Regularized Co-Clustering on Manifold

Ying Liu
Division of Computer Science, Mathematics and Science
St. John's University
Queens, NY 11349
Email: yliu1 {at} stjohns.edu

Henry Han
Department of Computer and Information Science
Fordham University
New York, NY 10023

Chengcheng Shen
Amazon.com LLC
Seattle, WA 98109

Abstract—**Co-clustering is to partition rows and columns of a matrix simultaneously. It has been an important research field in data mining and machine learning. It is preferred over traditional homogeneous clustering techniques in many real applications. In this paper, we present a co-clustering algorithm based on local information and regularization. The algorithm seeks to preserve the local intrinsic geometry and measure smoothness of indicator functions with respect to the bipartite graph. The minimization of the objective function can be formulated as a generalized eigenvalue problem. The experimental results show that the algorithm outperforms the existing spectral and information-theoretic co-clustering algorithms. The results also show that the algorithm correctly co-clusters documents with related words.**

Keywords-co-clustering, data mining, machine learning, algorithm

I. INTRODUCTION

Clustering is the grouping together of similar objects. It achieves simplification by representing complex data objects by a few clusters such that data objects within the same cluster are similar while data objects in different clusters are dissimilar. Research efforts have been devoted to one-way clustering based on pair-wise similarity of homogeneous data objects. Most of the conventional algorithms require the data objects to be homogeneous. However, there has been a growing interest in developing algorithms capable of simultaneously cluster both dimensions of a relational matrix. Co-clustering has been used in a wide variety of applications such as text mining [11, 15], web-log mining [35], market-basket data analysis [11, 15], and biological microarray data analysis[18] etc. In these applications, data is represented as an inter-relational matrix or co-occurence table such as document-term matrix in text mining.

Dhillon et al. [11] propose a co-clustering algorithm in which authors model the relationship between data objects as a bipartite graph and seek to find the minimal normalized cut in the graph with spectral relaxation. El-Yaniv et al. [13] use an agglomerative hard clustering version of the information bottleneck method [29] to cluster documents and then words. Dhillon et al. [12] propose an information theoretic co-clustering algorithm to monotonically increase the preserved mutual information by intertwining both the row and column clusterings at all stages. Later, a more generalized co-clustering framework based on Bregman divergence is presented by Banerjee et al. [2]. A soft co-clustering algorithm[21] is also proposed, which is able to work with any regular exponential family distribution and corresponding Bregman divergences. Besides, approximation algorithms [25, 1] are also proposed for co-clustering problems. Shafiei and Milios [20] present a hierarchical Bayesian model for simultaneously clustering documents and terms, where each document is modeled as a random mixture of document topics and each topic is a distribution over some segments of the text.

The use of kernel functions has provided a powerful way of detecting nonlinear relations using linear methods. The assumption is that the nonlinear pattern appears linear after the embedding of data into a new feature space, but without the prior knowledge of data distribution, the kernel function is not guaranteed to be consistent with the characteristics of data. In supervised learning, target function is imposed with smoothness condition with respect to the labeled data points. In unspervised problems, in terms of graph theory, we can let the kernel function to respect the smoothness to the graph. Besides, the local information also plays an important role in many learning algorithms. In this paper, we combine both local and global information and propose an algorithm called Regularized Co-Clustering on Manifold (RCCM), which imposes smoothness condition to bipartite graph and preserves local geometry structure. The objective function can be optimized by spectral relaxation, so RCCM can be regarded as belonging to category of spectral clustering approaches. In order to show the effectiveness of our algorithm, we conduct experiments in which we compare our algorithm with seven related clustering algorithms in eight datasets.

The remaining of the paper is organized as follows. In section 2, we introduce background information and related work. The details of our co-clustering algorithm are presented in section 3. Experimental results are then provided in section 4. Finally we conclude the paper in the last section.

II. RELATED WORK

A. Manifold Regularization

Manifold regularization [5] is a family of learning algorithms based on regularization that exploit the geometry of the marginal distribution. The framework focuses on semi-supervised learning problems that involve both labeled and unlabeled data. This framework brings together three distinct concepts: spectral graph theory, manifold learning and regularization in Reproducing Kernel Hilbert Space (RKHS) which leads to a class of kernel based algorithms. Two regularization terms are involved in this framework: one controls the complexity of the classifier in the ambient space and the other controls the complexity as measured by the geometry of the distribution.

Let X denote the data matrix with n data points and d features. For a Mercer kernel $K: X \times X \to \mathbb{R}$, there is an associated RKHS H_k of functions $X \to \mathbb{R}$ with the corresponding norm $\| \ \|_k$. Given a set of labeled examples $(x_i, y_i), i = 1, \ldots, l$, the framework estimates an unknown function by minimizing

$$f^* = \text{argmin}_{f \in H_k} \frac{1}{l} \sum_{i=1}^{l} V(x_1, y_i, l) \quad (1)$$

$$+\gamma_A \|f\|_K^2 + \gamma_I \|f\|_I^2 \quad (2)$$

where V is some loss function, such as squared loss $(y_i - fxi)2$, $fK2$ imposes smoothness conditions on possible solution by penalizing the RKHS norm, $\|f\|_I^2$ is a smoothness penalty corresponding to the intrinsic structure of marginal distribution P_X for unlabeled samples $x \in X$ based on the probability distribution P derived from labeled samples. Studies have shown that the support of P_X is a compact manifold $M \subset \mathbb{R}^n$. In this case, $\|f\|_I^2$ can be defined as:

$$\int_{x \epsilon M} \|\nabla_M f\|^2 dP_X(x) \quad (3)$$

where ∇_M is the gradient of f along the manifold M and the integral is taken over the marginal distribution. The term $\int_{x \epsilon M} \|\nabla_M f\|^2 dP_X(x)$ may be approximated on the basis of labeled and unlabeled data using the graph Laplacian associated to the data. Eq. (3) can be used to measure the smoothness of the mapping function f in the intrinsic geometry of the data set. Thus, if we have a set of u unlabeled samples $\{x_j\}_{j=l+1}^{j=l+u}$, we have the following optimization problem:

$$f^* = \underset{f \in H_k}{\text{argmin}} \frac{1}{l} \sum_{i=1}^{l} V(x_1, y_i, f) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2} \sum_{i,j=1}^{l+u} \left(f(x_i) - f(x_j)\right)^2 W_{ij}$$

$$= \text{argmin}_{f \in H_k} \frac{1}{l} \sum_{i=1}^{l} V(x_1, y_i, f) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2} f^T L f \quad (4)$$

where W_{ij} are edge weight in the data adjacency graph, $f = [f(x_1, \ldots, f(x_{l+u})]^T$, and L is the graph Laplacian given by $L = D - W$. Here the diagonal matrix D is given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$. γ_I and γ_A are coefficients controlling the complexity of functions in both terms.

