Fast Gaussian Pairwise Constrained Spectral Clustering[*]

David Chatel[1], Pascal Denis[1], and Marc Tommasi[1,2]

[1] INRIA Lille
[2] Lille University

Abstract. We consider the problem of spectral clustering with partial supervision in the form of must-link and cannot-link constraints. Such pairwise constraints are common in problems like coreference resolution in natural language processing. The approach developed in this paper is to learn a new representation space for the data together with a distance in this new space. The representation space is obtained through a constraint-driven linear transformation of a spectral embedding of the data. Constraints are expressed with a Gaussian function that locally reweights the similarities in the projected space. A global, non-convex optimization objective is then derived and the model is learned via gradient descent techniques. Our algorithm is evaluated on standard datasets and compared with state of the art algorithms, like [14,18,31]. Results on these datasets, as well on the CoNLL-2012 coreference resolution shared task dataset, show that our algorithm significantly outperforms related approaches and is also much more scalable.

1   Introduction

Clustering is the task of mapping a set of points into groups (or "clusters") in such a way that points which are assigned to the same group are more similar to each others than they are to points assigned to other groups. Clustering algorithms have a large range of applications in data mining and related fields, from exploratory data analysis to well-known partitioning problems like noun phrase coreference resolution to more recent problems like community detection in social networks.

Over the recent years, various approaches to clustering have relied on spectral decomposition of the graph representing the data, whether the data inherently come in the form of a graph (e.g., a social network) or the graph is derived from the data (e.g., a similarity graph between data vectors). One way to understand spectral clustering is to view it as a continuous relaxation of the NP-complete normalized- or ratio-cut problems [28,22,21]. Spectral clustering has important advantages over previous approaches like k-means, one being that it does not

---

make strong assumptions on the shape (e.g., convexity) of the underlying clusters. Spectral clustering first consists in computing the first k eigenvectors associated with the smallest eigenvalues of the graph Laplacian. Discrete partitions are then obtained by running k-means on the space spanned by these eigenvectors. This leads to approximations of different optimal cuts of the graphs, which are known to be potentially quite loose [10,11]. Spectral clustering can also be understood in terms of the spectral embedding of the graph, the change of representation of the data represented by nodes. Indeed, the spectral decomposition of the graph Laplacian gives a projection of the data in a new feature space in which Euclidean distance corresponds to a similarity given by the graph (e.g., the resistance distance [15,27]).

In practice, it is often the case that the space spanned by the first k eigenvectors is not rich enough to single out the correct partition. Running k-means in a transformation of this space may yield a better partition than the one found in the original space. We propose to exploit pairwise constraints to guide the process of finding such a transformation. From this perspective, our work builds upon and extends previous attempts at incorporating constraints in spectral clustering [30,16,34,14,5,19,32,18,32,26]. While clustering is often performed in a unsupervised way, there are many situations in which some form of supervision is available or can easily be acquired. For instance, part of the domain knowledge in natural language processing problems, like noun phrase coreference resolution, naturally translates into constraints. For instance, gender and number mismatches between noun phrases (e.g., *Bill Clinton* vs. *she/they*) give strong indication that these noun phrases should not appear in the same cluster.

In this paper, we consider the setting wherein supervision is only partial, which is arguably more realistic setting when annotation is costly. Partial supervision takes the form of *pairwise constraints*, whereby two points are assigned to identical (*must-link*) or different clusters (*cannot-link*), irrespective of the clusters labels. All must-link constraints can be satisfied in polynomial time using a simple transitive closure. In some problems, constraints may be inconsistent, due to noisy preprocessing of the data for instance, and satisfying all cannot-link constraints is NP-complete for $k > 2$, see [7]. These constraints can contradict the unconstrained cuts of the graph. For example, two nodes close in graph could be constrained as cannot-link and conversely two nodes far away in the graph could be constrained as must-link. One open research question is how does one best integrate this type of partial supervision into the clustering algorithm.

In this paper, we propose to learn a linear transformation $\mathbf{X}$ of the spectral embedding of the graph with the partial supervision given by the constraints. Our algorithm also learns a similarity in order to find a partition such that similar nodes are in the same cluster, dissimilar nodes are in different clusters, and the maximum number of pairwise constraints are satisfied. When two nodes must link (respectively cannot link), their similarity is constrained to be close to 1 (respectively close to 0). In the learning step, the similarity is locally distorted around constrained nodes using a Gaussian function applied on the Euclidean distance in the feature space obtained by $\mathbf{X}$. In order to increase the gap between
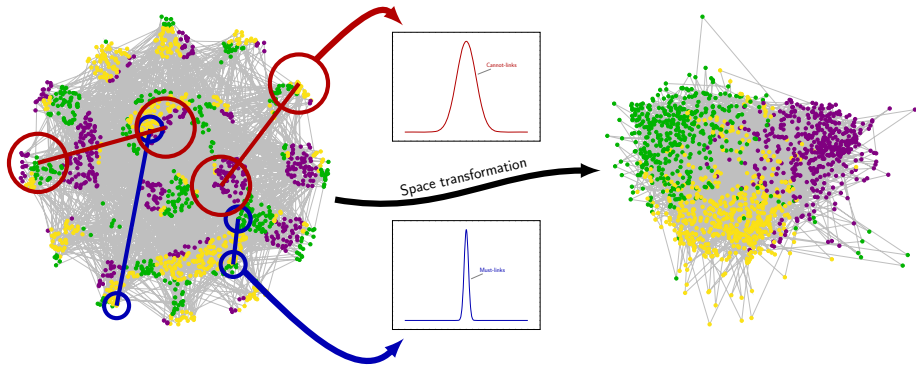
Fig. 1. This figure shows intuitively the process behind FGPWC. From a spectral embedding of a graph, Gaussian functions distort the distance between constrained pairs of nodes such that it become smaller or larger depending depending on the quality (must-link or cannot-link) attributed to the constraint. Gaussian functions act as a new similarity for the pair of nodes and it should be close to 1 if the pair must link and close to 0 if the pair cannot link.

must-link and cannot-link constraints, we use two Gaussian functions of different variances. As illustrated in Figure 1, this technique ensures that the distance in the new feature space between nodes in cannot-link constraints is significantly larger than the distance between nodes that must link. From this modeling, we derive a non-convex optimization problem to learn the transformation $\mathbf{X}$. We solve this problem using a gradient descent approach with an initialization for $\mathbf{X}$ that coincides with the unconstrained solution of the problem.

