Locally Discriminative Coclustering

Lijun Zhang, *Student Member*, *IEEE*, Chun Chen, *Member*, *IEEE*, Jiajun Bu, *Member*, *IEEE*, Zhengguang Chen, Deng Cai, *Member*, *IEEE*, and Jiawei Han, *Fellow*, *IEEE*

Abstract—Different from traditional one-sided clustering techniques, coclustering makes use of the duality between samples and features to partition them simultaneously. Most of the existing co-clustering algorithms focus on modeling the relationship between samples and features, whereas the intersample and interfeature relationships are ignored. In this paper, we propose a novel coclustering algorithm named Locally Discriminative Coclustering (LDCC) to explore the relationship between samples and features as well as the intersample and interfeature relationships. Specifically, the sample-feature relationship is modeled by a bipartite graph between samples and features. And we apply local linear regression to discovering the intrinsic discriminative structures of both sample space and feature space. For each local patch in the sample and feature spaces, a local linear function is estimated to predict the labels of the points in this patch. The intersample and interfeature relationships are thus captured by minimizing the *fitting errors* of all the local linear functions. In this way, LDCC groups strongly associated samples and features together, while respecting the local structures of both sample and feature spaces. Our experimental results on several benchmark data sets have demonstrated the effectiveness of the proposed method.

Index Terms—Coclustering, clustering, bipartite graph, local linear regression.

1 Introduction

Data clustering is a fundamental topic in unsupervised learning, and becomes a common technique for data mining, information retrieval, pattern recognition, bioinformatics, etc. The goal of clustering is to partition the data points into clusters such that those within each cluster are more closely related to one another than points assigned to different clusters [1]. Typically, the data is formulated as a 2D matrix where one dimension represents samples, and the other represents features. Traditional clustering algorithms [2], [3], [4], [5] are one-sided in the sense that they only consider clustering samples based on their distributions on features, or vice versa.

Recently, coclustering has become a topic of significant interest in text mining [6], [7], [8], microarray analysis [9], [10], [11], and Collaborative Filtering (CF) [12], [13]. Instead of clustering one dimension of the data matrix, coclustering makes use of the duality between samples and features to partition both dimensions simultaneously. It has been shown that coclustering often yields impressive performance improvement over traditional one-sided clustering algorithms. More importantly, the resulting coclusters may reveal valuable insights about the data. For example, clustering documents and words simultaneously provides one way to describe the semantics, i.e., using the words in a cocluster to annotate itself; coclustering of genes expression data can be used to identify groups of genes showing similar activity patterns under a set of conditions; coclustering in CF helps to discover groups of users that exhibit highly correlated ratings on groups of items.

Most of the existing coclustering algorithms focus on modeling the relationship between samples and features, but with different strategies. The graph-based coclustering methods [7], [8] construct a bipartite graph to represent the relationship between samples and features. In the information theory-based coclustering methods [14], [15], [16], samples and features are treated as instances of two discrete random variables, and the joint probability distribution between them is used to encode the sample-feature relationship. In the matrix factorization-based coclustering techniques [17], [18], sample-feature relationship is modeled from the perspective of data reconstruction. Despite of their successes in making use of the sample-feature relationship, these algorithms fail to consider the intersample and interfeature relationships, which are essential for data clustering.

In this paper, we propose a novel coclustering algorithm named Locally Discriminative Coclustering (LDCC) to explore the sample-feature relationship as well as the intersample and interfeature relationships. Specifically, the sample-feature relationship is modeled by a bipartite graph [7], [8], where the edge signifies an association between a sample and a feature. And we apply local linear regression to discovering the intrinsic discriminative structures of both sample space and feature space. For each local patch in the sample and feature spaces, a local linear function is trained to predict the labels of the points belonging to this patch. The intersample and interfeature relationships are thus encoded in the local regression functions by minimizing the fitting errors over all the local patches. In this way, LDCC

- *L. Zhang, C. Chen, J. Bu, and Z. Chen are with the Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Cao Guangbiao Building, Yuquan Campus, Hangzhou 310027, China. E-mail: {zljzju, chenc, bjj, cerror}@zju.edu.cn.*
- *D. Cai is with the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, 388 Yu Hang Tang Road, Hangzhou 310027, China. E-mail: dengcai@cad.zju.edu.cn.*
- *J. Han is with the Department of Computer Science, University of Illinois at Urbana-Champaign, Room 2132, Siebel Center for Computer Science, 201 N. Goodwin Avenue, Urbana, IL 61801. E-mail: hanj@cs.uiuc.edu.*

Manuscript received 4 May 2010; revised 31 May 2010; accepted 16 Feb. 2011; published online 18 Mar. 2011.
Recommended for acceptance by B.C. Ooi.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2010-05-0265.
Digital Object Identifier no. 10.1109/TKDE.2011.71.

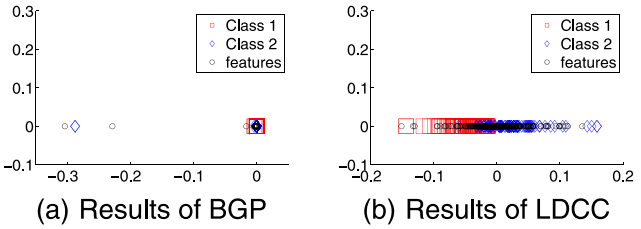


Fig. 1. The 1D projection results of one 2-Class subset selected from the WebKB corpus. To distinguish between features and samples, we plot them in different shape, size, and color.

groups strongly associated samples and features together, while respecting the local discriminative structures of both sample and feature spaces. We further develop an efficient computational scheme to solve the corresponding optimization problem.

Our optimization scheme first projects the samples and features into a common subspace and then performs coclustering in this subspace. The original Bipartite Spectral Graph Partitioning (BGP) [7], [8] methods also adopt this strategy, and the low-dimensional representation is computed directly from the data matrix. Thus, by visualizing the projection results of BGP and LDCC, we can see the effect of capturing the intersample and interfeature relationships intuitively. Fig. 1 shows the 1D projection results obtained by applying BGP and LDCC to one 2-Class subset of the WebKB corpus. As can be seen from Fig. 1a, BGP maps most of the features and samples together, which is clearly undesirable for coclustering. That is probably because BGP only considers the sample-feature relationship contained in the data matrix. When the data matrix is highly sparse, the projection results obtained from this limited information may be very unstable. From Fig. 1b, we can see that LDCC makes a big improvement compared with BGP. The two sample classes are almost separable, and features are distributed more evenly.

The outline of the paper is as follows: in Section 2, we provide a brief review of the related work. Our proposed Locally Discriminative Coclustering algorithm is introduced in Section 3. In Section 4, we compare our algorithm with the state-of-the-art clustering and coclustering algorithms. Finally, we provide some concluding remarks and suggestions for future work in Section 5.

