2. Solve (4.10) for $\hat{\mathbf{x}}_{\min}, \hat{\mathbf{z}}_{\min}$, starting with $\mathbf{x} = \mathbf{x}^{[k]}$
3. Set $\gamma = \|\hat{\mathbf{x}}_{\min}^T\|_D$
4. Set $\mathbf{x}^{[k+1]} = \hat{\mathbf{x}}_{\min}/\gamma$
5. Set $\mathbf{z}^{[k+1]} = \hat{\mathbf{z}}_{\min}/\gamma$

Return: $\mathbf{x}^{[\text{final}]}$

Proof.

I. As a solution to a minimization problem, we have

$$\begin{aligned} &\tfrac{1}{2}\hat{\mathbf{x}}_{\min}^T L\hat{\mathbf{x}}_{\min} + \mu\|\hat{\mathbf{x}}_{\min} - \mathbf{x}\|_D^2 + \lambda\|\hat{\mathbf{z}}_{\min}\|_1 \\ &\quad \le \tfrac{1}{2}\mathbf{x}^T L\mathbf{x} + \mu\|\mathbf{x} - \mathbf{x}\|_D^2 + \lambda\|\mathbf{z}\|_1, \end{aligned}$$

since $\mathbf{x}$ is feasible for (4.10). Hence (unless $\hat{\mathbf{x}}_{\min} = \mathbf{x}$)

$$\begin{aligned} &\tfrac{1}{2}\hat{\mathbf{x}}_{\min}^T L\hat{\mathbf{x}}_{\min} + \lambda\|\hat{\mathbf{z}}_{\min}\|_1 \le \tfrac{1}{2}\mathbf{x}^T L\mathbf{x} \\ &\quad + \lambda\|\mathbf{z}\|_1 - \mu\|\hat{\mathbf{x}}_{\min} - \mathbf{x}\|_D^2 < \tfrac{1}{2}\mathbf{x}^T L\mathbf{x} + \lambda\|\mathbf{z}\|_1. \end{aligned}$$

Hence step 2 reduces the objective value.

II. We show $\gamma > 1$ (unless $\hat{\mathbf{x}}_{\min} = \mathbf{x}$). We have that $(\hat{\mathbf{x}}_{\min} - \mathbf{x})^T D\mathbf{x} = \hat{\mathbf{x}}_{\min}^T D\mathbf{x} - \mathbf{x}^T D\mathbf{x} = 0$, since both $\hat{\mathbf{x}}_{\min}, \mathbf{x}$ satisfy the last constraint of (4.10). Hence $\|\hat{\mathbf{x}}\|_D^2 = \|\hat{\mathbf{x}} - \mathbf{x}\|_D^2 + \|\mathbf{x}\|_D^2 > \|\mathbf{x}\|_D^2 = 1$.

III. The objective value with $\mathbf{x}^{[k+1]}, \mathbf{z}^{[k+1]}$ is then $\frac{1}{2}(\mathbf{x}^{[k+1]})^T L\mathbf{x}^{[k+1]} + \lambda\|\mathbf{z}^{[k+1]}\|_1 = \frac{1}{2}\hat{\mathbf{x}}_{\min}^T L\hat{\mathbf{x}}_{\min}/\gamma^2 + \lambda\|\hat{\mathbf{z}}_{\min}\|_1/\gamma$ which is less than the starting objective value $\frac{1}{2}(\mathbf{x}^{[k]})^T L\mathbf{x}^{[k]} + \lambda\|\mathbf{z}^{[k]}\|_1$ by point I.

∎

4.4 Setting the parameters μ and λ The parameter λ controls the number of constraints satisfied. The problem of finding the right value of λ is the same in any of the regularization algorithms. In many cases the problem is solved with a variety of λs. As λ is increased, the constraints get tighter and tighter until a desired level of sparsity is reached. To simplify the selection in our cases, we selected λ from the range $[0.1, 10]$ to give the desired performance in terms of the constraints satisfied. The parameter μ controls the convergence and determines the number of iterations of the convex subproblem that are required. The larger the μ value, the more robust is the iteration at a cost of slower convergence. The optimal value should be in a range where the μ term does not dominate nor is dominated by the other terms.

Data	No of instances	No of attributes
Wine	119	13
Glass	146	9
Ionosphere	351	32
Hepatitis	155	19
WDBC	569	30
Diabetes	768	8

Table 1: UCI dataset.