Our algorithm, FGPWC (for Fast Gaussian PairWise Clustering), is evaluated empirically on a large variety of datasets, corresponding either to genuine network data or to vectorial data converted into graphs. Two sets of experiments are conducted: the first one involves classification task, using commonly used data sets in the field. Empirical results place our algorithm above competing systems on most of the data sets. The second one involves a real task in the field of natural language processing: namely, the noun phrase coreference resolution task as described in the CoNLL-2012 shared task [25]. Our results show our algorithm compares favorably with the unconstrained spectral clustering approach of [6], outperforming it on medium-size and large clusters.

2   Background and Notation

Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ be an undirected connected graph with node set $\mathcal{V} = \{v_1, \ldots, v_n\}$, edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and non-negative similarity matrix $\mathbf{W}$, such that $\mathbf{W}_{ij}$ is the weight on the edge (v_i, v_j). Let $(\lambda_1, \boldsymbol{u}_1), \ldots, (\lambda_n, \boldsymbol{u}_n)$ be eigenvalue/eigenvectors pairs of the graph Laplacian $\mathbf{L_{sym}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$,

such that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. The matrix $\mathbf{U} = \left(\sqrt{\frac{1}{\lambda_1}} \boldsymbol{u}_1 \ \sqrt{\frac{1}{\lambda_2}} \boldsymbol{u}_2 \cdots \ \sqrt{\frac{1}{\lambda_n}} \boldsymbol{u}_n \right)$ is a spectral embedding of the graph. It can be thought as an Euclidean feature space where each node v_i is represented by a data point whose coordinates in this space are the components of the vector $\boldsymbol{v}_i$ equal to the ith row of the matrix $\mathbf{U}$. The first eigenvector $\boldsymbol{u}_1$ is the constant vector $\mathbb{1}$ biased by the degrees of the nodes, $\boldsymbol{u}_1 = \mathbf{D}^{1/2} \mathbb{1}$ and can be dropped from the feature space, as it does not provide any information for characterizing nodes. Eigenvectors $\boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$ are functions that map the manifold of the graph to real lines. If $\boldsymbol{f}$ is such a function, then $\boldsymbol{f}^\top \mathbf{L_{sym}} \boldsymbol{f} = \frac{1}{2} \sum_{i,j=1}^n \mathbf{W}_{ij} (\boldsymbol{f}_i - \boldsymbol{f}_j)^2$ provides an estimate of how far nearby points will be mapped by $\boldsymbol{f}$ [3]. As m increases, the space spanned by $\boldsymbol{u}_2, \ldots, \boldsymbol{u}_m$ with $m n$ will describe smaller and smaller details in the data. In the following, we consider a spectral embedding $\mathbf{V}_m = \left(\boldsymbol{u}_2 \ldots \ \boldsymbol{u}_m \right) = \left(\boldsymbol{v}_1 \ldots \boldsymbol{v}_n \right)^\top$. To each each node of the graph v_i correspond a vector $\boldsymbol{v}_i$ that lives in this space.

Pairwise constraints are defined as follows. Let $\mathcal{M}, \mathcal{C} \subset \mathcal{V} \times \mathcal{V}$ be two sets of pairs of nodes, describing must-link and cannot-link constraints. Let K be the total number of constraints. If $(v_i, v_j) \in \mathcal{M}$, then v_i and v_j should be in the same cluster, and if $(v_i, v_j) \in \mathcal{C}$ then v_i and v_j should be in different clusters. We introduce the $K \times m$ matrices $\mathbf{A}$, $\mathbf{B}$ and the K-dimensional vector $\boldsymbol{q}$:

$$\mathbf{A} = \begin{pmatrix} \boldsymbol{v}_{i_1} \\ \vdots \\ \boldsymbol{v}_{i_K} \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} \boldsymbol{v}_{j_1} \\ \vdots \\ \boldsymbol{v}_{j_K} \end{pmatrix} \qquad \boldsymbol{q}_k = \begin{cases} 1 & \text{if } (v_{i_k}, v_{j_k}) \in \mathcal{M} \\ 0 & \text{if } (v_{i_k}, v_{j_k}) \in \mathcal{C} \end{cases}$$

where $(\boldsymbol{v}_{i_k}, \boldsymbol{v}_{j_k})$ are vectors describing the kth pair of nodes (v_{i_k}, v_{j_k}) in $\mathcal{M} \cup \mathcal{C}$.

3   Problem Formulation

We propose to learn a linear transformation ϕ of the feature space $\mathbf{V}_m$ that best satisfies the constraints. Let $\phi(\boldsymbol{v}_i) = \boldsymbol{v}_i \mathbf{X}$ where $\mathbf{X}$ is a $m \times m$ matrix describing the transformation of the space. We want to find a projection of the feature space $\phi(\boldsymbol{v}_i)$ such that the clusters are dense and far away from each other. Ideally, if nodes $(v_i, v_j) \in \mathcal{M}$ then the distance between $\phi(\boldsymbol{v}_i)$ and $\phi(\boldsymbol{v}_j)$ should equal zero and if nodes $(v_i, v_j) \in \mathcal{C}$ then the distance between $\phi(\boldsymbol{v}_i)$ and $\phi(\boldsymbol{v}_j)$ should be very large. We introduce two Gaussian functions to locally distort the similarities for constrained pairs. Gaussian parameters σ_m and σ_c are chosen such that $\sigma_m \leq \sigma_c$. The similarity between two nodes v_i and v_j is $\exp^{-\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 / \sigma_m}$ if $(v_i, v_j) \in \mathcal{M}$ and $\exp^{-\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 / \sigma_c}$ if $(v_i, v_j) \in \mathcal{C}$ where $\|\cdot\|$ is the Frobenius norm. Therefore, we want to ensure that $\mathbf{X}$ is such that $\exp^{-\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 / \sigma_m}$ is close to 1 if $(v_i, v_j) \in \mathcal{M}$ and $\exp^{-\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 / \sigma_c}$ is close to 0 if $(v_i, v_j) \in \mathcal{C}$. We now encode the set of all constraints in a matrix form. Let us first consider the K-dimensional vector $\boldsymbol{\sigma} \in \{1/\sigma_m, 1/\sigma_c\}^K$.