Notation. Small letters (e.g., α) are used to denote scalars. Lower case bold letters (e.g., $\mathbf{w}$) are used to denote column vectors and $\|\cdot\|$ is used to denote the ℓ_2-norm of a vector. Capital letters (e.g., A) are used to denote matrices. We use $\mathrm{Tr}(\cdot)$ to denote the trace of a matrix, and $\|\cdot\|_F$ to denote the Frobenius norm of a matrix. Script capital letters (e.g., $\mathcal{X}$) are used to denote ordinary sets. Blackboard bold capital letters (e.g., $\mathbb{R}$) are used to denote number sets.

2 Related Work

The research literature on clustering is vast and mainly about one-sided clustering [19]. Although introduced quite early [20], coclustering receives much attention only in recent years due to its application to many practical problems, including text mining and microarray analysis. In this section, we briefly review three types of clustering or coclustering algorithms: graph partition-based, information theory-based, and matrix factorization-based.

Through constructing a similarity graph where vertices correspond to data points and edge weights represent degrees of similarity, clustering can be formulated as the problem of graph partitioning. Spectral partition methods have been used effectively for solving several graph partitioning objectives, such as ratio cut [21] and Normalized Cut (Ncut) [22]. In [7] and [8], the authors model the problem of coclustering documents and words as finding minimum cut vertex partitions in a bipartite graph between documents and words, which is then relaxed and solved by spectral method. Recently, a new method for partitioning the document-word bipartite graph called Isoperimetric Coclustering Algorithm (ICA) is proposed [6]. The ICA heuristically minimizes the ratio of the perimeter of the bipartite graph partition and the area of the partition under an appropriate definition of graph-theoretic area. In [23], a novel algorithm named Consistent Bipartite Graph Copartitioning is proposed for star-structured high-order heterogeneous data co-clustering. The bipartite graph model has also been successfully applied to cocluster genes and conditions in microarray analysis [10]. It is important to note that there are no intersample and interfeature edges in the bipartite graph model.

In information theory-based methods, samples, and features are treated as the instances of two random variables, of which the joint distribution can be empirically estimated from the data matrix. The problem of clustering samples or features can be viewed as the process of compressing the associated random variable. The Information Bottleneck (IB) Method [24] is a one-sided clustering algorithm which compresses one random variable so that the mutual information about the other is preserved as much as possible. Later, an agglomerative hard vision of the IB method is applied to clustering documents after clustering word, which is called Double Clustering (DC) [14]. Iterative Double Clustering (IDC) [15] extends DC to cluster documents and words iteratively. Both DC and IDC are heuristic procedures, whereas the Information-Theoretic Coclustering (ITCC) [16] clearly quantifies the loss in mutual information due to coclustering and presents an algorithm that reduces this loss function monotonically. A more generalized coclustering framework was proposed in [25] wherein any Bregman divergence can be used in the objective function, and various conditional expectation-based constraints can be supported.

Matrix factorization techniques have been widely studied and used for clustering and coclustering. The early work is mainly based on Singular Value Decomposition (SVD) or eigenvalue decomposition. Latent Semantic Indexing (LSI) [26] is a typical algorithm for one-sided document clustering, which first projects documents onto a lower dimensional subspace through SVD and then clusters documents in the reduced subspace. Nonnegative Matrix Factorization (NMF) [27] is a recently popularized technique which approximates the nonnegative data matrix by the product of two nonnegative matrices. Although the original motivation of NMF is to learn parts-based representations, it has been successfully

applied to one-sided clustering [2]. As an extension, Nonnegative Matrix Trifactorization (NM3F) is proposed for coclustering [17], [18]. In NM3F, the original data matrix X is decomposed into the production of three nonnegative (or two nonnegative and one unconstrained) matrices: $X = GSH^T$, where G gives row clusters and H gives column clusters. Graph regularized nonnegative matrix trifactorization for coclustering is introduced in [28]. To reduce the computational cost, a general coclustering framework named Coclustering based on Column and Row Decomposition (CRD) is proposed [29]. CRD does not require the whole data matrix to be in the main memory, and the execution time is linear in m (the number of samples) and n (the number of features).

3 Locally Discriminative Coclustering

The coclustering problem considered in this paper is formally defined as follows: given a nonnegative data matrix $X \in \mathbb{R}^{m \times n}$, we use the $\mathbf{x}_i^T$ to denote the ith row (sample) and $\mathbf{f}_j$ to denote the jth column (feature) in X. The goal of coclustering is to simultaneously group the samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subset \mathbb{R}^n$ and features $\{\mathbf{f}_1, \ldots, \mathbf{f}_n\} \subset \mathbb{R}^m$ into c coclusters. The clustering result of samples is represented by a Partition Matrix (PM) $G \in \{0,1\}^{m \times c}$, such that $G_{ir} = 1$ if $\mathbf{x}_i$ belongs to cluster r and $G_{ir} = 0$ otherwise. Similarly, the clustering result of features is represented by a PM $H \in \{0,1\}^{n \times c}$.

As discussed before, most of the existing coclustering algorithms only consider the sample-feature relationship. In the following, we introduce our Locally Discriminative Coclustering which makes use of the sample-feature relationship, as well as the intersample and interfeature relationships for coclustering. We begin with the discussion on modeling the sample-feature relationship.

3.1 Modeling the Sample-Feature Relationship

Similar to the previous approaches [7], [8], we model the relationship between samples and features using a bipartite graph. In the bipartite graph model, m samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ and n features $\{\mathbf{f}_1, \ldots, \mathbf{f}_n\}$ are two sets of vertices. An edge $<\mathbf{x}_i, \mathbf{f}_j>$ exists if and only if the jth feature is observed in the ith sample. And the edge weight is set to be X_{ij}, which represents the association between the sample $\mathbf{x}_i$ and the feature $\mathbf{f}_j$. It is natural to require that the labels of a feature and a sample should be the same if they are strongly associated.