The minimizer of above optimization problem is in form of the following equation based on the Rerepsenter Theorem [5].

$$f^*(x) = \sum_{i=1}^{l+u} \alpha_i k(x_i, x) \quad (5)$$

B. Bipartite Graph Partition

In the co-clustering algorithm introduced in [11], the relationship between data objects and features is modeled as a bipartite graph. An edge *(i, j)* exists if feature $y_j$has a non-zero value in x_i, the edge weight is given as X_{ij}. The bipartite graph can be transformed to a homogeneous-like adjacency matrix,

$$W = \begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix}$$

and the normalized Laplacian is given as

$$M = I - D^{-1/2} W D^{-1/2}$$

I is the $(n + d) \times (n + d)$ identity matrix. [11] proposes to find an optimal partition of the bipartite graph by minimizing the relaxation of the objective function derived from the normalized cut using the spectral clustering method. The graph partition can be obtained by performing *k*-means on the smallest eigenvectors of *M*. It is also shown that the spectral clustering in the algorithm can be related to the singular values of normalized *X*.

III. CO-CLUSTERING ALGORITHM

A. Objective Function

Let's consider row element *x* to be member of $R \subset \mathbb{R}^n$, and column element *y* to be member of $C \subset \mathbb{R}^d$. The co-clustering problem is formulated as follows: for a data matrix *X* defined above, we want to co-cluster both row and column into *k* clusters. The algorithm should output a row partition function: $\pi_r: R \to \{i\}_{i=1}^k$ and a column partition function $\pi_c: C \to \{i\}_{i=1}^k$ that give cluster assignment to row and column indices respectively. In this paper, x_i is the *i*th row element and y_i is the *i*th column element in the data matrix. Let us also introduce

$k_r: R \times R \to \mathbb{R}$ to be the row kernel that defines an associated RKHS H_r . Similarily, $k_c: C \times C \to \mathbb{R}$ denotes the column kernel that defines an associated RKHS H_c.

Consider a simultaneous assignment of rows and columns into k classes. For any data object x, $P_r(x) = [p_r^1(x), \dots, p_r^k(x)]^T \in \mathbb{R}^k$ to be a vector whose elements are soft cluster assignments where $p_r^j \in H_r$ for all j. Then we have $P_r$as the $n \times k$ cluster assign matrix. Similarly, $P_c(y)$ is defined for column (feature) $y \in C$, P_c is $d \times k$ cluster assign matrix for column data objects.

Many algorithms map data objects onto a new space and then hidden structure could be explored on the new basis. Based on manifold assumption [4, 9], if two data objects x_i, x_j are close in the intrinsic geometry of the data distribution, then the representation of this two objects are also close to each other in the new space. This rule plays an important role in developing various kinds of algorithms including dimension reduction [4] and semi-supervised learning algorithms [5, 36, 37]. Under the semi-supervised setting, Eq.(3) can be approximate by graph Laplacian based on the labeled samples, but in the unsupervised case, there is no known information about the data manifold, thus Eq.(3) cannot be computed. Recent studies on spectral graph [10] and manifold learning theory[3] have demonstrated that $\|f\|_I^2$ can be discretely approximated by the nearest neighbor graph on the scatter of data objects.

Consider a graph with *n* vertices, each vertex corresponds to a data object. We can define the edge weight matrix U as follows:

$$U_{ij} = \begin{cases} 1, & if\ x_i \in N_s(x_j) or\ x_j \in N_s(x_i) \\ 0, & otherwise \end{cases} \tag{6}$$

where $N_s(x_i)$ denotes the set of s nearest neighbors of x_i. Define $L = D - U$, where D is a diagonal matrix whose entries are column or row sums of U, $D_{ii} = \sum_j U_{ij}$. L is called graph Laplacian, which is a discrete approximation to the Laplace-Beltrami operator on the manifold. Thus, let $f(x_i) = p_i$ be the mapping of original data object x_i onto the new basis. Eq.(3) can be computed as follows:

$$\begin{aligned} \|f\|_I^2 &= \frac{1}{2}\sum_{i,j=1}^{n}(f(x_i) - f(x_j))^2 U_{ij} \\ &= \sum_{i=1}^{n} f(x_i)^2 D_{ii} \\ &\quad - \sum_{i,j=1}^{n} f(x_i) f(x_j) U_{ij} = \sum_{i=1}^{n} p_i^2\, D_{ii} \\ &\quad - \sum_{i,j=1}^{n} p_i p_j U_{ij} = p^T D p - p^T U p = p^T L p \end{aligned} \tag{7}$$

By penalizing $\|f\|_I^2$, we can get a smooth mapping function on the data manifold. The intuition is that when two data objects are close in original data distribution, then $f(x_i)$ and $f(x_j)$ are also close to each other. Based on this idea, we have the following optimization function for co-clustering problem.

$$\begin{aligned} J = \underset{P_r \in H_r^k, P_c \in H_c^k}{\operatorname{argmin}} & \gamma_r \sum_{i=1}^{k} \|p_r^i\|^2 H_r + \gamma_c \sum_{i=1}^{k} \|p_c^i\|^2 H_c \\ & + tr(P_r^T L_r P_r) + tr(P_c^T L_c P_c) \\ & + \mu tr((P_r^T P_c^T) M \begin{pmatrix} P_r \\ P_c \end{pmatrix}) \end{aligned} \tag{8}$$

L_c and L_r are graph Laplacian of nearest neighbor graphs for row and column respectively. γ_r, γ_c, μ are parameters that tradeoff various regularization terms. γ_r, γ_c are ratios between the first and the third terms, the second and the fourth terms respectively. *tr* denotes the trace operator of a matrix. The first two terms are usual RKHS norms on the cluster indicator functions for rows and columns. The middle two terms measure the smoothness of the intrinsic geometry based on the nearest neighbor graphs for rows and columns. The last term is used to measure the smoothness of row and column cluster indicator functions regarding to the bipartite graph partition in section II (B). Eq.(8) extends the classic manifold regularization to unsupervised coclustering by imposing restrictions on local information and global bipartite graph partition.