Let $\mathbb{1}$ be the m-dimensional vector of all ones. Notice that $[(\mathbf{A} - \mathbf{B})\mathbf{X}]^2 \mathbb{1}$, is the vector whose components are equal to the distance between pairs of constrained nodes in the transformed space. Let $\circ$ be the Hadamard product. Then

(a) No regularization   (b) Regularized
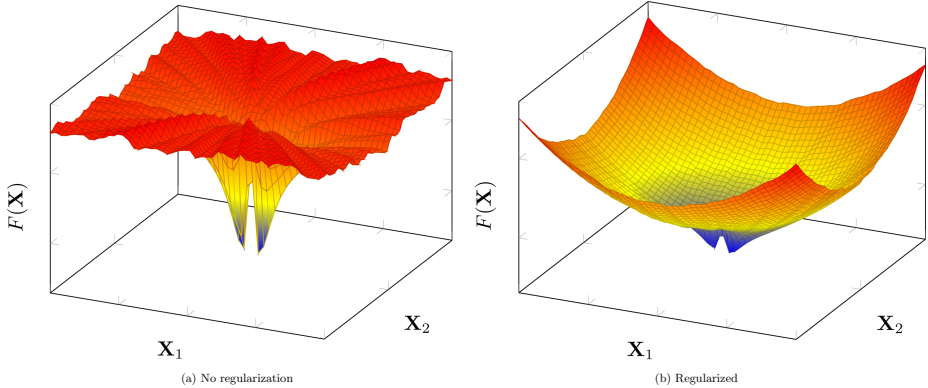
Fig. 2. Normalization effect on a simple example. 900 data points in $\mathbb{R}^2$ were drawn using a normal distribution $\mathcal{N}(0,1)$. Only 1‰ must and cannot-links have been uniformly drawn to separate data in two groups of positive and negative points. These figures plot $F(\mathbf{X})$ in the neighborhood of $\mathbf{X}^\star$. The two dimensions of $\mathbf{X}^\star$ in this example are referred by $\mathbf{X}_1$ and $\mathbf{X}_2$.

$\exp^{-[(\mathbf{A}-\mathbf{B})\mathbf{X}]^2\mathbb{1}\circ\boldsymbol{\sigma}}$ is the vector whose components equal the corresponding must-link or cannot-link similarity depending on whether the associated pairs of nodes are in $\mathcal{M}$ or $\mathcal{C}$. The values in $\mathbf{X}$ are not bounded in this expression. So, we propose to add a regularization term on $\mathbf{X}$. This gives the optimization problem:

$$\min_{\mathbf{X}} F(\mathbf{X}) = \left\| \exp^{-[(\mathbf{A}-\mathbf{B})\mathbf{X}]^2\mathbb{1}\circ\boldsymbol{\sigma}} -\boldsymbol{q} \right\|^2 + \gamma \|\mathbf{X}\|^2 \tag{1}$$

The effect of this regularization step is depicted in Figures 2a and 2b. In this toy example, data points where drawn using a normal distribution with mean 0. Constraints are added in order to separate positive and negative points in two clusters. Only 1‰ must and cannot-links have been uniformly drawn. We can see that in both non regularized and regularized cases, global optimums are identical. However, Figure 2a shows that far away from the global optimum, the non regularized objective function is not smooth. The regularization handles this issue, see figure 2b.

3.1 Algorithm

Our algorithm for learning the transformation $\mathbf{X}$ is presented in Algorithm 1. It takes as input a weighted adjacency matrix of a graph, and two matrices for must-link and cannot-link constraints. Parameters are the number k of clusters as usual in k-MEANS, but also the widths of the Gaussian functions σ_m and σ_c and the dimension m of $\mathbf{X}$.

The target dimension m is related to the amount of contradiction between the graph and the constraints. Remember that eigenvectors of $\mathbf{L_{sym}}$ are functions

which maps nodes from the manifold of the graph to real lines and the associated eigenvalues provides us with an estimate of how far apart these functions maps nearby points [3]. When the pairwise constraints do not contradict the manifold of the graph, i.e. must-link pairs are already close on the manifold and cannot-link pairs are already far apart, m does not need to be large, because the eigenvectors associated with smallest eigenvalues will provide eigenmaps which do not contradict the constraints. Hence, a solution can be found in the very first eigenvectors. However, when the pairwise constraints contradict the manifold of the graph: must-links that are initially far apart on the manifold or cannot-links that are close, we need to consider a larger number of eigenvectors m, because the eigenvectors providing the eigenmaps that will not contradict the constraints will be later dimensions of the embedded space, describing smaller details.

Our algorithm is a typical gradient descent and its initialization can be at random. However, we propose to initialize it close to unconstrained spectral clustering $\mathbf{X}_0 = (\mathbf{V}_m^\top \mathbf{L_{sym}} \mathbf{V}_m)^{-1/2}$. We stop the descent after imax iterations or when the Frobenius norm of the partial derivative $\frac{\partial F(\mathbf{X})}{\partial \mathbf{X}}$ is less than ϵ.