Let $\mathbf{g}_i^T$ be the ith row of G, i.e., $\mathbf{g}_i$ is the indicator vector of sample $\mathbf{x}_i$. $\mathbf{h}_i^T$ is the ith row of H, i.e., the indicator vector of feature $\mathbf{f}_j$. Let $D^S \in \mathbb{R}^{m \times m}$ be the diagonal degree matrix of samples with $D^S_{ii} = \sum_{k=1}^{n} X_{ik}$, and $D^F \in \mathbb{R}^{n \times n}$ be the diagonal degree matrix of features with $D^F_{jj} = \sum_{k=1}^{m} X_{kj}$. In order to group strongly associated samples and features together, the following loss function can be used [30]:

$$\epsilon_1(G,H) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left\|\frac{\mathbf{g}_i}{\sqrt{D^S_{ii}}} - \frac{\mathbf{h}_j}{\sqrt{D^F_{jj}}}\right\|^2 X_{ij}. \quad (1)$$

By minimizing (1), we expect that if $\mathbf{x}_i$ and $\mathbf{f}_j$ are strongly associated (with a large X_{ij}), the indicator vectors of them should be the same. After some algebraic steps, (1) can be rewritten in the matrix form as follows:

$$\begin{aligned}\epsilon_1(G,H) &= \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\frac{\|\mathbf{g}_i\|^2}{D^S_{ii}} - \frac{2\mathbf{g}_i^T\mathbf{h}_j}{\sqrt{D^S_{ii}D^F_{jj}}} + \frac{\|\mathbf{h}_j\|^2}{D^F_{jj}}\right)X_{ij} \\ &= \sum_{i=1}^{m}\|\mathbf{g}_i\|^2 + \sum_{j=1}^{n}\|\mathbf{h}_j\|^2 - \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{2X_{ij}\mathbf{g}_i^T\mathbf{h}_j}{\sqrt{D^S_{ii}D^F_{jj}}} \\ &= m + n - 2\mathrm{Tr}\,(G^T(D^S)^{-1/2}X(D^F)^{-1/2}H).\end{aligned} \quad (2)$$

Since $m + n$ is a constant, the loss function in (2) can be simplified as

$$\tilde{\epsilon}_1(G,H) = -2\mathrm{Tr}(G^T X^N H), \quad (3)$$

where

$$X^N = (D^S)^{-1/2}X(D^F)^{-1/2}. \quad (4)$$

3.2 Modeling the Intersample and Interfeature Relationships

Inspired from recent developments in local learning [31], [32], [33], we propose to discover the intrinsic discriminative structure of both sample and feature spaces using local linear regression. The key mathematical derivations stated below stem from the work in [34], where Bach and Harchaoui apply global linear regression to clustering.

3.2.1 Local Linear Regression in the Sample Space

For each sample $\mathbf{x}_i$, we define the local patch $\mathcal{M}(\mathbf{x}_i)$ be the set containing $\mathbf{x}_i$ and its neighbor points, with the size m_i. We define $\mathcal{A}_i = \{k \mid \mathbf{x}_k \in \mathcal{M}(\mathbf{x}_i)\}$ to be the set containing the indices of samples in $\mathcal{M}(\mathbf{x}_i)$. Let $X_i \in \mathbb{R}^{m_i \times n}$ be the local data matrix consisting of samples in $\mathcal{M}(\mathbf{x}_i)$, that is, $X_i = [\mathbf{x}_k]^T$ for $k \in \mathcal{A}_i$. Let $G_i \in \mathbb{R}^{m_i \times c}$ be the local Partition Matrix of $\mathcal{M}(\mathbf{x}_i)$, that is, $G_i = [\mathbf{g}_k]^T$ for $k \in \mathcal{A}_i$. Since the local Partition Matrix G_i is a part of G, we can construct a selection matrix $S_i \in \{0,1\}^{m_i \times m}$ for each G_i such that

$$G_i = S_i G. \quad (5)$$

S_i is constructed as follows: $S_i = [\mathbf{e}_k]^T$ for $k \in \mathcal{A}_i$, where $\mathbf{e}_k$ is a m-dimensional vector whose kth element is one and all other elements are zero.

We consider fitting a multioutput linear function $f(X_i) = X_i W_i + \mathbf{1}_{m_i}\mathbf{b}_i^T$ for each local path $\mathcal{M}(\mathbf{x}_i)$ to model the relationship between X_i and G_i [31]. In this linear function, $\mathbf{1}_{m_i}$ is a m_i-dimensional vector of all ones, $W_i \in \mathbb{R}^{n \times c}$ is the coefficient matrix, and $\mathbf{b}_i \in \mathbb{R}^c$ is the intercept. Fitting this function can be mathematically formulated as

$$\min_{W_i, \mathbf{b}_i} \frac{1}{m_i}\left\|G_i - X_i W_i - \mathbf{1}_{m_i}\mathbf{b}_i^T\right\|_F^2 + \lambda\|W_i\|_F^2. \quad (6)$$

The penalty term $\lambda\|W_i\|_F^2$ is introduced to avoid overfitting [1].

Taking the first order partial derivatives of (6) with respective to W_i, $\mathbf{b}_i$ and requiring them to be zero, we get the optimal W_i^* and $\mathbf{b}_i^*$ [34]

$$W_i^* = \left(X_i^T \Pi_{m_i} X_i + m_i \lambda I\right)^{-1} X_i^T \Pi_{m_i} G_i, \tag{7}$$

$$\mathbf{b}_i^* = \frac{1}{m_i}\left(G_i^T - (W_i^*)^T X_i^T\right)\mathbf{1}_{m_i}, \tag{8}$$

where I is the identity matrix and $\Pi = I - \frac{1}{m_i}\mathbf{1}_{m_i}\mathbf{1}_{m_i}^T$ is the centering matrix.

Substituting the values of W_i^* and $\mathbf{b}_i^*$ into (6), we obtain the *fitting error* of the local function [34]

$$\begin{aligned}
J_i &= \frac{1}{m_i}\|G_i - X_i W_i^* - \mathbf{1}_{m_i}(\mathbf{b}_i^*)^T\|_F^2 + \lambda\|W_i^*\|_F^2 \\
&= \frac{1}{m_i}\left\|G_i - X_i W_i^* - \frac{\mathbf{1}_{m_i}\mathbf{1}_{m_i}^T}{m_i}(G_i - X_i W_i^*)\right\|_F^2 + \lambda\|W_i^*\|_F^2 \\
&= \frac{1}{m_i}\|\Pi(G_i - X_i W_i^*)\|_F^2 + \lambda\|W_i^*\|_F^2 \\
&= \frac{1}{m_i}\|\Pi\left(I - X_i\left(X_i^T\Pi X_i + m_i\lambda I\right)^{-1}X_i^T\Pi\right)G_i\|_F^2 \\
&\quad + \lambda\|\left(X_i^T\Pi X_i + m_i\lambda I\right)^{-1}X_i^T\Pi G_i\|_F^2 \\
&= \frac{1}{m_i}\mathrm{Tr}\left(G_i^T\left(\Pi - \Pi X_i\left(X_i^T\Pi X_i + m_i\lambda I\right)^{-1}X_i^T\Pi\right)^2 G_i\right) \\
&\quad + \lambda\mathrm{Tr}\left(G_i^T\Pi X_i\left(X_i^T\Pi X_i + m_i\lambda I\right)^{-2}X_i^T\Pi G_i\right) \\
&= \frac{1}{m_i}\mathrm{Tr}\left(G_i^T\left(\Pi - \Pi X_i\left(X_i^T\Pi X_i + m_i\lambda I\right)^{-1}X_i^T\Pi\right)G_i\right).
\end{aligned} \tag{9}$$

In the above derivations, we have used the fact that the centering matrix is idempotent, so that $\Pi = \Pi^k$ for $k = 1, 2, \ldots$. For each local patch $\mathcal{M}(\mathbf{x}_i)$, we define

$$L_i^S = \frac{1}{m_i}\left(\Pi - \Pi X_i\left(X_i^T\Pi X_i + m_i\lambda I\right)^{-1}X_i^T\Pi\right), \tag{10}$$

which characterizes the local discriminative structure of $\mathcal{M}(\mathbf{x}_i)$.