In our experiments we found $\mu = 1$ worked well for all the data sets we tried, yielding convergence in 6-8 inner iterations at most.

5 Results

We considered two widely used measure of cluster evaluation, namely, cluster purity [17] and normalized mutual information (NMI) [18]. Cluster purity is measured by first assigning the dominant class label as the label for a cluster, so that purity is defined as:

$$Purity(\hat{\mathbf{x}}, \mathbf{y}) = \sum_k max_j \left\{ \frac{|c_k \cap l_j|}{|c_k|} \right\} \tag{5.11}$$

where $\hat{\mathbf{x}} = \{c_1, c_2, \ldots, c_K\}$ is the set of cluster assignments and $\mathbf{y} = \{l_1, l_2, \ldots, l_J\}$ is the set of true labels. One disadvantage of cluster purity is that it increases with the increase in the number of clusters, but this does not apply here since the number of clusters is fixed at 2. We also used another popularly used measure to evaluate cluster quality i.e. Normalized Mutual Information (NMI). NMI is defined as the mutual information between the cluster assignments ($\hat{\mathbf{x}}$) and the labeling of the dataset ($\mathbf{y}$) normalized by the arithmetic mean of the maximum possible entropies of the empirical marginals.

$$NMI(\hat{\mathbf{x}}, \mathbf{y}) = \frac{2 \cdot I(\hat{\mathbf{x}}, \mathbf{y})}{H(\hat{\mathbf{x}}) + H(\mathbf{y})} \tag{5.12}$$

An advantage of NMI is that it does not necessarily increase when the number of clusters increase. Both measures lie in the range $[0, 1]$ such that the higher the value, the better is the clustering quality.

5.1 UCI dataset We tested the model performance of our algorithm using six datasets from the UCI Repository [19], namely wine, glass, ionosphere, hepatitis, breast cancer and diabetes. The details of these datasets is shown in the Table 1. These datasets contain labels for the appropriate classes which the samples belong in. We first constructed a graph from these datasets by treating the nodes in the graph to be the sample points in the dataset and the edge weight to be the similarity between the features of the different samples. Edge weights were determined via the RBF kernel with σ^2 set to the variance present in the respective dataset:

$$A(i, j) = exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right) \tag{5.13}$$

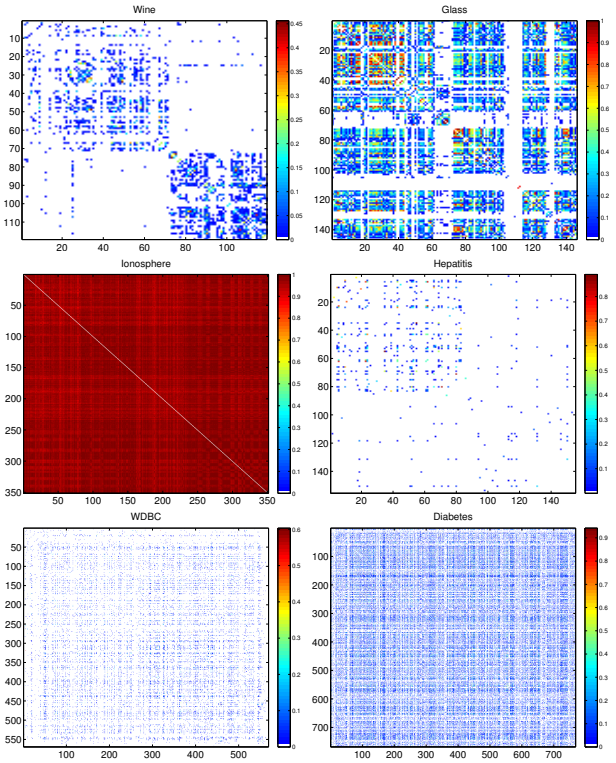


Figure 1: Graph created from six UCI datasets: Wine, Glass, Ionosphere, Hepatitis, WDBC, and Diabetes

Fig. 1 shows the graph affinity structure for the six datasets.

5.1.1 Model Performance For each of the UCI dataset, we compare our model with state of the art prior work [6]. We selected this method as it is the only method that can handle *cannot-link* constraints and hence is a natural predecessor to our work.