Algorithm 1. FGPWC

Input: $\mathbf{W} \in \mathbb{R}^{n \times n}, \mathbf{M} \in \mathbb{R}^{n \times n}, \mathbf{C} \in \mathbb{R}^{n \times n}, m, k, \sigma_m, \sigma_c$
Output: $\mathbf{X}^\star \in \mathbb{R}^{m \times d}, \mathcal{P}$ partition of $\mathcal{V}$

1 **begin**
2     $\mathbf{L_{sym}} \leftarrow \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$
3     $\mathbf{V}_m \leftarrow$ first m smallest eigenvectors of $\mathbf{L_{sym}}$
4     $\mathbf{X}_0 \leftarrow (\mathbf{V}_m^\top \mathbf{L_{sym}} \mathbf{V}_m)^{-1/2}$
5     $i \leftarrow 0, \alpha \leftarrow 1$
6   **repeat**
7         $i \leftarrow i + 1, \mathbf{X}_i \leftarrow \mathbf{X}_{i-1} - \alpha \partial F(\mathbf{X}_{i-1})/\partial \mathbf{X}$
8   **if** $F(\mathbf{X}_i) >= F(\mathbf{X}_{i-1})$ **then**
9             $\alpha \leftarrow \alpha/2$
10   **else**
11            $\mathbf{X}^\star \leftarrow \mathbf{X}_i$
12   **until** $\|\partial F(\mathbf{X}_i)/\partial \mathbf{X}\|^2 < \epsilon$ **or** $i > $ imax
13    $\mathcal{P} \leftarrow k\text{-MEANS}(\mathbf{V}_m \mathbf{X}, k)$
14   **return** $\mathcal{P}$

---

4   Related Work

The use of supervision in clustering tasks has been addressed in many ways. A first related approach is that of [33], which is inspired by distance learning. Constraints are given through a set of data point pairs that should be close. The authors then consider the problem of learning a weighted matrix of similarities.

They derive an optimization problem of high complexity, which they solve by doing alternate gradient ascent on two objectives, one bringing closer points that are similar and the other putting off the other points. Similarly, in [13] learning spectral clustering is the problem of finding weighted matrix or the spectrum of the Gram matrix given a known partition. A related field is supervised clustering [9], the problem of training a clustering algorithm to produce desirable clusterings: given sets of items and complete clusterings over these sets, we learn how to cluster future sets of items.

Another set of related approaches are constrained versions of the k-means clustering algorithm. In [30], it is proposed that, at each step of the algorithm, each point is assigned to the closest centroid provided that must-link and cannot-link constraints are not violated. It is not clear how the choice of the ordering on points affects the clustering. Moreover, constraints are considered as hard constraints which makes the approach prone to noise effects. Kulis *et al* improve on the work of [30] in [16]. Their algorithm relies on weighted kernel k-means ([8]). The authors build a kernel matrix $\mathbf{K} = \sigma\mathbf{I} + \mathbf{W} + \mathbf{S}$, where $\mathbf{W}$ is a similarity matrix, $\mathbf{S}$ is a supervision matrix such that $\mathbf{S}_{ij}$ is positive (respectively negative) when nodes i and j must link (respectively cannot link) or zero when unconstrained. The addition of $\sigma\mathbf{I}$ ensures the positive semi-definiteness of $\mathbf{K}$ (otherwise, $\mathbf{K}$ would not be a kernel, would not have any latent Euclidean space, a requirement for k-means to converge and for theoretical justification).

Introducing constraints in spectral clustering has received a lot of attention in the last decade ([34,14,5,19,32]). In many cases, the proposed approaches rely on a modification of the similarity matrix and then the resolution of the associated approximated normalized cut. For instance, in [14], weights in the similarity matrix are forced to 0 or 1 following must-link and cannot-link constraints. But this kind of weights may have a limited impact on the result of the clustering, in particular when the considered two nodes have many paths that link them together. [34] consider three kinds of constraint and cast them into an optimization problem including membership constraints in a 2-partitioning graph problem. To guarantee a smooth solution, they reformulate the optimization problem so that it involves computing the eigen decomposition of the graph Laplacian associated with the data. The approach relies on an optimization procedure that includes nullity of the flow from labeled nodes in cluster 1, to labeled nodes in cluster 2. The algorithm closely resembles the semi-supervised harmonic Laplacian approach developed for instance in [35]. But this approach is also limited to the binary case. In [19], pairwise constraints are used to propagate affinity information to the other edges in the graph. A closed form of the optimal similarity matrix can be computed but its computation requires one matrix inversion per cannot-link constraint.

In [18], constrained clustering is done by learning a transformation of the spectral embedding into another space defined by a kernel. The algorithm attempts to project data points representing nodes onto the bound of a unit-hypersphere. The inner product between vectors describing nodes that must link is close to 0, and the inner product between vectors describing nodes that cannot-link is close

to 1. That way, if a node v_i belongs to the cluster j, then the vector $\boldsymbol{v}_i$ describing v_i will be projected to $\mathbb{1}_j$ where e_j is a vector of length k full of zeros except on the jth component where it is equal to 1. The number of dimensions of the hypersphere is directly related to the ability to separate clusters. One drawback is that the algorithm uses semidefinite programs whose size is quadratic in that number of dimensions.

Recently, [31,32] propose to include constraints by modifying directly the optimization problem rather than modifying the Laplacian. In their algorithm called CSP, they introduce a matrix $\mathbf{Q}$ where $\mathbf{Q}_{ij}$ is 1 if i and j must-link, -1 if i and j cannot-link and 0 otherwise. Then, a constraint $\mathbf{f}^\top \mathbf{Q} \mathbf{f} > \alpha$ is added to the normalized cut objective considered in unconstrained spectral clustering. Parameter α is considered as a way to soften constraints. Their approach outperforms previous approaches such as the one based on kernel k-means defined in [16]. An original approach based on tight relaxation of graph cut ([11]) is presented in [26]. The approach deals with must and cannot-links but in the two clusters case. It guarantees that no constraints are violated as long as they are consistent. For problems with more than two clusters, hierarchical clustering is proposed. Unfortunately in this case, the algorithm loses most of its theoretical guarantees.

5   Experiments

We conducted two sets of experiments. In the first experiments, we evaluate our algorithm on a variety of well-known clustering and classification datasets, and compare it to four related constrained clustering approaches: CCSR [18], SL [14], CSP [32] and COSC [26]. CCSR also seeks a projection of space in which constraints are satisfied. SL modifies the adjacency matrix and puts 0 for cannot-link pairs and 1 for must-link pairs. CSP modifies the minimum cut objective function introducing a term for solving a part of the constraints. COSC is based on a tight relaxation of the constrained normalized cut into a continuous optimization problem.