It is obvious that the solutions for row and column cluster indicator function have the following form by Representer Theorem,

$$p_r^j(x) = \sum_{i=1}^{n} \alpha_{ij} k_r(x, x_i), 1 \le j \le k \tag{9}$$

$$p_c^j(y) = \sum_{i=1}^{n} \beta_{ij} k_c(y, y_i), 1 \le j \le k \tag{10}$$

B. Computation

Let α, β denote the corresponding optimal expansion coefficient matrices from Eq.(9) and (10). Let K_r and K_c be gram matrices over data points and features respectively. Put them back into the Eq.(8), since K_r and K_c are symmetric, we can rewrite *J* as follows:

$$\begin{aligned} J(\alpha, \beta) = \underset{\alpha, \beta}{\operatorname{argmin}} & \gamma_r tr(\alpha^T K_r \alpha) + \gamma_c tr(\beta^T K_c \beta) \\ & + tr(\alpha^T K_r L_r K_r \alpha) + tr(\beta^T K_c L_c K_c \beta^T) \\ & + \mu tr((\alpha^T K_r \beta^T K_c) M \begin{pmatrix} K_r & \alpha \\ K_c & \beta \end{pmatrix}) \end{aligned} \tag{11}$$

After some mathematical manipulations, Eq. (11) can be written as:

$$J(\alpha,\beta) = \underset{\alpha,\beta}{\operatorname{argmin}}\, tr\left((\alpha^T\beta^T)\begin{pmatrix}\gamma_r K_r & O\\ O & \gamma_c K_c\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix}\right) + tr\left((\alpha^T\beta^T)\begin{pmatrix}K_r L_r K_r & O\\ O & K_c L_c K_c\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix}\right) + \mu tr((\alpha^T\beta^T)\begin{pmatrix}K_r & O\\ O & K_c\end{pmatrix} M \begin{pmatrix}K_r & O\\ O & K_c\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix}) \quad (12)$$

O is a matrix of all zeros. Then we combine all terms in Eq. (12) together, the final objective function for regularized co-clustering on manifold is as follows:

$$J(\alpha,\beta) = \left(\underset{\alpha,\beta}{\operatorname{argmin}}\, \gamma_r tr((\alpha^T\beta^T) A\begin{pmatrix}\alpha\\ \beta\end{pmatrix}\right) \quad (13)$$

where A is defined as follows:

$$A = \begin{pmatrix}\gamma_r K_r & O\\ O & \gamma_c K_c\end{pmatrix} + \begin{pmatrix}K_r L_r K_r & O\\ O & K_c L_c K_c\end{pmatrix} + \begin{pmatrix}K_r & O\\ O & K_c\end{pmatrix} + \mu M\begin{pmatrix}K_r & O\\ O & K_c\end{pmatrix} \quad (14)$$

Note that to avoid degenerate solutions, we need to impose some additional constraints of orthogonality [5, 4].

$$1^T K_r \alpha = 0, \alpha^T K_r^2 \alpha = 1$$
$$1^T K_c \beta = 0, \beta^T K_c^2 \beta = 1 \quad (15)$$

where 1 is the vector of all ones.

C. Spectral Relaxation

The minimization problem of Eq. (13) and (14) with constraints in Eq. (15) can be formulated as minimization of a generalized Rayleigh quotient which can be converted to a generalized eigenvalue problem. We can write the generalized eigenvalue problem [5, 22] as:

$$A\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \lambda\begin{pmatrix}K_r & O\\ O & K_c\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix} \quad (16)$$

where λ is a diagonal matrix of eigenvalues. We are looking for eigenvectors corresponding to the smallest k eigenvalues to minimize Eq. 913). For simplicity, if we let

$$B = \begin{pmatrix}K_r & O\\ O & K_c\end{pmatrix}$$
$$S = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$$

then Eq. (16) is equivalent to

$$AS = \lambda B^2 S \quad (17)$$

If B is invertible, let $S = B^{-1}V'$, Eq. (17) can be converted to a standard eigenvalue problem:

$$B^{-1}AB^{-1}V = \lambda V \quad (18)$$

where the matrix $B^{-1}AB^{-1}$ is symmetric. Let V' be the matrix of eigenvectors corresponding to the smallest k eigenvalues of Eq. (18),

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = B^{-1}V' \quad (19)$$

Then the partition functions are defined by:

$$\pi_r(x) = \underset{1\le j\le k}{\operatorname{argmax}} \sum_{i=1}^{n} \alpha_{ij}\, k_r\,(x, x_i) \quad (20)$$

$$\pi_c(y) = \underset{1\le j\le k}{\operatorname{argmax}} \sum_{i=1}^{d} \beta_{ij}\, k_c\,(y, y_i) \quad (21)$$

The regularized co-clustering on manifold algorithm is summarized in Table I.

TABLE I. REGULARIZED CO-CLUSTERING ON MANIFOLD

	Regularized Co-Clustering on Manifold
1	Input: Data matrix X, number of clusters k, parameters: $\boldsymbol{\gamma_r, \gamma_c, \mu}$
2	Form matrixes K_r, K_c with a chosen kernel function. Form Laplacian matrices M, L_r, L_c Form matrices A, B in sections III.B and III.C
3	Solve the eigenvalue problem in Eq. (18) Obtain α and β in Eq. (19)
4	Output clusters using Eq. (20) and Eq. (21)

D. Relation to other Approaches

The graph Laplacian $L = D - W$ or normalized graph Laplacian M in section II.B is a central object in many graph-based learning algorithms. Let $p^T = (p_r^T p_c^T)$,

$$p^T L p = \frac{1}{2}\sum W_{ij}(p(i) - p(j))^2 \ge 0$$

where the inequality holds when W has non-negative entries. We see that the last term in Eq. (8) measures the smoothness of p on the graph. It also means that $p(i) \approx p(j)$for pares with large values, thus smaller value of $p^T L p$ means smoother p. If we remove terms $tr(P_r^T L_r P_r)$and $tr(P_c^T L_c P_c)$ in Eq. (8), then we have

$$J = \underset{P_r\in H_r^k, P_c\in H_c^k}{\operatorname{argmin}} \gamma_r \sum_{i=1}^{k} \left\|p_r^i\right\|^2 H_r + \gamma_c \sum_{i=1}^{k} \left\|p_c^i\right\|^2 H_c + tr((P_r^T P_c^T) M \begin{pmatrix}P_r\\ P_c\end{pmatrix})$$

This equation is regularized spectral clustering [5] on rows and columns. It tries to respect the smoothness to the original bipartite graph W.

The Laplacia eigenmap algorithm [4] constructs a weighted graph *U* with edges connecting nearby points. It chooses to minimize

$$\frac{1}{2}\sum_{i,j}(x_i - x_j)^2 U_{ij} = x^T L x$$

L is Laplacian matrix of graph *U*. The graph U can actually be the nearest neighbor graph constructed in section III.A. If let μ=0 in Eq. (8), then we have

$$J = \underset{P_r \in H_r^k, P_c \in H_c^k}{\operatorname{argmin}} \gamma_r \sum_{i=1}^{k} \left\| p_r^i \right\|^2 H_r + \gamma_c \sum_{i=1}^{k} \left\| p_c^i \right\|^2 H_c + tr(P_r^T L_r P_r) + tr(P_c^T L_c P_c)$$

This is regularized Laplacian eigenmap functions for rows and columns with P_r and P_c be the mapping to RKHS.