To compare the model performance, we randomly constructed the *must-link* and *cannot-link* constraints using a varying percentage of the given label information. In each case, we ran the algorithm 100 times and computed the mean performance of the models by measuring the cluster purity and NMI values. Fig. 3 shows the purity and NMI values for the different runs of the models using the UCI dataset. Both cluster purity and NMI effectively measure the satisfaction of constraints in the given datasets. When the number of known labels is in-

Dataset	α (baseline)	α (baseline)	α (baseline)	α (baseline)
	$0.6 \leq nmi < 0.7$	$0.7 \leq nmi < 0.8$	$0.8 \leq nmi < 0.9$	$nmi \geq 0.9$
Wine	-	56.56	88447	2.1624e+05
Glass	-	-	-	-
Ionosphere	0.62	0.82	0.9294	1.0325
Hepatitis	6.19e+07	5.36e+09	6.42e+09	6.424e+09
WDBC	-	1.99e+03	6.77e+14	6.16e+23
Diabetes	217	485.50	2.12e+03	2.28e+03

Table 2: Parameter value for the baseline method for a fixed NMI. The λ of our model lie in the range $[0.1, 10]$ to achieve the same performance as baseline.. The blanks indicate that the required NMI value was not reached.

Dataset	α (baseline)	α (baseline)	α (baseline)	α (baseline)
	$\%known = 20$	$\%known = 40$	$\%known = 60$	$\%known = 80$
Wine	1.85e+02	1.33e+03	8.85e+04	3.03e+05
Glass	3.27e+06	3.27e+06	3.29e+06	4.34e+34
Ionosphere	0.20	0.41	0.61	0.82
Hepatitis	6.19e+07	1.20e+05	6.36e+09	6.424e+09
WDBC	6.77e+14	1.53e+14	6.16e+23	6.16e+23
Diabetes	279	346	2.12e+03	2.28e+03

Table 3: Parameter value for the baseline method for a fixed % of known labels. The λ of our model lie in the range $[0.1, 10]$ to achieve the same performance as baseline.

creased the performance of both the models increase. From the results, we see that our approach consistently matches the performance of the prior state-of-art model. In the case of Glass dataset, it beats the baseline approach. Overall the result indicates that our model is consistent with the prior state-of-art model.

For some datasets, we see that the NMI is poor only when a few constraints are imposed. This is because the natural clustering of the data set leads to a poor partitioning. We see this trend in the following datasets: (1) glass, (2) ionosphere, (3) hepatitis, and (4) diabetes. In this cases, the only way to get a correspondence to the labels is by applying supervision in the form of constraints, and in some cases, a large fraction of these constraints are needed. When the NMI value is relatively high even with the application of a few constraints (like in wine and wdbc dataset), then the natural clustering in the graph is well correlated with the underlying labels. As we can see from the Fig. 1, the wine dataset has a relatively well defined 2-class cluster structure as compared to the other datasets. Hence, as expected, both the algorithms give very good performance on the Wine dataset. In the glass dataset, there is a poor correspondence between the natural clusterings and the labels and some of the sample points do not show a high similarity with any other points. *Our method was able to take advantage of the prior knowledge in the form of constraints especially when enough of them are provided, whereas the baseline failed in these case.*

5.1.2 Parameter Value Here, we investigate the quality of the partitions proposed by the two approaches. To do this, we compare the parameter values of the two methods. The λ parameter in our approach and the α parameter in the baseline method (Eq. 3.7) controls the trade-off between the normalized cut partition on the original graph and the satisfaction of constraints. In the baseline method, the value of α is set to the largest eigenvalue of the normalized matrix Q.

Fixed NMI. In order to compare the two values, we first fixed the NMI and varied the percentage of known labels. Then we measure the parameter values that are required to achieve the fixed NMI. *The λ of our method, was found to be in the range $[0.1, 10]$ for all the datasets and for all NMI values.* The value of α for the baseline method is shown in Table 2. We observe here that λ is much smaller than α in all cases. The value of α goes as high as 10^{23}. A high value of α means that the baseline method was focused more on satisfying the constraints rather than maintaining the quality of the partitions. Only in the case of the Ionosphere dataset, the value of α is close to λ. This is because the graph from the

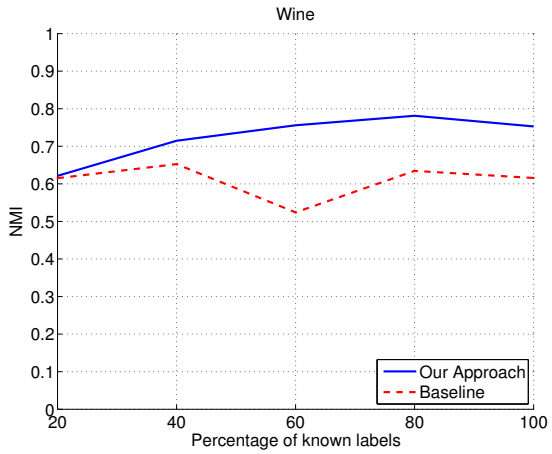


Figure 2: Effect of adding 10% noisy constraints to NMI on the Wine dataset. The experiment is run 100 times and graph plots average NMI for the two methods.