In a second set of experiments, we apply our algorithm to the problem of noun phrase coreference resolution, a very important problem in Natural Language Processing. The task consists in determining for a given text which noun phrases (e.g., proper names, pronouns) refer to the same real-world entity (e.g., Bill Clinton). This problem can be easily recast as a (hyper-)graph partitioning problem [24,6]. We evaluate our algorithm on the CoNLL-2012 English dataset and compare it to the unconstrained spectral clustering approach of [6], a system that ranked among the top 3 systems taking part in the CoNLL-2012 shared task.

5.1   Clustering on UCI and Network Data Sets

Dataset and Preprocessing. We first consider graphs built from UCI datasets and networks. Table 1 summarizes their properties and the characteristics of the associated clustering problem. Graph construction uses a distance that is

computed based on features. First, continuous features are normalized between 0 and 1 and nominal features are converted into binary features. Second, given feature vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ associated with two datapoints, we consider two kinds of similarities: either RBF kernels of the form $\exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2 / 2\sigma^2)$ or cosine similarity $\boldsymbol{x} \cdot \boldsymbol{x}' / (\|\boldsymbol{x}\| \times \|\boldsymbol{x}'\|)$. In the case of cosine similarity we also apply k-NN and weight edges with similarity. For instance, from the imdb movie dataset we extract records in which Brad Pitt, Harrison Ford, Robert De Niro and Sylvester Stallone have played. The task is to determine which of the four actors played in which movie. The movies in which more than one of these actors have played are not part of the dataset so that classes do not overlap. We have collected all the actors (except for the four actors that serve as classes) who played in 1606 movies. Each movie is described by binary features representing the presence or absence of an actor in its cast. The similarity measure between movies is the cosine similarity.

Evaluation Metric. We use Adjusted Rand Index (ARI) [12] as our main evaluation measure. The standard Rand Index compares two clusterings by counting correctly classified pairs of elements. It is defined as: $\mathcal{R}(\mathcal{C}, \mathcal{C}') = \frac{2(TP+TN)}{n(n-1)}$ where n is the number of nodes in the graph and *TP*, *TN* are true positive and true negative pairs. By contrast, the Adjusted Rand Index which is the normalized difference of the Rand Index and its expected value under the null hypothesis. This index has an expected value of zero for independant clusterings and maximum value 1 for identical clusterings. We report the mean over the 10 runs corresponding to 10 sets of constraints of the ARI computed against the ground truth. As an additional measure, we also report the number of violated constraints in the computed partition and the computation time for each algorithm.

System Settings. For each dataset, 10 different sets of constraints were selected at random. The number of constraints is chosen to avoid trivial solutions. Indeed, if the number of must-link constraints is high, a transitive closure quickly gives a perfect solution. So, the interesting cases are when only a few number of constraints is considered. Given a graph with n nodes, a set of pairs is added to the set of constraints with probability $1/n$. A pair forms a must-link constraint if the two nodes have the same class and a cannot-link constraint otherwise.

All algorithms (except COSC) rely on a k-means step which is non deterministic. So, we repeat 30 times each execution and select the partitions that violates a minimal number of constraints. The results evaluated on unconstrained pairs are averaged considering the 10 different sets of constraints.

All experiments were conducted using octave with openblas. For CCSR and COSC, we use the code provided by the authors on their webpages. We are using k-MEANS with smart initialization [1]. Finally, note that we found that initializing gradient descent so that it is close to unconstrained spectral clustering performs better than random initialization.

Results and Discussion. Results for the first set of experiments for 22 datasets are presented in Table 1. Empty cells corresponds to the case where the algorithm did not terminate after 15 minutes.

Table 1. Summary of data sets. First 5 columns show the data set properties: number of nodes in the graph, number of classes, how they have been constructed and number of dimensions in the spectral embedding used for the experiments. The following columns report performances for the various algorithms. Columns COSC and SL report poor results. This is mainly due to the fact that the supervision by must-link constraints is very weak. They do not fully exploit the cannot-link constraints. In our experiments, graphs are not sparse but constraints are sparse. COSC is expecting a sparse graph as an input and satisfy all the constraints when the number of clusters is equal to 2. When the number of clusters is greater than two, COSC looses its guarantees. Moreover, when constraints are very sparse, there is many different ways to satisfy them, and the hierarchical 2-way clustering COSC is performing for more than two clusters can achieve very poor results when the earliest cuts are wrong.

Dataset	size	k	Similarity	m	FGPWC tuning viols.		no tuning viols.		SL viol.		CSP viol.		COSC viol.		CCSR viol.	
breasttissue	106	6	RBF	20	0.3088	3	0.3271	2	−0.0050	35	0.1339	52	0.0695	9	0.2104	5
glass	214	6	RBF	20	0.2552	16	0.1461	23	0.0115	73	0.0182	124	0.0347	20	0.1872	26
hayes-roth	132	3	Cosine	20	0.2783	3	0.1736	13	−0.0146	35	0.0170	78	0.0079	12	0.0842	21
hepatitis	80	2	RBF	10	0.1910	10	0.1220	11	0.0822	17	0.0106	42	0.0184	0	−0.0127	17
imdb	1606	4	Cosine	400	0.1385	93	0.1558	74	−0.0001	648	-	-	−0.0181	298	-	-
interlaced circles	900	3	RBF	60	0.6458	53	0.3023	131	0.1260	208	0.0002	574	0.0110	172	-	-
ionosphere	351	2	RBF	50	0.5041	37	0.4037	11	0.0045	68	0.0045	172	0.0889	19	-	-
iris	150	3	RBF	10	0.9410	1	0.8841	2	0.5657	16	0.0142	68	0.0797	0	0.8485	4
moons	900	2	RBF	10	0.9215	19	0.9045	22	0.0643	231	0.0000	468	-	-	0.6684	72
phoneme	4509	5	RBF	200	0.7073	126	0.0461	746	−0.0002	1842	-	-	-	-	-	-
promoters	106	2	Cosine	10	0.7182	3	0.4307	3	0.0007	21	0.0043	70	0.0341	0	0.5946	8
spam	4601	2	RBF	20	0.9783	21	0.0002	1127	0.0002	1067	-	-	-	-	0.9783	26
tic-tac-toe	958	2	RBF	200	1.0000	0	0.9541	5	0.0037	242	0.0056	404	-	-	-	-
vehicles	846	4	RBF	100	0.3175	55	0.3456	92	0.0001	316	0.0000	728	0.0038	116	-	-
wdbc	569	2	RBF	10	0.8568	14	0.8699	19	0.0024	126	0.0024	264	-	-	0.7255	35
webkb-cornell	195	5	Cosine	10	0.4868	13	0.1166	2	−0.0021	77	−0.0079	134	0.0577	13	0.3317	13
webkb-texas	187	5	Cosine	10	0.4705	11	0.2525	4	−0.0087	68	0.0045	122	0.0707	9	0.2848	25
webkb-wisconsin	265	5	Cosine	10	0.6719	21	0.1018	10	0.0131	77	0.0072	164	0.0226	23	0.3346	32
wikipedia	835	3	Network	10	0.6298	49	0.0105	23	0.4621	111	0.0001	474	0.6960	33	0.5409	76
wine	178	3	RBF	10	0.9649	0	0.9040	1	0.0004	70	0.0031	84	0.0091	41	0.8566	10
xor	900	2	RBF	10	1.0000	0	1.0000	0	−0.0011	223	0.0000	470	-	-	1.0000	0
zoo	101	7	Cosine	10	0.9218	0	0.6536	0	0.1326	25	0.0092	50	0.1447	1	0.7025	2