The local learning idea has been used in supervised learning[6], dimension reduction [32, 17, 19, 28] etc. For each data point, a model is trained only based on its neighboring data points in supervised learning. Then the output function (classifier) is used on all the unlabeled data points for prediction. It has been reported that local learning algorithms often perform better than global learning algorithms [6]. In unsupervised problems, kernel ridge regression (LLCA)[31] and kernel regression [27] are used as local label predictor respectively, both objective functions are then transformed to a spectral clustering problem. Wang et al. [30] use local ridge regression combined with a global regularizer for document clustering problem. In these methods, a data point estimates its own label information from its neighbors by minimizing sum of squared error or kernel density estimation. Take LLCA as an example, let $F = [f^1, \dots, f^c] \in \mathbb{R}^{n \times c}$ be a scaled partition matrix, *n* is number of objects and c is number of clusters. The basic idea is to minimize:

$$min_{F \in \mathbb{R}^{n \times c}} \sum_{l=1}^{n} \| f^l - o^l \|^2$$

where $o^l = [o_1^l(x_1), \dots o_n^l(x_n)]^T \in \mathbb{R}^n$ denotes the output function of a kernel machine trained with data from nearest neighbors N_i. Based on kernel machine, $o_i^l(x_i)$ can be written as:

$$o_i^l(x_i) = \sum_{x_j \in N_i} \beta_{ij}^l K(x_i, x_j)$$

For each *x*, we may use kernel ridge regression to obtain $o_i^l(x_i)$, thus we have:

$$min_{\beta_i^l \in \mathbb{R}^{n_i}} \lambda (\beta_i^l)^T \mathrm{K}_i \beta_i^l + \left\| \mathrm{K}_i \beta_i^l - f_i^l \right\|^2$$

where $\beta_i^l = [\beta_{ij}^l]^T$.

This function is very similar to the supervised regularized least squares in manifold regularization framework. We have *l* labeled examples $(x_i, y_i)_{i=1}^l$, the regularized least squares method can be written as:

$$min_{f \in \mathcal{H}_K} \frac{1}{l}\sum_{i=1}^{l} \left(y_i - f(x_i)\right)^2 + \gamma \|f\|_K^2$$

Based on Representer Theorem, the solution can be shown as:

$$f^* = \sum_{i=1}^{l} \beta_i \mathrm{K}(x, x_i)$$

We see that the difference between LLCA and regularized least squares is that the output function $o_i^l(.)$ in LLCA is trained with nearby data objects while regularized least squares is trained with labeled data objects. In our approach, since there are no labeled data objects, we try to minimize the difference of cluster labels among a data object and its neighbors so that we can preserve the local geometry of the data distribution, so the basic idea is different from local learning approach. In [9] and [7], authors use nearest neighbor graph to preserve local information and then data points are projected onto a lower-dimensional semantic space in which documents related to same semantics are close to each other. These methods are based on Euclidean space, while our approach is kernel-based. In addition, we add RKHS norms as regularization terms to smooth cluster indicator functions. [24] proposes a semi-supervised regularized co-clustering method based on the manifold regularization framework. The method also uses bipartite graph partitioning as a regularization term; but it is semi-supervised and without considering local geometry information. Another co-clustering algorithm Dual Regularized Co-Clustering (DRCC) [16] is very similar to Graph-regularized NMF (GNMF) [9]. DRCC considers the geometric structures in both the data points and features while GNMF only considers the geometric structures in data points. RCCM extracts cluster information from eigenstructure of a matrix (A in Eq.(14)), so it can be considered as a spectral co-clustering approach. The matrix used to generate eigenvectors plays an important role in spectral clustering. Overall, RCCM imposes smoothness of indicator functions with respect to both local geometric structure and global bipartite graph co-clustering, which is different from current available co-clustering algorithms.

IV. Experimental Evaluation

In this section, we empirically compare RCCM with the kernel spectral clustering [34, 23], regularized spectral clustering [5], spectral co-clustering [14, 11], information theoretic co-clustering algorithm (ITCC) [12], GNMF and LLCA. These algorithms either are co-clustering algorithms or have close relationships with our approach.

A. Datasets

Eight datasets are used in the experiments. News-1 and News-2 are from the 20-Newsgroup data which contains about

20000 articles from 20 newgroups (http://people.csail.mit.edu/~jrennie/20Newsgroups). News-1 is sampled from four categories of rec.∗ . News-2 is sampled from four categories of sci.∗ . 120 documents are sampled from each category. We use text classification package Rainbow (http://www.cs.cmu.edu/~mccallum/bow/rainbow/) to preprocess the data by removing stop words and file headers and selecting words with more than 7 counts. The document-word matrix is built based on tf-idf. The WebKB dataset contains webpages gathered from university computer science departments. There are about 8280 documents and divided into 7 categories: student, faculty, staff, course, project, department and other. Among these 7 categories, student, faculty, course and project are four most popular entity representing categories. The associated subset is typically WebKB4. CSTR is the dataset of the abstracts of technical reports published in the Department of Computer Science are a university. The dataset contains 476 abstracts, which are divided into four research areas: Natural language Processing (NLP), Robotics/Vision, Systems, and Theory. The top 1000 words by mutual information with class labels are selected from datasets WebKB4 and CSTR (http://feiwang03.googlepages.com/textdata.rar). We also use tr11, tr12, tr21 and tr23, they were downloaded and originally from TREC (http://www.cs.umn.edu/~han/data/tmdata.tar.gz/). The top 800 words by mutual information with class labels are selected. We keep 5 classes in tr11 and tr12. In order to avoid singular kernel matrix, we removed duplicate row or column entries in each dataset. tf-idf is also used to build document-word matrices. Table II shows the number of documents (n), words (d) and classes (k) in each dataset.

TABLE II. DESCRIPTION S OF DOCUMENT DATASETS

Datasets	*n*	*d*	*k*
News-1	480	882	4
News-2	480	843	4
CSTR	457	998	4
WebKB4	4178	1000	4
tr11	258	798	5
tr12	239	798	5
tr21	336	799	6
tr23	203	799	6

B. Peformance Measure

In the experiments, we set the number of clusters equal to the number of classes in the original datasets. To compare their performance, we compare document clusters generated by these algorithms with the true classes by computing the following two performance measures.