Ionosphere dataset is particularly densely connected compared to the other graphs (Fig. 1). Overall the result indicates that our method provides a good balance between constraint satisfaction and maintaining cluster quality, whereas the high parameter value of baseline indicates its overfitting on the constraints.

Fixed % labels. Next, we fix the percentage of known labels and compared the parameter values of the two methods. Table 3 shows the value of α of the baseline method using a fixed percentage of the known labels. Again λ of our approach was found to be in the range $[0.1, 10]$. The baseline required very large α values to satisfy equivalent percentage of constraints in comparison to our approach. The higher parameter values indicate that the baseline approach would have problems in cases when the constraints are noisy.

Noisy Constraints. Here we see the impact on the average NMI when we add noisy constraints to the dataset. For every scenario, we randomly flipped the value of 10% of the known labels to introduce observation noise in the constraints. Fig. 2 shows that our approach is more robust to noisy constraints as compared to the baseline method for the Wine dataset where both the methods yield close to perfect clustering in the non-noisy case. This results highlights the effectiveness of our model over the baseline method in real-world datasets where the constraints can be noisy.

5.2 Co-cluster Dataset We further show that our method can be extended to find clusters in constrained spectral co-clustering. Spectral co-clustering is used to find clusters in a bipartite graph [20] and is shown to perform better that clustering for various scenarios. For co-clustering, we use the `Classic3` dataset provided by the SMART project at Cornell University [21]. This dataset can be freely downloaded at ftp://ftp.cs.cornell.edu/pub/smart. We use a subset of the dataset containing the two classes `Medline` (200 medical abstracts) and `Cranfield` (200 aeronautical systems abstract) and there are 3141 unique words in the dataset. The nodes of our bipartite graph consist of documents and words, respectively and the edge weight represents the TF/IDF score of the word in the document. The details of the dataset are shown in the Table 4. Our goal is to find co-cluster partitions of the documents and the words.

Data	No of documents	No of edges
`Medline`	200	10510
`Cranfield`	200	10210
Total	400	20720

Table 4: Co-clustering dataset.

5.2.1 Model Performance In order to compare the performance of our method, we use the constrained spectral co-clustering approach by Shi et al. [8]. This is the only prior work that handles constraints in Spectral Co-clustering. The main idea behind the approach is to overlay the constraint graph on the original graph and minimize the normalized cut of the resulting graph Laplacian. This approach can only handle *must-link* constraints whereas we can handle both type of constraints. We randomly set some percentage of the known labels as constraints just like we did in the previous section. Note that for co-clustering, we can define constraints between the rows or columns or both. Since we know the labels of the documents, we define the constraints between them. We run both our approach and the baseline method 100 times. Fig. 4 shows the mean cluster purity and NMI values using the two approaches. From the figure, we see that our approach performs much better in comparison to the baseline. Overall it indicates that our approach beats the prior approach for spectral co-clustering.

6 Conclusion

In this paper, we present a novel approach to add *must-link* and *cannot-link* constraints to spectral clustering. Our approach proposes the objective function that can be formulated as convex subproblems and hence can be solved easily using some of the known solvers. We use the L_1 regularization to control the number of captured constraints and show that this allows us to effectively control the quality

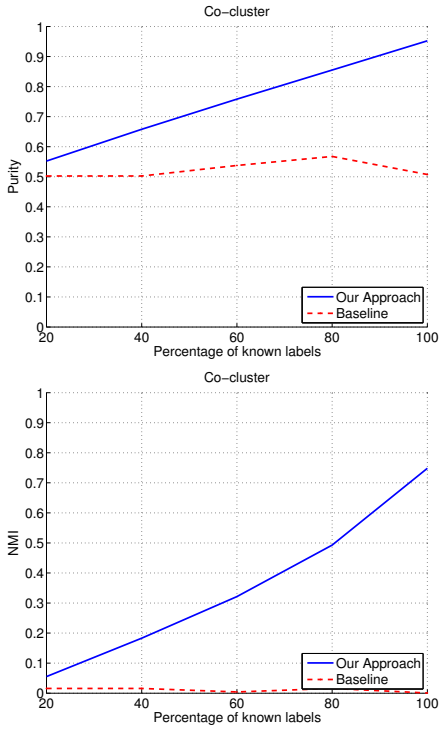


Figure 4: Model performance for constrained spectral co-clustering.