The column FGPWC "no tuning" is the case where hyperparameters have been set to the following values: $\sigma_m = .15$, $\sigma_c = 1.5$ and m equals the number of eigenvalues lower than .9. The complete spectral embedding of the graph is row normalized, thus the original space is bounded by the unit-hypersphere. Consequently, in the spectral embedding before transformation, distances between data points are less than one. In the column FGPWC "tuning", we tune the σ_m and σ_c parameters using an exhaustive search in the interval $[0.01, 1]$ for σ_m and in the interval $[0.01, 2]$ for σ_c both uniformly splited in 10 equal-size parts.

Without tuning hyperparameters any further, we obtain better results than other approaches in 12 cases. We can also see that our approach is capable of returning a result within a few minutes, whereas some other methods will not within 15 minutes on large data sets. When we tune hyperparameters, we observe that FGPWC outperforms all methods on all datasets while keeping a reasonnable computational time.

We can see that COSC is able to return partitions with 0 violated constraints when the number of clusters $k = 2$, however, the partitions are not necessarily close to the ground-truth partition. An explanation of this phenomenon is that we are providing very few constraints to the different algorithms. Hence, there are many different ways to fullfill the constraints. Columns CSP and SL give poor results. This is mainly due to the fact that the supervision by must-link constraints is very weak. They do not fully exploit the cannot-link constraints. In our experiments, graphs are not sparse but constraints. COSC is expecting a sparse graph as an input and satisfy all the constraints when the number of clusters is equal to 2. When the number of clusters is greater than two, COSC looses its guarantees. Moreover, when constraints are very sparse, there are many different ways to satisfy them, and the hierarchical 2-way clustering COSC is performing for more than two clusters can achieve very poor results when the earliest cuts are wrong. It is particularly interesting to compare FGPWC to CCSR, since the the approaches developped in the two algorithms are both based on a change of representation of the spectral embedding. CCSR is competitive with FGPWC w.r.t. the ARI measure in many cases. However, we can see that CCSR becomes intractable as the size of the embedding m increases, while this is not a problem for FGPWC. This is also confirmed by the computational time.

Small graphs can be harder if constraints contradict the similarity $\mathbf{W}$, because in this case m needs to be larger, but for a large enough m, our algorithm will over-fit. It is related to the degree of freedom in solving a system of K equations, where K is the fixed number of constraints, with more and more variables (as m increases).

5.2   Noun Phrase Coreference Resolution

Dataset and Preprocessing. For the coreference resolution task, we use the English dataset used for the CoNLL-2012 shared task [25]. Recall that the task consists, for each document, in partitioning a set of noun phrases (aka *mentions*) into classes of equivalence that denote real-wold *entities*. This task is illustrated on the following small excerpt from CoNLL-2012:

> *Was Sixty Minutes unfair to [Bill Clinton]$_1$ in airing Louis Freeh's charges against [him]$_1$?*

In this case, noun phrases "Bill Clinton" and "him" both refer to the same entity (i.e. William Jefferson Clinton), encoded here by the fact that they share the same index[1]. The English CoNLL-2012 corpus contains over over 2K documents (1.3M words) that fall into 7 categories, corresponding to different domains (e.g., newsiwre, weblogs, telephone conversation). We used the official train/dev/test splits that come with the data. Since we were specifically interested in comparing approaches rather than developing the best end-to-end system, we used the *gold mentions*; that is, we clustered only the noun phrases that we know were part of ground-truth entities.

The mention graphs are built from a model of pairwise similarity, which is trained on the training section of CoNLL-2012. The similarity function is learned using logistic regression, each pair of mentions being described by a set of features. We re-use features that are commonly used for mention pair classification (see e.g., [23],[4]), including grammatical type and subtypes, string and substring matches, apposition and copula, distance (number of separating mentions/sentences/words), gender and number match, synonymy/hypernym and animacy (based on WordNet), family name (based on closed lists), named entity types, syntactic features and anaphoricity detection.

Evaluation Metrics. The systems' outputs are evaluated using the three standard coreference resolution metrics: MUC [29], B^3 [2], and Entity-based CEAF (or CEAF$_e$) [20]. Following the convention used in CoNLL-2012, we report a global F1-score (henceforth, CoNLL score), which corresponds to an unweighted average of the MUC, B^3 and CEAF$_e$ F1 scores. Micro-averaging is used throughout when reporting scores for the entire CoNLL-2012 test. Additionally, we are reporting the adjusted rand index.