The fitting error J_i consists G_i as the variable and a good local G_i should give rise to minimal fitting error. In other words, we are looking for a local partition such that the clusters are most linearly separated, where the separability of clusters is measured through the minimum of the discriminative cost in (6) [34]. Then, it is naturally to require the global Partition Matrix G to minimize the summation of the fitting errors over all the local patches $\{\mathcal{M}(\mathbf{x}_i)\}_{i=1}^m$, which leads to the following loss function:

$$\begin{aligned}
\epsilon_2(G) &= \sum_{i=1}^m J_i = \sum_{i=1}^m \mathrm{Tr}\left(G_i^T L_i^S G_i\right) \\
&= \sum_{i=1}^m \mathrm{Tr}\left(G^T S_i^T L_i^S S_i G\right) = \mathrm{Tr}\left(G^T L^S G\right),
\end{aligned} \tag{11}$$

where

$$L^S = \sum_{i=1}^m \left(S_i^T L_i^S S_i\right). \tag{12}$$

The formulation of L_i^S in (10) involves the inverse of one $m \times m$ matrix, which is computationally expensive when the dimensionality is high. In the following, we use the Woodbury-Morrison formula [35] to derive a more efficient equation [34]

$$\begin{aligned}
L_i^S &= \frac{1}{m_i}\left(\Pi - \Pi X_i\left(X_i^T\Pi X_i + m_i\lambda I\right)^{-1}X_i^T\Pi\right) \\
&= \frac{1}{m_i}\Pi\left(I - \Pi X_i\left(X_i^T\Pi X_i + m_i\lambda I\right)^{-1}X_i^T\Pi\right)\Pi \\
&= \frac{1}{m_i}\Pi\left(I - I\Pi X_i\left(X_i^T\Pi I\Pi X_i + m_i\lambda I\right)^{-1}X_i^T\Pi I\right)\Pi \\
&= \frac{1}{m_i}\Pi\left(I + \frac{1}{m_i\lambda}\Pi_{m_i}X_i X_i^T\Pi\right)^{-1}\Pi \\
&= \lambda\Pi(m_i\lambda I + \Pi X_i X_i^T\Pi)^{-1}\Pi.
\end{aligned} \tag{13}$$

Using the above equation, we only need to inverse a $m_i \times m_i$ matrix, which would be much efficient, since the size of the local patch is usually very small.

3.2.2 Local Linear Regression in the Feature Space

Similarly, we can use the local linear regression to model the interfeature relationship.

For each feature $\mathbf{f}_j$, we define the local patch $\mathcal{N}(\mathbf{f}_j)$ be the set containing $\mathbf{f}_j$ and its neighbors, with the size n_j. And we define $\mathcal{B}_j = \{k \mid \mathbf{f}_k \in \mathcal{N}(\mathbf{f}_j)\}$ be the set containing the indices of features in $\mathcal{N}(\mathbf{f}_j)$. Let $F_j \in \mathbb{R}^{n_j \times m}$ be the local feature matrix consisting of features in $\mathcal{N}(\mathbf{f}_j)$. Let $H_j \in \mathbb{R}^{n_j \times c}$ be the local Partition Matrix of $\mathcal{N}(\mathbf{f}_j)$. Define a selection matrix $U_j = [\mathbf{e}_k]^T$ for $k \in \mathcal{B}_j$, where $\mathbf{e}_k$ is a n-dimensional vector whose kth element is one and all other elements are zero. We have

$$H_j = U_j H. \tag{14}$$

We also train a local linear function $g(F_j) = F_j V_j + \mathbf{1}_{n_j}\mathbf{a}_j^T$ for each local patch $\mathcal{N}(\mathbf{f}_j)$ to best approximate H_j. As before, we minimize the fitting errors of all the local functions to capture the interfeature relationship. Following the same derivations in Section 3.2.1, we obtain the following loss function of H:

$$\epsilon_3(H) = \mathrm{Tr}(H^T L^F H), \tag{15}$$

where

$$L^F = \sum_{j=1}^n \left(U_j^T L_j^F U_j\right), \tag{16}$$

$$\begin{aligned}
L_j^F &= \frac{1}{n_j}\Pi\left(I - F_j\left(F_j^T\Pi F_j + n_j\lambda I\right)^{-1}F_j^T\right)\Pi \\
&= \lambda\Pi\left(n_j\lambda I + \Pi F_j F_j^T\Pi\right)^{-1}\Pi.
\end{aligned} \tag{17}$$

3.3 The Objective

Define $P \in \{0,1\}^{(m+n)\times c}$ be the total Partition Matrix consisting of G and H

$$P = \begin{bmatrix} G \\ H \end{bmatrix}. \tag{18}$$

Combining (3), (11), and (15), the loss function of P is given by

$$\begin{aligned}\epsilon(P) &= \tilde{\epsilon}_1(G,H) + \alpha\epsilon_2(G) + \beta\epsilon_3(H)\\ &= \mathrm{Tr}(-2G^T X^N H + \alpha G^T L^S G + \beta H^T L^F H)\\ &= \mathrm{Tr}\left([G^T \quad H^T]\begin{bmatrix}\alpha L^S & -X^N\\ -(X^N)^T & \beta L^F\end{bmatrix}\begin{bmatrix}G\\ H\end{bmatrix}\right)\\ &= \mathrm{Tr}\left(P^T\begin{bmatrix}\alpha L^S & -X^N\\ -(X^N)^T & \beta L^F\end{bmatrix}P\right),\end{aligned} \tag{19}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are the tradeoff parameters.

With the loss function in (19), we define the LDCC problem as:

Definition 1. *Locally Discriminative Coclustering:*

$$\begin{aligned}\min_{P} \quad & \mathrm{Tr}(P^T L P)\\ \text{s.t.} \quad & L = \begin{bmatrix}\alpha L^S & -X^N\\ -(X^N)^T & \beta L^F\end{bmatrix}\\ & P \in \{0,1\}^{(m+n)\times c}, \ \ P\mathbf{1}_c = \mathbf{1}_{m+n},\end{aligned} \tag{20}$$

where X^N, L^S, and L^F are given by (4), (12), and (16), respectively.