1) Normalized Mutual Information

The Normalized Mutual Information (NMI) [26, 31] is widely used for evaluating the quality of clusters. For two random variables X and Y, the NMI is defined as:

$$NMI(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}} \quad (22)$$

where $I(X, Y)$ is the mutual information between X and Y, while $H(X)$ and $H(Y)$ are entropies of X and Y respectively. The maximal possible value of NMI is 1. Given a certain clustering result, the NMI in Eq.(22) can be calculated as:

$$NMI = \frac{\sum_{l=1}^{k}\sum_{h=1}^{k} n_{l,h}\log\left(\frac{n\cdot n_{l,h}}{n_l \hat{n}_h}\right)}{\sqrt{\left(\sum_{l=1}^{k} n_l log\frac{n_l}{n}\right)\left(\sum_{h=1}^{k}\hat{n}_h\, log\frac{\hat{n}_h}{n}\right)}} \quad (23)$$

where n_l denotes the number of data contained in the cluster $C_l (1 \le l \le k)$, $\hat{n}_h$ is the number of data belonging to the h[th] class ($1 \le h \le k$), and $n_{l,h}$ denotes the number of data that are in the intersection between the cluster C_l and the h[th] class. We use the value from Eq.(23) to evaluate the quality of a given clustering.

2) Clustering Accuracy

To calculate the clustering accuracy, we count the number of documents correctly clustered corresponding to the original class it belongs to.

$$accuracy = \frac{number\ of\ correctly\ clustered\ dat\ objects}{total\ number\ of\ data\ objects} \quad (24)$$

Since the order of clusters may not match the order of the original classes, the clustering accuracy is defined as the maxmimal value among all possible class permutations.

3) Parameter Selection

There are five parameters in the algorithms, δ for kernel function, γ_r, γ_c, μ and the number of nearest neighbors m. The kernel function used in the experiments is Gaussian kernel for both rows and columns:

$$\mathrm{K}(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\delta^2}\right) \quad (25)$$

The values of δ for row and column kernels in Eq.(25) are searched through $\left\{\frac{\delta_0}{\sqrt{2}}, \delta_0, \sqrt{2}\delta_0, 2\sqrt{2}\delta_0, 4\delta_0\right\}$, where δ_0 is the mean norm of the given data x_i, $1 \le i \le n$. The kernel function used in the objective function must be non-singular, since a matrix inversion is involved in spectral relaxation. For example, vector space kernel [22] has the following formula:

$$\mathrm{K}(x_i, x_j) = x_i x_j^T \quad (26)$$

The kernel matrix can be singular and the eigen-problem of Eq.(18) cannot be stably solved. One could add some constant values to the diagonal elements of matrix B in Eq.(17), as B + αI, for some α > 0. Since the minimization problem in Eq.(13) with constraints in Eq.(15) is like the maximization problem of Linear Discriminant Analysis (LDA), both seek to solve a generalized eigenvalue problem,

one might follow alternative methods in LDA to avoid the singularity problem [8].

γ_r, γ_c and μ are all tuned through {0.1, 1, 10}. The number of nearest neighbors *m* is searched through {2, 3, 4, 5, 6, 7, 10, 12, 15} for News-1, News-2, CSTR, tr11, tr12, tr21 and tr23 datasets. For WebKB4, it is searched through {2, 5, 10, 15, 25, 35, 45, 50, 100}.

In kernel-based spectral clustering, the kernel function is also Gaussian kernel with δ searched in the same set as that in RCCM. For regularized spectral clustering, the parameter is search through {0.01, 0.1, 1, 10, 100}. For LLCA and GNMF, the number of nearest neighbor is also searched through in the same range as RCCM. In GNMF, the parameter λ is searched through {0.1, 1, 10, 100, 500, 1000}.

C. Numerical Results

We pick *k* eigenvectors corresponding to the smallest 2 - (k+1)th eigenvalues and find cluster assignment based on Eq.(20) and Eq.(3.21). Table III and Table IV summarize the accuracy rates and NMI values for clusters generated by all seven algorithms. The best results of these algorithms are listed. The best values are in bold and the second best values are in italic. We observe that RCCM completely outperforms traditional spectral co-clustering, ITCC, GNMF. It also shows the advantage over LLCA, kernel spectral clustering and regularized spectral clustering. We also show the performance (accuracy and normalized mutual information) of RCCM in tr11, tr12, tr21 and tr23 datasets over the different number of nearest neighbors in Figures 1, 2. Figure 3 shows the performance of RCCM with different values of δ_r for Guassian kernels for tr11, tr12, tr21 and tr23. δ_0 is the mean norm of rows in the figure. Figure 4 shows accuracy and NMI values of RCCM in tr11, tr12, tr21 and tr23 datasets over the changes of parameters γ and μ. In these tests, when we vary one parameter, we keep the other parameters at the optimal values. Both $_r$ and $_c$ are set to be the same in the testing, by doing so we can give both rows and columns equal weight in the algorithm. We use to refer to γ_r and γ_c in Figure 4. As we can see, RCCM is stable with respect to both parameters γ_r, γ_c and μ, δ_r value for kernel function is preferred to be around $4\delta_0$.

TABLE III. THE COMPARISON OF CLUSTERING ACCURACY FOR CLUSTERS GENERATED BY TESTED ALGORITHMS

Algorithms	News-1	News-2	WebKB4	CSTR	tr11	tr12	tr21	tr23
Spec-CoCls	0.4729	0.6062	0.5798	0.5624	0.8411	0.6653	0.6458	0.4039
ITCC	0.6104	0.4521	0.6032	0.7046	0.6860	0.6025	0.4524	0.3990
Kernel Spec	0.6583	0.5188	*0.7166*	0.5864	*0.9302*	0.7615	*0.7560*	0.5074
Reg. Spec	*0.7980*	*0.8188*	0.7025	*0.7812*	0.7868	*0.7699*	0.7500	*0.6010*
LLCA	0.6417	0.6062	**0.7319**	0.4517	0.7946	0.5732	0.7024	0.4975
GNMF	0.5583	0.5792	0.6216	0.7017	0.4961	0.4686	0.6756	0.3941
RCCM	**0.8021**	**0.8438**	0.6778	**0.8556**	**0.9380**	**0.8117**	**0.7857**	**0.6897**

TABLE IV. THE COMPARISON OF NMI VALUES FOR CLUSTERS GENERATED BY TESTED ALGORITHMS

Algorithms	News-1	News-2	WebKB4	CSTR	tr11	tr12	tr21	tr23
Spec-CoCls	0.4569	0.4570	0.4610	0.5127	0.7206	0.5290	*0.4469*	0.2850
ITCC	0.4452	0.1028	*0.5624*	*0.7482*	0.5932	0.5004	0.3112	0.3975
Kernel Spec	**0.6282**	0.3828	0.5113	0.6293	*0.8272*	*0.5959*	0.3374	0.4208
Reg. Spec	0.5735	*0.5544*	0.3701	0.6819	0.5430	0.5735	0.3816	*0.4532*
LLCA	0.4121	0.3521	0.4820	0.2911	0.5875	0.4403	0.4033	0.3898
GNMF	0.3428	0.3403	0.3925	0.5495	0.3490	0.1962	0.1604	0.1558
RCCM	*0.5932*	**0.6005**	**0.5845**	**0.8108**	**0.8427**	**0.7248**	**0.5151**	**0.4804**