of the partitions in the underlying graph while satisfying many of the constraints. We evaluate our approach using several datasets and show that we can match the performance of the baseline method while still having a lower value of the parameters. Furthermore, we show that our approach is robust to the addition of noisy labels, and that our approach can be applied to spectral co-clustering in order to find partitions in the bipartite graph using a real world dataset.

As part of future work, we would like to extend our algorithm to handle more than two clusters. One simple approach is to hierarchically run the algorithm to generate two partitions in each of the cluster detected. However, there are further research questions like when should we stop looking for hierarchical partitions. Additionally, the choice of λ is domain dependent, since the usefulness of any particular clustering depends on the needs of specific applications. In future we can adopt fast methods to track the solution as λ varies.

References

[1] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *ICML*, pages 577–584, 2001.

[2] Sugato Basu, Ian Davidson, and Kiri Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall, 2008.

[3] Sepandar D. Kamvar, Dan Klein, and Christopher D. Manning. Spectral learning. In *IJCAI*, pages 561–566, 2003.

[4] Qianjun Xu, Marie desJardins, and Kiri Wagstaff. Active constrained clustering by examining spectral eigenvectors. In *Discovery Science*, pages 294–307, 2005.

[5] Xiang Ji and Wei Xu. Document clustering with prior knowledge. In *SIGIR*, pages 405–412, 2006.

[6] Xiang Wang and Ian Davidson. Flexible constrained spectral clustering. In *KDD*, pages 563–572, 2010.

[7] Zhengdong Lu and Miguel Á. Carreira-Perpiñán. Constrained spectral clustering through affinity propagation. In *CVPR*, 2008.

[8] Xiaoxiao Shi, Wei Fan, and Philip S. Yu. Efficient semi-supervised spectral co-clustering with constraints. In *ICDM*, pages 1043–1048, 2010.

[9] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[10] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[11] Stella X. Yu and Jianbo Shi. Grouping with bias. In *NIPS*, pages 1327–1334, 2001.

[12] Fei Wang, Chris H. Q. Ding, and Tao Li. Integrated kl (k-means - laplacian) clustering: A new clustering approach by combining attribute data and pairwise relations. In *SDM*, pages 38–48, 2009.

[13] Tijl De Bie, Johan A. K. Suykens, and Bart De Moor. Learning from general label constraints. In *SSPR/SPR*, pages 671–679, 2004.

[14] Tom Coleman, James Saunderson, and Anthony Wirth. Spectral clustering with inconsistent advice. In *ICML*, pages 152–159, 2008.

[15] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[16] M. Grant, S. Boyd, and Y. Ye. Cvx: Matlab software for disciplined convex programming. *Online accessiable: http://stanford. edu/˜ boyd/cvx*, 2008.

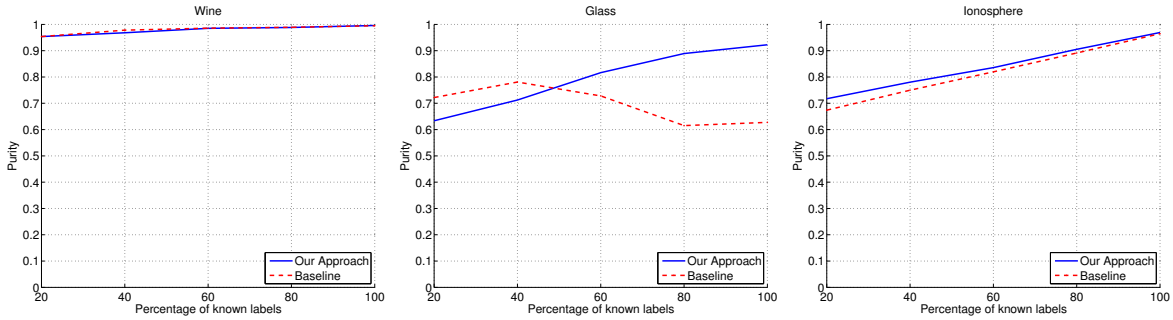
[17] Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.