3.4 The Algorithm

The LDCC problem is essentially a combinatorial optimization problem which is hard to solve. In order to solve it efficiently, we relax it according to the spectral clustering method in [3] and [36]. First, we map all the features and samples into a common low-dimensional subspace, and then cluster features and samples simultaneously in this subspace. Let Z be a $(m+n)\times r$ matrix whose rows give the low-dimensional embeddings of all the samples and features in a r-dimensional subspace. The optimal Z^* is obtained by solving the following problem:

$$\begin{aligned}\min_{Z} \quad & \mathrm{Tr}(Z^T L Z)\\ \text{s.t.} \quad & Z \in \mathbb{R}^{(m+n)\times r}, \ Z^T Z = I.\end{aligned} \tag{21}$$

Let $\mathbf{z}_1, \ldots, \mathbf{z}_r$ be the smallest eigenvectors of L ordered according to their eigenvalues. Then, the optimal solution Z^* of (21) is given by

$$Z^* = [\mathbf{z}_1, \ldots, \mathbf{z}_r]. \tag{22}$$

After normalization each row of Z^* [3], we perform Kmeans to group the samples and features into c coclusters.

In summary, the algorithm of LDCC is stated below.

1. Constructing the matrix L.
   a. Calculate X^N by normalizing the data matrix X according to (4).
   b. Find the k nearest neighbors of each sample, and calculate L^S according to (12).
   c. Find the k nearest neighbors of each feature, and calculate L^F according to (16).
   d. Construct L from X^N, L^S, and L^F according to (20).
2. Dimensionality reduction.
   a. Calculate the r smallest eigenvectors of L: $\mathbf{z}_1, \ldots, \mathbf{z}_r$.
   b. Form the matrix $Z^* = [\mathbf{z}_1, \ldots, \mathbf{z}_r]$ by stacking the eigenvectors in columns.
3. Coclustering in the low-dimensional subspace.
   a. Normalize each row of Z^* to have unit length.
   b. Cluster the samples and features into c coclusters via Kmeans.

3.5 Complexity Analysis of LDCC

The computational complexity of LDCC is dominated by the following steps:

- Find the k nearest neighbors of each sample, and calculate L^S according to (12).
  - $O(m^2 n)$ is used to calculate the pairwise distances between the m samples, and $O(m^2 \log m)$ is used for k-nearest neighbors finding for all the m samples.
  - $O(m(nk^2 + k^3))$ is used to calculate L_i^S according to (13) for all the m samples.
- Find the k nearest neighbors of each feature, and calculate L^F according to (16).
  - $O(n^2 m)$ is used to calculate the pairwise distances between the n features, and $O(n^2 \log n)$ is used for k-nearest neighbors finding for all the n features.
  - $O(n(mk^2 + k^3))$ is used to calculate L_j^F according to (17) for all the n features.
- Calculate the r smallest eigenvectors of L.
  - There are almost mk^2 and nk^2 nonzero elements in L_S and L^F, respectively. In practise, the data matrix X is usually very sparse, so the $(m+n)\times(m+n)$ dimensional matrix L is sparse too. As a result, the Lanczos algorithm [37] can be used to efficiently compute the first r eigenvectors within $O(qr(m+n)s)$, where q is number of iterations in Lanczos, and s is the number of nonzero elements per row of L.

The total cost for LDCC is $O((mn+qrs)(m+n) + m^2 \log m + n^2 \log n + (mn+mk+nk)k^2)$.

The main memory cost of LDCC is to store the pairwise distance matrices and the sparse matrix L. Thus, the memory complexity is $O(m^2 + n^2 + s(m+n))$.

4 Experiments

In this section, we perform text and gene expression clustering (coclustering) experiments to show the effectiveness of LDCC.

4.1 Experimental Design

We compare our method with the following five methods:

1. Kmeans on the original data matrix (Kmeans) [4].
2. Normalized cut [22].
3. Bipartite spectral graph partitioning [7], [8].
4. Information-theoretic coclustering [16].[1]
5. Dual regularized coclustering (DRCC) [28].

In the above methods, Kmeans and NCut are one-sided clustering algorithms, while BGP, ITCC, DRCC, and LDCC

1. http://www.cs.utexas.edu/users/dml/Software/cocluster.html.

are coclustering algorithms. Note that the procedures for solving Kmeans and ITCC can only find the local optimum. In the experiments, we ran both Kmeans and ITCC 10 times with different random starting points and the best result in terms of their objective functions was recorded.

4.1.1 Evaluation Metric

In practice, the ground truth about the feature clusters is usually unknown. Thus, we evaluate the result of sample clustering in our experiments. The sample clustering performance is evaluated by comparing the label obtained from the clustering or coclustering algorithms with that provided by the data set. Two metrics, the accuracy (AC) and the normalized mutual information ($\overline{MI}$), are used to measure the clustering performance [2], [38]. Given a sample x_i, let p_i and q_i be the obtained cluster label and the label provided by the data set, respectively. The AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{m} \delta(q_i, \mathrm{map}(p_i))}{m}, \tag{23}$$

where m is the total number of documents, $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and map(p_i) is the permutation mapping function that map each cluster label p_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm [39].

Let C denote the set of clusters provided by the data set and C' obtained from our algorithm. Their mutual information metric $MI(C, C')$ is defined as following:

$$MI(C, C') = \sum_{c_i \in C,\, c'_j \in C'} p(c_i, c'_j). \log_2 \frac{p(c_i, c'_j)}{p(c_i).p(c'_j)}, \tag{24}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a sample arbitrarily selected from the data set belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected sample belongs to the cluster c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information $\overline{MI}$ as follows:

$$\overline{MI} = \frac{MI(C, C')}{\max(H(C), H(C'))}, \tag{25}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $\overline{MI}$ takes values between 0 and 1.

4.1.2 Parameter Settings

There are several parameters to be tuned in each clustering or coclustering algorithm considered in our experiments. In order to compare these algorithms fairly, we run them under different parameter settings, and report the best result.

In Kmeans and Ncut, the number of sample clusters is set to the true number of classes for all the data sets. In BGP, ITCC, DRCC, and LDCC, the number of sample clusters and feature clusters are both set to the true number of classes for all the data sets.

In Ncut [22], the similarity between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is computed with Heat kernel: $w_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t})$. The parameter t is searched from the grid: $\{1e{-}2, 1e{-}1, 1, 1e1, 1e2\}$.