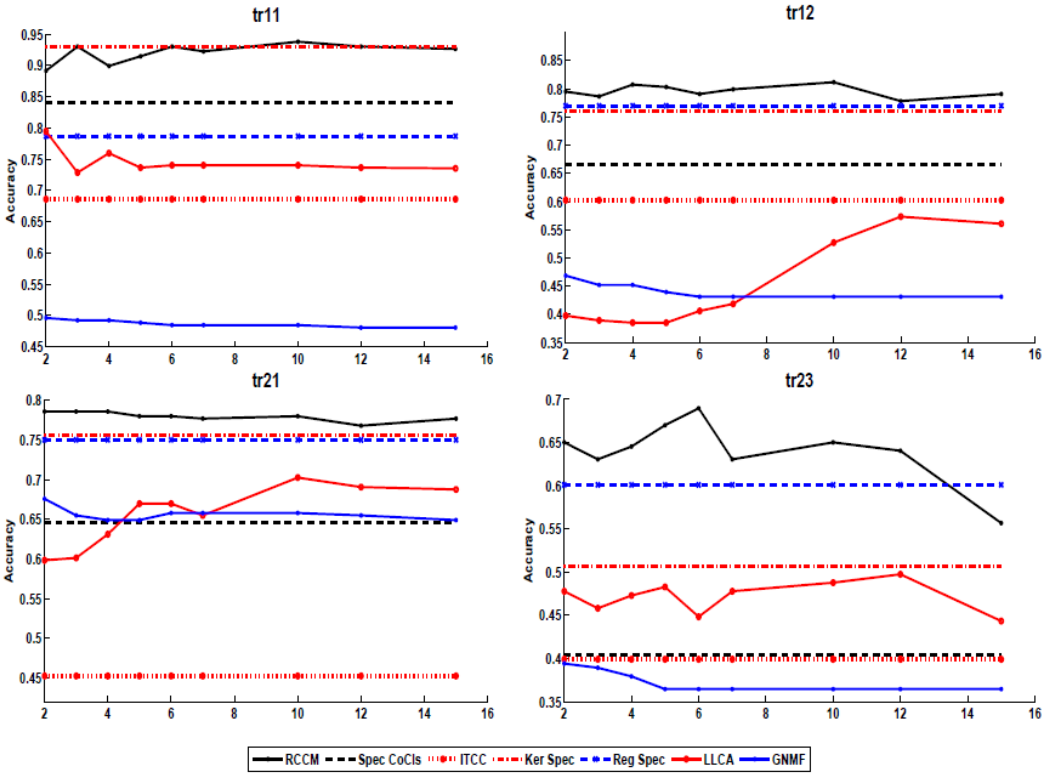

Figure 1. Clustering accuracy vs. number of neighbors

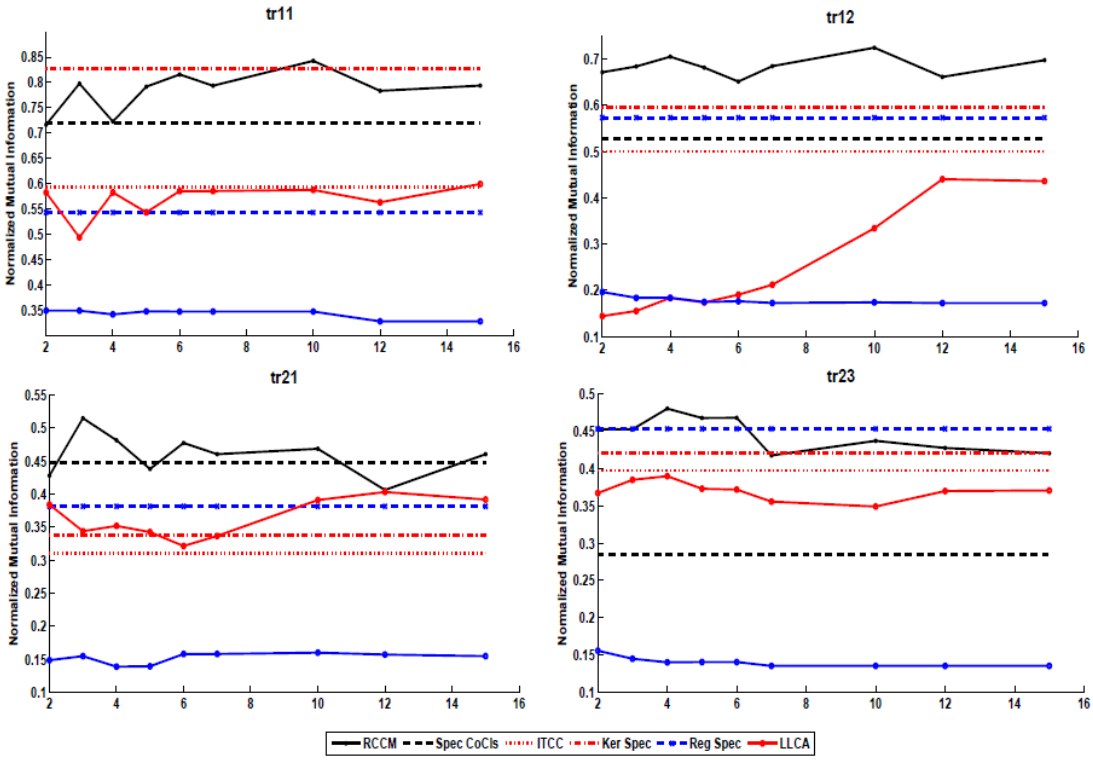

Figure 2. Normalized mutual information vs. number of neighbors

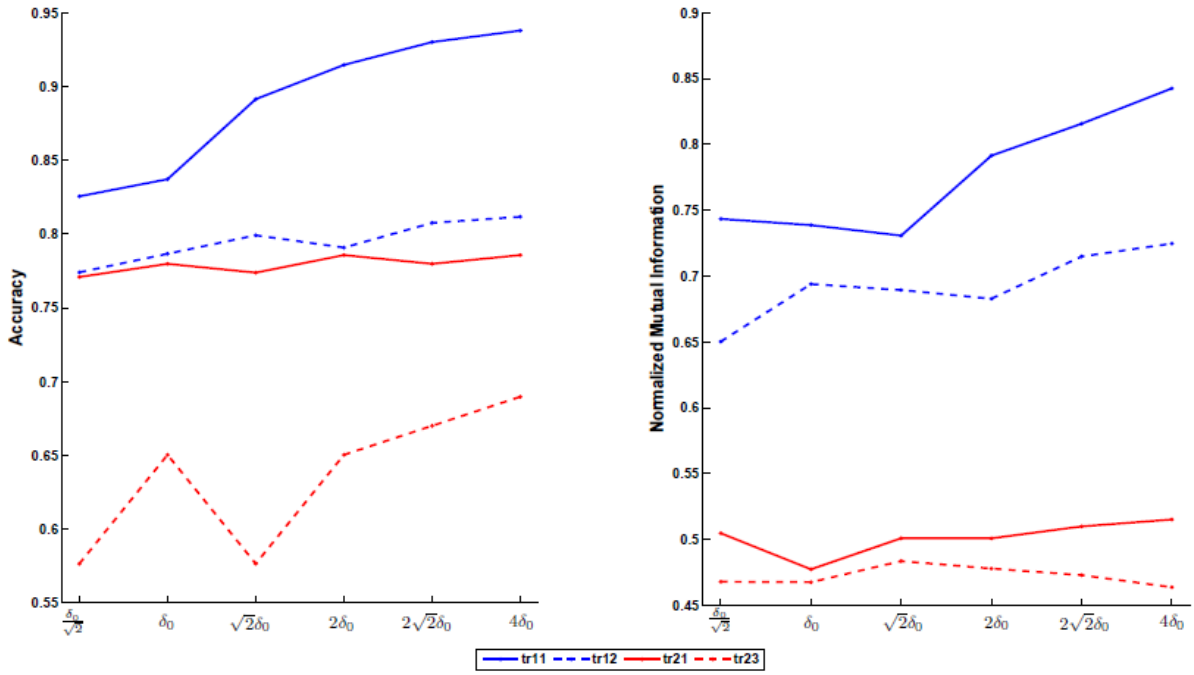

Figure 3. Normalized mutual information and cluster accuracy of RCCM vs. parameters δ

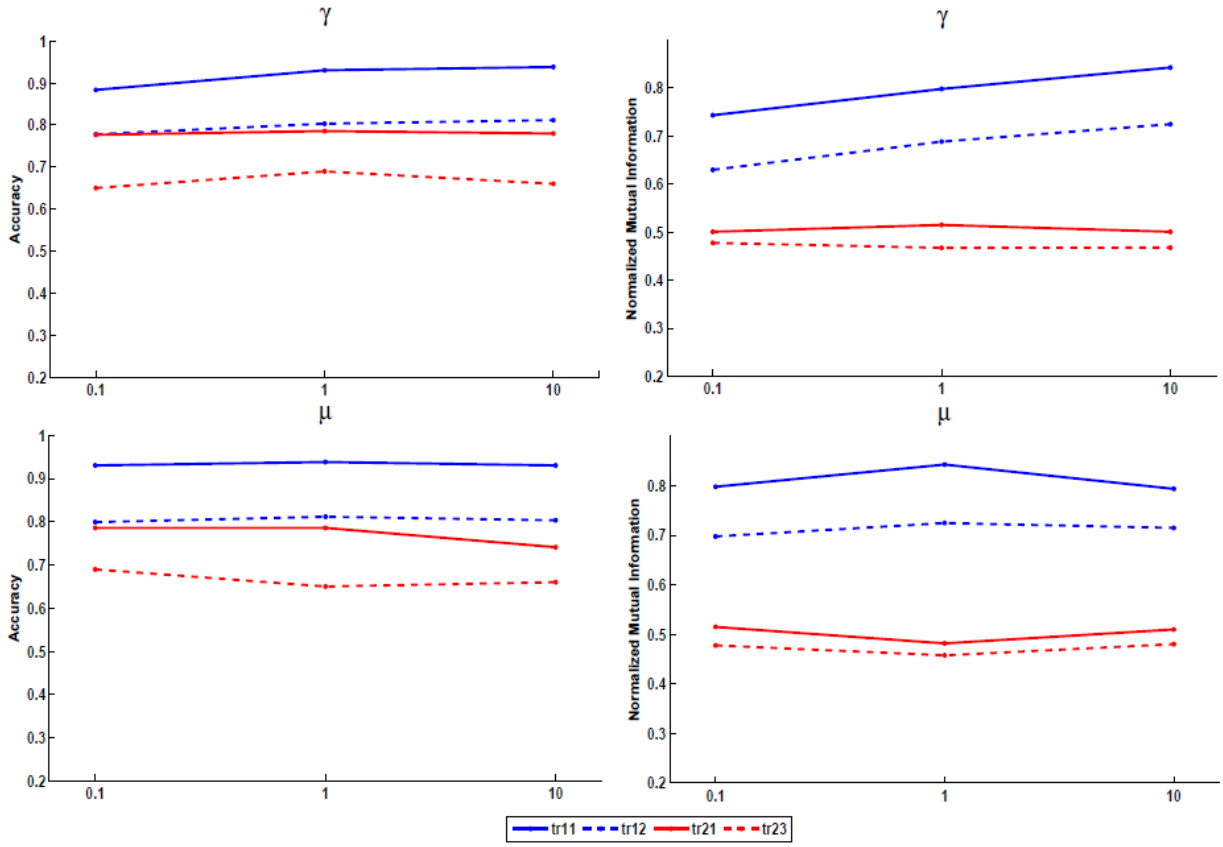

Figure 4. Normalized mutual information and cluster accuracy of RCCM vs. parameters μ and γ

D. Document-word Co-clustering

To check for the co-clustering quality, we calculate the mutual information [33] for each word in each category from News-1 and News-2. If a category *c* and *a* term *t* have probabilities *P(c)* and *P(t)*, then their mutual information *I(t, c)* is defined to be:

$$I(t,c) = log_2 \frac{p(t,c)}{p(t) \times p(c)} = log_2 \frac{p(t \wedge c)}{p(t) \times p(c)}$$

We then sort those words based on the mutual information values to each category. We check if words with higher value of mutual information have been correctly clustered with corresponding category. Table V and Table VI show the percentage of top 100 words correctly co-clustered with corresponding category in News-1 and News-2 from clusters generated by RCCM. We observe high rate of correct co-clustering. Table VII and Table VIII list top 15 words with highest mutual information in each category. This result shows that RCCMis able to reveal the hidden semantic relationship between documents and frequent words.