[18] Shi Zhong and Joydeep Ghosh. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3):374–384, 2005.

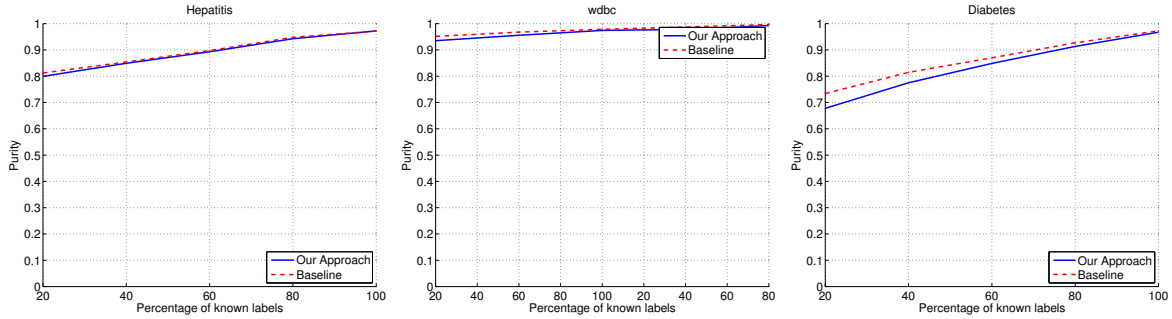
[19] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[20] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, pages 269–274, 2001.

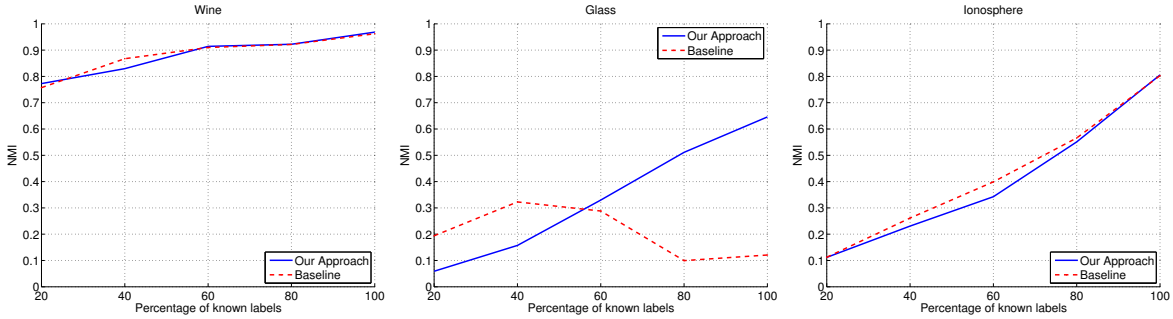
[21] G. Salton. The smart document retrieval project. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 356–358. ACM, 1991.



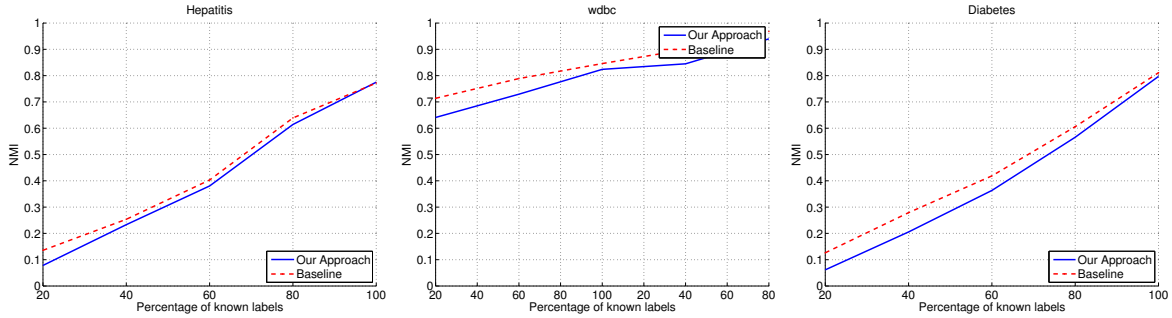
(a) Cluster Purity



(b) Cluster Purity



(c) NMI



(d) NMI

Figure 3: Comparison of model performance over the UCI datasets