In DRCC, two k-nearest neighbor graphs are constructed to encode the manifold structure in the data space and feature space. As in [28], the neighborhood size of the sample graph is set to be the same as that of the feature graph. k is searched from the grid: $\{1, 2, \ldots, 10\}$. The two regularization parameters in DRCC are set to be the same, and searched from the grid: $\{1e{-}2, 1e{-}1, 1, 1e1, 1e2\}$.

In LDCC, the regularization parameter λ for local linear regression is set to 1, and the dimensionality r is searched from the grid: $\{5, 10, \ldots, 50\}$. The other parameter settings of LDCC are the same as DRCC.

4.2 Data Sets

Three text corpora and two gene expression data sets are used in our experiments.

20 Newsgroups.[2] The 20 Newsgroups corpus is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. On this data set, we select 2,000 words with the largest contribution to the mutual information between the words and the documents [14], and then remove the empty documents.

WebKB.[3] The WebKB corpus contains web pages collected from computer science departments of various universities in January 1997. The 8,282 pages were manually classified into the following seven categories: student, faculty, staff, department, course, project, and other. We select the top 1,000 words by mutual information for this data set.

TechTC-100.[4] The Technion Repository of Text Categorization Data sets (TechTC) provides a large number of diverse test collections for use in text categorization research. We use the TechTC-100 corpus, which contains 100 binary text data sets. Each data set in TechTC-100 consists of a total of 150 to 200 documents from two Open Directory Project (ODP) categories. In our experiments, we select the top 2,000 words by mutual information for each data set.

Leukemia.[5] Leukemia data set is a benchmark in gene expression analysis [40]. It contains 72 samples and 7,129 genes. Each sample belongs to either Acute Lymphoblastic Leukemia (ALL) or Acute Myeloid Leukemia (AML). We use the subset provided by Brunet et al. [41], which consists of 38 bone marrow samples (27 ALL samples and 11 AML samples). We screen out genes with $\max/\min < 15$ and $\max - \min < 500$, leaving a total of 1,999 genes.

Medulloblastoma.[6] This gene expression data set [42] is collected from childhood brain tumors known as medulloblastomas. The pathogenesis of these tumors is not well understood, but it is generally accepted that there are two known histological subclasses: classic and desmoplastic. We use the subset provided by Brunet et al. [41], which consists

2. http://people.csail.mit.edu/jrennie/20Newsgroups/.
3. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/.
4. http://techtc.cs.technion.ac.il/.
5. http://www.broadinstitute.org/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=89.
6. http://www.broadinstitute.org/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=89.

TABLE 1
Performance Comparisons on the 20 Newsgroups Corpus (mean ± std-dev)

c	Accuracy (%)					
	Kmeans	NCut	BGP	DRCC	ITCC	LDCC
2	80.5 ± 14.8	78.8 ± 14.5	79.6 ± 16.9	79.7 ± 15.1	82.6 ± 11.6	**88.5 ± 9.9**
3	82.3 ± 9.2	75.3 ± 10.3	78.4 ± 11.2	83.1 ± 10.3	83.4 ± 10.5	**91.6 ± 3.1**
4	68.8 ± 10.6	59.4 ± 8.5	71.0 ± 12.5	68.9 ± 12.1	64.9 ± 13.0	**82.6 ± 8.4**
5	65.6 ± 6.6	59.9 ± 6.6	63.5 ± 5.8	64.1 ± 6.3	65.1 ± 5.9	**80.4 ± 7.0**
6	62.6 ± 6.6	54.6 ± 7.7	64.3 ± 7.4	61.9 ± 5.2	63.6 ± 6.7	**78.3 ± 7.6**
7	57.8 ± 5.0	49.4 ± 6.2	62.1 ± 8.9	56.5 ± 5.0	56.0 ± 7.5	**79.2 ± 6.8**
8	54.2 ± 9.6	49.0 ± 7.5	60.2 ± 10.0	53.7 ± 6.2	54.0 ± 7.7	**70.6 ± 8.5**
9	51.4 ± 6.7	44.8 ± 4.2	61.4 ± 7.6	50.8 ± 5.2	52.9 ± 6.0	**73.9 ± 6.0**
10	48.7 ± 5.9	43.2 ± 3.8	55.9 ± 4.7	46.9 ± 4.0	50.0 ± 4.4	**70.3 ± 5.9**
c	Normalized Mutual Information (%)					
	Kmeans	NCut	BGP	DRCC	ITCC	LDCC
2	40.8 ± 27.4	36.2 ± 26.1	41.1 ± 32.3	39.0 ± 27.1	40.4 ± 26.6	**54.9 ± 24.0**
3	55.6 ± 12.8	44.1 ± 12.6	55.6 ± 13.0	56.4 ± 13.8	58.9 ± 12.4	**72.1 ± 8.1**
4	43.4 ± 13.0	32.8 ± 6.6	49.2 ± 14.8	43.8 ± 14.3	44.9 ± 13.5	**59.7 ± 13.2**
5	45.0 ± 7.0	37.4 ± 7.3	49.9 ± 6.3	43.2 ± 6.1	50.4 ± 5.9	**60.2 ± 7.3**
6	43.6 ± 5.4	34.1 ± 6.3	51.7 ± 6.7	41.9 ± 5.1	49.9 ± 6.7	**60.3 ± 7.8**
7	42.1 ± 3.9	33.3 ± 3.8	50.2 ± 5.4	40.1 ± 4.7	47.6 ± 4.0	**62.5 ± 6.5**
8	40.1 ± 8.2	34.0 ± 7.1	50.1 ± 9.1	38.8 ± 5.7	45.4 ± 7.2	**56.9 ± 8.2**
9	39.5 ± 5.8	32.4 ± 4.4	52.0 ± 5.3	38.0 ± 4.9	46.8 ± 5.0	**60.0 ± 5.7**
10	37.7 ± 5.0	31.3 ± 3.6	49.4 ± 2.8	35.0 ± 3.2	44.4 ± 3.1	**56.9 ± 3.7**

The best result is in bold.

of 25 classic samples and 9 desmoplastic samples. We screen out genes with $\max/\min < 15$ and $\max - \min < 500$, leaving a total of 1,710 genes.

4.3 Experimental Results

4.3.1 Performance on Balanced Data Sets

In this section, we compare the performance of different algorithms on balanced data sets, where all the sample classes have the same size.

The first experiment is done on the 20 Newsgroups corpus. The evaluations were conducted with different number of document classes c, ranging from 2 to 10. At each run of the test, c document classes are randomly selected from the whole corpus and 300 documents are randomly selected from each selected class. For each given class number c, 10 test runs are conducted, and the average performance is computed over these 10 tests. Table 1 shows the average clustering performance, as well as the standard deviation, for each algorithm. As can be seen, our LDCC algorithm significantly outperforms the other clustering or coclustering algorithms in all the cases.