In order to analyze performance for different cluster sizes, we also computed per-cluster precision and recall scores. Precision p_i and recall r_i are computed for each reference entity class $\mathcal{C}_i$ for all documents. Then, the micro-averaged F1-score score is computed as follows:

$$\bar{p} = \sum_i \frac{|\mathcal{C}_i|\, p_i}{\sum_j |\mathcal{C}_j|} \qquad \bar{r} = \sum_i \frac{|\mathcal{C}_i|\, r_i}{\sum_j |\mathcal{C}_j|} \qquad \text{F1} = \frac{2\bar{p}\bar{r}}{\bar{p} + \bar{r}}$$

System Settings. Following the approach in [6], we first create for each document a fully connected similarity^2 graph between mentions and then run our clustering algorithm on this graph. Compared to the tasks on the UCI dataset, the main difficulties are the determination of the number of clusters and the fact

---

[1] Note that noun phrases like "Sixty Minutes" and "Louis Freeh" also denote entities, but such singleton entities are not part of the CoNLL annotations.

[2] Pamameter estimation for this pairwise mention model was performed using Limited-memory BFGS implemented as part of the Megam package `http://www.umiacs.umd.edu/~hal/megam/version0_3/`. Default settings were used.

that we have to deal with many small graphs (documents contain between 1 and 300 mentions).

The same defaut values were used for the σ_m and σ_c parameters, as in the previous experiments (that is, 0.15 and 1.5, respectively). In our aglorithm we need to fix parameter m. We fix a value that is a tradeoff between the dimension of $\mathbf{L_{sym}}$ and the number of constraints. Indeed, we want to keep structural information comming from the graph through the eigendecomposition of $\mathbf{L_{sym}}$. Also, we reject the situations where m is much larger than the number of constraints because they can lead to solutions that are non satisfactory. In that latter case, the optimization problem can be solved without any impact on non-constrained pairs and therefore without any generalization based on the given constraints. Because the multiplicity of eigenvalue 1 is large in this dataset, m is estimated by $m = |\{\lambda_i : \lambda_i \leq 0.99\}|$ where λ_i are the eigenvalues of $\mathbf{L_{sym}}$. The number of clusters k is estimated by $k = \left|\{\lambda_i : \lambda_i \geq 10^{-5}\}\right|$ where λ_i are the eigenvalues of $\mathbf{X}^\top\mathbf{X}$.

As for the inclusion of constraints, we experimented with two distinct settings. In the first setting, we automatically extracted based on domain knowledge (setting (c) in the results below). Must-link constraints were generated for pairs of mention that have the same character string. For cannot-link constraints, we used number, gender, animacy, and named entity type dismatches (e.g., noun phrases with different values for gender cannot corefer). These constraints are similar to some of the deterministic rules used in [17] and overlap with the information already in the features. This first constraint extraction generates a lot of constraints (usually, more than 50% of all available constraints for a document), but it is also noisy. Some of the constraints extracted this way are incorrect as they are based on information that is not necessarily in the dataset (e.g., gender and number are predicted automatically). The precision of these constraints is usually higher than 95%. In a second simulate interactive setting, we extracted a smaller set of must-link and cannot-link constraints directly from the ground-truth partitions, by drawing coreferential and non-coreferential mention pairs at random according to a uniform law (setting (b) below). In turn, all of these constraints are correct. Each mention pair has a probability $1/n$ to be drawn, with n the mention count.

Results and Discussion. We want to show that FGPWC works better on large graphs and larger clusters. We perform per-cluster evaluation, this is summarized in Figure 3. All plots represent the F1-score, averaged on runs all documents per cluster size. Plot (a) reports results for the unconstrained spectral clustering approach of [6]. Their method uses a recursive 2-way spectral clustering algorithm. The parameter used to stop the recursion has been tuned on a development set. The other plots are obtained using (b) FGPWC with constraints generated uniformly at random from an oracle and (c) FGPWC with constraints derived automatically from text based on domain knowledge.

In the latter case (c) FGPWC has not been able to improve the results obtained by (a). We think that constraints extracted from text does not add new information but change the already optimized measure in the similarity graph.

However, even adding less constraints at random from an oracle using a uniform distribution is more informative. When we are using constraints that do not comes from the features used for the similarity construction step, we see that FGPWC outperform other methods for clusters larger than 5. However, we can see that FGPWC can degrade smallest clusters. There are two explanations for this: we obtain better performance on larger clusters because the way we select random constraints. Using a uniform distribution, there is more chance to add constraints for larger clusters. And moreover, clusters with few or no constraints, in our case: small clusters, are usually scattered around the space, because FG-PWC globally transforms the space to fit the constraints. We can also see that (b) outperforms (c) on small clusters. Probably because more constraints are being added for small clusters in (b). All of this supports the idea that constraints in this kind of task should be generated from another set of features applicable to all mentions, regardless of the size of the clusters they belong to.

Overall, we obtain a CoNLL score of 0.71 (0.80 MUC, 0.75 B^3, 0.57 $CEAF_e$, 0.48 ARI), for [6], 0.56 (0.76 MUC, 0.57 B^3, 0.36 $CEAF_e$, 0.31 ARI) using our method along with extracted constraints and 0.58 (0.67 MUC, 0.58 B^3, 0.49 $CEAF_e$, 0.40 ARI) with ground-truth random constraints. That is, we see a clear drop of performance when using the constraints, be they noisy or not. Closer examination reveals that this decrease stems from poor performance on small clusters, while these clusters are the most representative in this task.