TABLE V. PERCENTAGE OF TOP 100 WORDS IN MUTUAL INFORMATION CO-CLUSTERED WITH CORRESPONDING CATEGORY IN NEWS-1 (REC.∗) FROM CLUSTERS GENERATED BY RCCM

Top n words	20	40	60	80	100
autos	0.95	0.875	0.8167	0.7625	0.74
motorcycles	1	0.95	0.9	0.9	0.84
baseball	1	0.95	0.8667	0.8625	0.86
hockey	0.95	0.9	0.8833	0.875	0.84

TABLE VI. PERCENTAGE OF TOP 100 WORDS IN MUTUAL INFORMATION CO-CLUSTERED WITH CORRESPONDING CATEGORY IN NEWS-2 (REC.∗) FROM CLUSTERS GENERATED BY RCCM

Top n words	20	40	60	80	100
crypt	1	1	1	0.9875	0.92
electronics	1	0.95	0.9167	0.9	0.84
med	1	1	0.9	0.875	0.81
space	1	0.975	0.9333	0.9	0.84

V. CONCLUSION

We have developed an unsupervised kernel co-clustering algorithm for rows and cloumns clustering. This method is an extension of manifold regularization and built on local information and bipartite graph partition. Then it is transformed to a generalized eigenvalue problem. The experimental evaluations with eight document datasets show better performance over current related clustering algorithms and the capability to co-cluster documents and related words simultaneously.

REFERENCES

[1] A. Anagnostopoulos, A. Dasgupta, and R. Kumar, Approximation algorithms for co-clustering, in Proc. of the 27th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2008,pp. 201–210.
[2] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha, A generalized maximum entropy approach to bregman co-clustering and matrix approximation, in Proc. of the 10th ACMInternational Conference on Knowledge Discovery and Data Mining, 2004, pp. 509–514.
[3] M. Belkin, Problems of Learning on Manifolds, PhD thesis, University of Chicago, 2003.
[4] M. Belkin and P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in Advances in Neural Information Processing Systems 14, 2001, pp. 585–591.
[5] M. Belkin, P. Niyogi, and V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, Journal of Machine Learning Research, 7 (2006), pp. 2399–2434.
[6] L. Bottou and V. Vapnik, Local learning algorithms, Neural Computation, 4 (1992), pp. 888–900.
[7] D. Cai, X. He, and J. Han, Document clustering using locality preserving indexing, IEEE Transactions on Knowledge and Data Engineering, 17 (2005), pp. 1624–1637.
[8] D. Cai, X. He, and J. Han, Srda: An efficient algorithm for large-scale discriminant analysis, IEEE Transactions on Knowledge and Data Engineering, 20 (2008), pp. 1–12.
[9] D. Cai, X. He, X. Wu, and J. Han, Non-negative matrix factorization on manifold, in Proceedings of the Eighth IEEE International Conference on Data Mining, 2008, pp. 63–72.
[10] F. R. K. Chung, Spectral Graph Theory, vol. 92 of Regional Conference Series in Mathematics, AMS, 1997.
[11] I. Dhillon, Co-clustering documents and words using bipartite spectral graph partitioning, in Proceedings of the seventh ACM SIGKDD, ACM, August 2001, pp. 269–274.
[12] I. Dhillon, S. Mallela, and D. Modha, Information-theoretic co-clustering, in Proceedings of the Ninth ACM SIGKDD, 2003, pp. 89–98.
[13] R. El-Yaniv and O. Souroujon, Iterative double clustering for unsupervised and semi-supervised learning, in Proc. of the 12th European Conference on Machine Learning, 2001, pp. 121–132.
[14] A. Gottlieb, Spectral coclustering (biclustering) matlab implementation. http://adios.tau.ac.il/SpectralCoClustering/.
[15] M. Gu, H. Zha, C. Ding, X. He, and H. S. J. Xia, Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering, Penn State University Technical Report, (2001).
[16] Q. Gu and J. Zhou, Co-clustering on manifolds, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, pp. 359–368.
[17] Y. Jia and C. Zhang, Local regularized least-square dimensionality reduction, in Proceedings of the 19th International Conference on Pattern Recognition, 2008, pp. 1–4.
[18] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, Spectral biclustering of microarray data: coclustering genes and conditions, Genome Research, 13 (2003), pp. 703–716.
[19] S. Roweis and L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science, 290 (2000), pp. 2323–2326.
[20] M. M. Shafiei and E. E. Milios, Latent dirichlet coclustering, in Proc. of the 6th International Conference on Data Mining, 2006, pp. 542–551.
[21] M. M. Shafiei and E. E. Milios, Model-based overlapping co-clustering, in Proc. of the 4th Workshop on Text Mining, 6th SIAM International Conference on Data Mining, 2006.
[22] J. Shawe-Taylor and N. Cristianini, Kernel methods for pattern analysis, Cambridge University Press, 2004.
[23] J. Shi and J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22 (2000), pp. 888–905.
[24] V. Sindhwani, J. Hu, and A. Mojsilovic, Regularized co-clustering with dual supervision, in Advances in Neural Information Processing Systems, 2008.

[25] S. Sra, S. Jegelka, and A. Banerjee, Approximation algorithms for bregman clustering co-clustering and tensor clustering, Technical Report of Max Planck Institute for Biological Cybernetics, No. 117 (2008).
[26] A. Strehl and J. Ghosh, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, The Journal of Machine Learning Research, 3 (2002), pp. 583–617.
[27] J. Sun, Z. Shen, H. Li, and Y. Shen, Clustering via local regression, in Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases, 2008, pp. 456–471.
[28] J. B. Tenenbaum, V. de Silva, and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science, 290 (2000), pp. 2319– 2323.
[29] N. Tishby, F. C. Pereira, and W. Bialek, The information bottleneck method, in Proc. of the 37- th Annual Allerton Conference on Communication, Control and Computing, 1999, pp. 368–377.
[30] F. Wang, C. Zhang, and T. Li, Regularized clustering for documents, in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 2007, pp. 95–102.
[31] M. Wu and B. Scholkopf, A local learning approach for clustering, in Advances in Neural Information Processing Systems 19, 2006.
[32] M. Wu, K. Yu, S. Yu, and B. Scholkopf, Local learning projections, in Proceedings of the 24th international conference on Machine learning, 2007, pp. 1039–1046.
[33] Y. Xu, G. Jones, J.-T. Li, and B.Wang, A study on mutual information-based feature selection for text categorization, Journal of Computational Information Systems, 3 (2007), pp. 1007–1012.
[34] D. Yan, L. Huang, and M. I. Jordan, Fast approximate spectral clustering, in Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining, 2009.
[35] H. Zeng, Z. Chen, and W. Ma, A unified framework for clustering heterogeneous web objects, in Proc. 3rd WISE, 2002, pp. 161–172.
[36] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, Learning with local and global consistency, in Advances in Neural Information Processing Systems 16, 2003.
[37] X. Zhu and J. Lafferty, Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning, in Proceedings of the 22nd international conference on Machine learning, 2005, pp. 1052–1059.