Although the WebKB corpus is unbalanced, we generate balanced subsets by choosing the same number of documents from each document class. We perform clustering or coclustering experiments with $c = 2, \ldots, 7$ document classes. At each run of the test, c document classes are randomly selected and 100 documents are randomly selected from each selected class. As before, 10 test runs are conducted for each c and the average performance is reported. The clustering results on the balanced WebKB data set are shown in Table 2. On this data set, LDCC still performs the best in most cases. Compared with the results on the 20 Newsgroup corpus, the advantage of LDCC is less obvious on this corpus. That is probably because the WebKB corpus is more difficult to be clustered.

From Tables 1 and 2, we can see that in general the accuracy keeps decreasing as the number of classes c increases. However, the Normalized Mutual Information

TABLE 2
Performance Comparisons on the Balanced WebKB Corpus (mean ± std-dev)

c	Accuracy (%)					
	Kmeans	NCut	BGP	DRCC	ITCC	LDCC
2	83.5 ± 8.8	**85.2 ± 8.3**	63.7 ± 14.3	83.1 ± 7.7	74.8 ± 10.9	83.2 ± 9.2
3	68.4 ± 12.9	71.2 ± 12.1	53.5 ± 7.0	68.9 ± 12.2	67.6 ± 8.0	**74.2 ± 9.5**
4	67.2 ± 7.7	67.2 ± 8.6	62.4 ± 5.3	67.8 ± 7.5	65.8 ± 7.1	**72.5 ± 7.6**
5	57.7 ± 6.6	60.7 ± 7.0	50.9 ± 5.3	57.3 ± 7.6	55.1 ± 7.2	**62.8 ± 5.9**
6	50.6 ± 5.1	56.4 ± 5.9	45.7 ± 1.9	51.0 ± 4.8	48.4 ± 4.0	**56.8 ± 5.0**
7	46.4 ± 1.8	43.5 ± 0.2	44.4 ± 0.9	45.3 ± 3.3	45.8 ± 1.9	**55.0 ± 0.6**
c	Normalized Mutual Information (%)					
	Kmeans	NCut	BGP	DRCC	ITCC	LDCC
2	40.1 ± 19.3	**43.7 ± 22.6**	16.0 ± 22.7	38.6 ± 17.9	27.2 ± 20.4	40.7 ± 23.3
3	37.0 ± 16.2	36.3 ± 14.7	21.5 ± 15.4	36.4 ± 14.6	34.3 ± 12.0	**39.6 ± 17.3**
4	43.7 ± 9.6	40.6 ± 9.3	42.9 ± 5.2	41.8 ± 9.8	44.7 ± 8.2	**46.3 ± 12.6**
5	35.2 ± 6.7	37.8 ± 7.0	31.2 ± 6.1	35.2 ± 6.8	35.6 ± 6.5	**40.6 ± 6.8**
6	32.8 ± 5.0	35.1 ± 6.0	28.1 ± 3.6	32.7 ± 5.0	32.1 ± 3.6	**36.7 ± 5.3**
7	31.1 ± 2.2	28.6 ± 0.1	28.1 ± 0.4	30.5 ± 2.0	32.2 ± 2.2	**38.0 ± 1.1**

The best result is in bold.

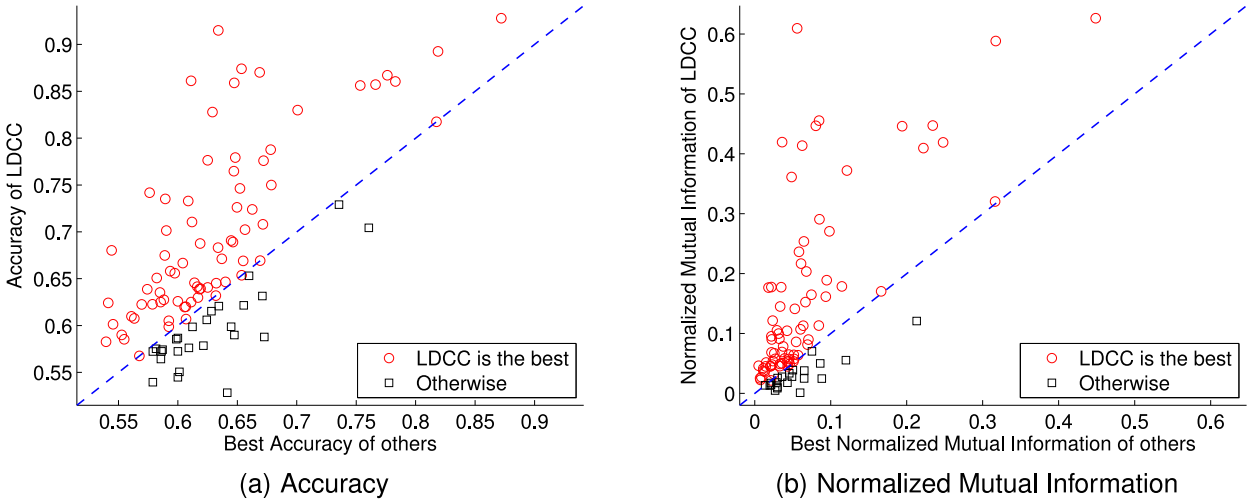


Fig. 2. Performance comparisons on the TechTC-100 corpus. We plot the result of LDCC in y-axis and best result of the other methods in x-axis.

TABLE 3
Performance Comparisons on the Unbalanced WebKB Corpus (mean ± std-dev)

c	Accuracy (%)					
	Kmeans	NCut	BGP	DRCC	ITCC	LDCC
2	66.8 ± 14.5	68.2 ± 14.3	68.9 ± 12.4	63.0 ± 12.7	61.7 ± 12.8	**73.8 ± 10.2**
3	51.5 ± 11.0	56.6 ± 9.9	57.7 ± 13.2	51.6 ± 9.6	53.6 ± 9.4	**58.1 ± 7.7**
4	45.6 ± 4.7	49.4 ± 5.2	46.2 ± 6.8	46.7 ± 3.0	48.5 ± 4.5	**53.4 ± 3.9**
5	45.9 ± 10.5	43.9 ± 9.8	46.7 ± 13.0	43.7 ± 8.2	43.8 ± 5.0	**52.4 ± 10.7**
6	40.2 ± 6.6	41.1 ± 4.7	41.4 ± 5.9	39.6 ± 4.7	40.6 ± 5.5	**47.7 ± 4.9**
7	38.7 ± 3.2	38.5 ± 0.0	39.1 ± 1.1	35.6 ± 1.6	36.0 ± 2.1	**43.2 ± 1.2**