The F1-score is lower than for the state of the art. But interestingly, in presence of uniformly distributed pairwise constraints, our algorithm can significantly improve clustering results on clusters larger than 5, compared to the state of the art [6]. This suggests that active methods can lead to dramatic improvements
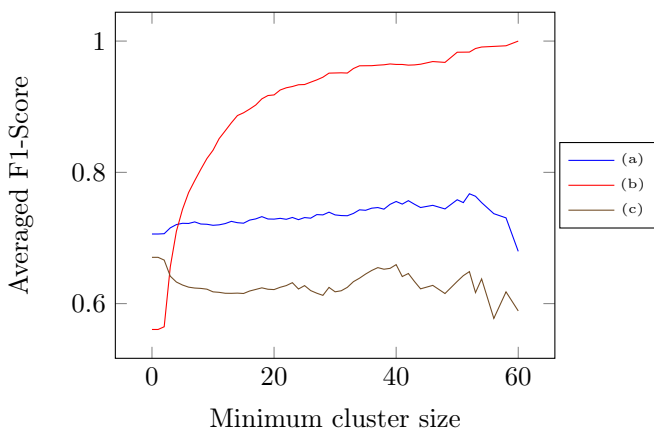
Fig. 3. Averaged F1-score vs minimum cluster size for FGPWC with CoNLL 2012 data set: (a) method in [6], (b) FGPWC uniformly distributed from reference; (c) FGPWC All extracted must/cannot-links

and our algorithm easily supports that through the introduction of pairwise constraints. Moreover, our method can be used to detect larger clusters, and leave the smaller cluster to another method.

6   Conclusion

We proposed a novel constrained spectral clustering framework to handle must-link and cannot-link constraints. This framework can handle both 2 clusters and more than 2 clusters cases using the exact same algorithm. Unlike previous methods, we can cluster data which require more eigenvectors in the analysis. We can also handle cannot-link constraints without giving up on computational complexity. We carried out experiments on UCI and network data sets. We also provide an experiment on the real task of noun-phrase coreference and discuss the results. We discuss the relationship between Laplacian eigenmaps and the constraints, that can explain why adding constraints can degrade clustering results. We empirically show that our method, that involves a simple and fast gradient descent, outperforms several state of the art algorithms on various data sets. For noun-phrase coreference, the challenge ahead will be to find rules to generate constraints from the text which are more uniformly distributed. We also want to find a way to better handle small clusters. A step in that direction is to investigate better adapted cut criteria and active learning methods.

References

1. Arthur, D., Vassilvitskii, S.: K-means++: The advantages of careful seeding. In: Proc. of SODA, pp. 1027–1035 (2007)
2. Bagga, A., Baldwin, B.: Algorithms for scoring coreference chains. In: Proc. of TREC, vol. 1, pp. 563–566 (1998)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396 (2002)
4. Bengtson, E., Roth, D.: Understanding the value of features for coreference resolution. In: Proc. of EMNLP, pp. 294–303 (2008)
5. De Bie, T., Suykens, J.A.K., De Moor, B.: Learning from General Label Constraints. In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 671–679. Springer, Heidelberg (2004)
6. Cai, J., Strube, M.: End-to-end coreference resolution via hypergraph partitioning. In: Proc. of COLING, pp. 143–151 (2010)
7. Davidson, I., Ravi, S.S.: Clustering with constraints: Feasibility issues and the k-means algorithm. In: Proc. of SIAM Data Mining Conference (2005)
8. Dhillon, I.S., Guan, Y., Kulis, B.: A unified view of kernel k-means, spectral clustering and graph cuts. Technical report, UTCS (2004)
9. Finley, T., Joachims, T.: Supervised clustering with support vector machines. In: Proc. of ICML, pp. 217–224. ACM Press (2005)
10. Guattery, S., Miller, G.L.: On the quality of spectral separators. SIAM Journal on Matrix Analysis and Applications 19(3), 701–719 (1998)
11. Hein, M., Setzer, S.: Beyond spectral clustering - tight relaxations of balanced graph cuts. In: Proc. of NIPS, pp. 2366–2374 (2011)

12. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification 2, 193–218 (1985)
13. Jordan, M., Bach, F.: Learning spectral clustering. In: Proc. of NIPS (2004)
14. Kamvar, S.D., Klein, D., Manning, C.D.: Spectral learning. In: Proc. of IJCAI, pp. 561–566 (2003)
15. Klein, D.J., Randic, M.: Resistance distance. Journal of Mathematical Chemistry 12, 81–95 (1993)
16. Kulis, B., Basu, S., Dhillon, I.S., Mooney, R.J.: Semi-supervised graph clustering: a kernel approach. In: Proc. of ICML, pp. 457–464 (2005)
17. Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., Jurafsky, D.: Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In: Proc. of CoNLL: Shared Task, pp. 28–34 (2011)
18. Li, Z., Liu, J., Tang, X.: Constrained clustering via spectral regularization. In: Proc. of CVPR, pp. 421–428 (2009)
19. Lu, Z., Carreira-Perpinan, M.A.: Constrained spectral clustering through affinity propagation. In: Proc. of CVPR (2008)
20. Luo, X.: On coreference resolution performance metrics. In: Proc. of HLT-EMNLP, pp. 25–32 (2005)
21. Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
22. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Proc. of NIPS, pp. 849–856. MIT Press (2001)
23. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proc. of ACL, pp. 104–111 (2002)
24. Nicolae, C., Nicolae, G.: Bestcut: A graph algorithm for coreference resolution. In: Proc. EMNLP, pp. 275–283 (2006)
25. Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y.: Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In: Joint Conference on EMNLP and CoNLL-Shared Task, pp. 1–40 (2012)
26. Rangapuram, S.S., Hein, M.: Constrained 1-spectral clustering. In: Proc. of AISTATS, pp. 1143–1151 (2012)
27. Saerens, M., Fouss, F., Yen, L., Dupont, P.E.: The principal components analysis of a graph, and its relationships to spectral clustering. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 371–383. Springer, Heidelberg (2004)
28. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
29. Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: Proc. of the Conference on Message Understanding, pp. 45–52 (1995)
30. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained K-means Clustering with Background Knowledge. In: Proc. of ICML, pp. 577–584 (2001)
31. Wang, X., Davidson, I.: Flexible constrained spectral clustering. In: Proc. of KDD, pp. 563–572 (2010)
32. Wang, X., Qian, B., Davidson, I.: On constrained spectral clustering and its applications. In: Data Mining and Knowledge Discovery (2012)
33. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance Metric Learning, with Application to Clustering with Side-information. In: Proc. of NIPS (2002)
34. Yu, S.X., Shi, J.: Grouping with Bias. In: Proc. of NIPS (2001)
35. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proc. ICML, p. 912 (2003)