c	Normalized Mutual Information (%)					
	Kmeans	NCut	BGP	DRCC	ITCC	LDCC
2	17.1 ± 18.6	**20.0 ± 19.5**	13.9 ± 21.4	14.9 ± 16.3	11.7 ± 17.5	18.5 ± 14.6
3	16.3 ± 12.8	19.2 ± 11.6	15.7 ± 13.2	18.0 ± 12.9	16.9 ± 8.7	**20.0 ± 8.9**
4	16.1 ± 4.4	18.5 ± 4.5	16.9 ± 2.9	15.5 ± 3.9	19.5 ± 4.4	**23.0 ± 3.4**
5	22.7 ± 12.2	19.4 ± 8.9	24.3 ± 10.6	19.6 ± 8.5	20.7 ± 6.1	**25.4 ± 9.4**
6	20.1 ± 6.4	19.4 ± 4.5	22.2 ± 5.6	19.3 ± 4.6	20.7 ± 4.9	**24.6 ± 4.3**
7	19.5 ± 0.6	19.1 ± 0.0	22.5 ± 0.2	18.5 ± 1.4	19.7 ± 1.3	**23.2 ± 1.5**

The best result is in bold.

($\overline{MI}$) does not have a clear pattern. The behavior of $\overline{MI}$ is quite complex and depends on the specific data set. For example, in [2], we can see that $\overline{MI}$ decreases on the TDT2 corpus and increases on the Reuters corpus, when c increases.

The TechTC-100 corpus contains 100 balanced binary data sets, and we perform clustering or coclustering experiments on each data set. The performance comparisons on the TechTC-100 corpus are shown in Fig. 2, where we plot the result of LDCC in y-axis and best result of the other methods in x-axis. Then, the number of points in the upper triangle is just the number of data sets on which LDCC performs the best. In terms of accuracy, LDCC performs the best on 74 sets. And in terms of $\overline{MI}$, LDCC performs the best on 78 sets. From Fig. 2b, we can see that the $\overline{MI}$ of many data sets is less than 0.1, which means this text corpus is very hard to be clustered. However, our algorithm still performs the best in most cases on this corpus.

4.3.2 Performance on Unbalanced Data Sets

In the following, we evaluate the performance of all the methods on unbalanced data sets, where the sizes of sample classes are highly skewed.

The WebKB corpus is a unbalanced data set, where the size of the document class ranges from 137 to 3,728. As before, we perform clustering or coclustering experiments with $c = 2, \ldots, 7$ document classes. At each run of the test, we select c document classes randomly and then select 10 percent documents from each selected class. In this way, the randomly generated subsets are highly unbalanced. Ten test runs are conducted for each c and the average performance is reported. The clustering results on the unbalanced WebKB data set are shown in Table 3. Compared with the results in Table 2, the performance of all the methods decreases in this experiments. Thus, the unbalanced data sets are more difficult to be clustered. Nevertheless, our LDCC still performs the best on most cases.

Finally, we show the clustering performance on the two gene expression data sets in Table 4. This two data sets are

TABLE 4
Clustering Performance on the Two Gene Expression Data Sets

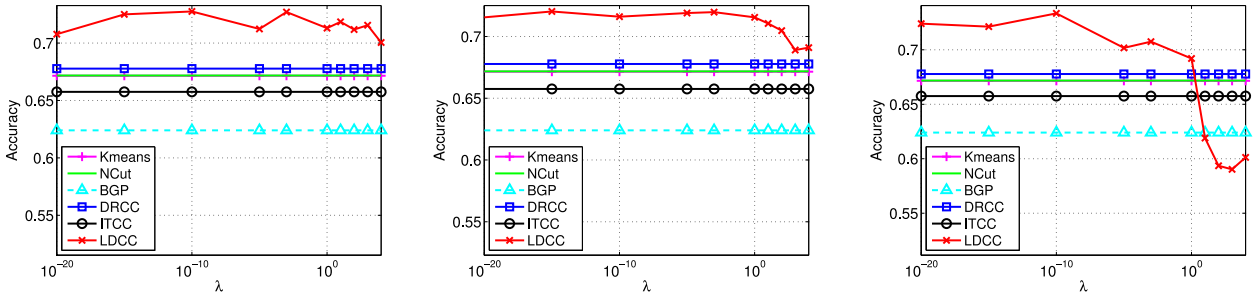
Data Sets	Accuracy (%)					
	Kmeans	NCut	BGP	DRCC	ITCC	LDCC
Leukemia	50.0	78.9	**94.7**	89.5	89.5	**94.7**
Medulloblastoma	64.7	76.5	61.8	58.8	55.9	**82.4**

Data Sets	Normalized Mutual Information (%)					
	Kmeans	NCut	BGP	DRCC	ITCC	LDCC
Leukemia	13.8	37.7	**70.8**	47.7	47.7	**70.8**
Medulloblastoma	0.1	**33.5**	0.0	0.3	4.2	26.4

The best result is in bold.

TABLE 5
Coclustering Results of LDCC on the 20 Newsgroups Corpus

		atheism	hardware	motorcycles	guns
	Document Cluster 1	232	1	1	5
	Document Cluster 2	0	223	2	1
	Document Cluster 3	14	26	245	32
	Document Cluster 4	4	0	2	212
Word Cluster 1	born, intended, word, book, approach, evil, words, imply, written, became				
Word Cluster 2	files, memory, vlb, cards, cache, vesa, motherboard, eisa, slots, performance				
Word Cluster 3	ride, design, mph, sun, lady, rider, like, honda, riding, hawk				
Word Cluster 4	tax, carried, only, animal, raid, mere, firing, nra, members, president				



(a) $\alpha = 0.1$ (b) $\alpha = 1$ (c) $\alpha = 10$

Fig. 3. The impacts of the parameters λ and α on the performance of LDCC.

rather challenging for coclustering algorithms, since they contain a small number of samples but large amount of features. As a result, BGP groups all the samples into the same cocluster on the Medulloblastoma data set, thus its $\overline{MI}$ is 0 on this data set. In contrast, our LDCC is the winner for most of the cases, which verifies that the intrasample and interfeature relationships are essential for coclustering.

4.3.3 A Case Study of Feature Clusters

In this section, we provide one case study of feature clusters. We form one text data set by choosing four classes ("atheism," "hardware," "motorcycles," and "guns") from the 20 Newsgroups corpus.

Table 5 summarizes the results of applying LDCC to this data set. The top of the table is the confusion matrix, from which we can see that LDCC is able to recover the original classes. Since LDCC partitions documents and words simultaneously, there is one associated word cluster for each document cluster. We list the words near the center of each word cluster in the bottom of Table 5. It should be observed that most of these words are consistent with the "concept" of the associated document cluster.

4.3.4 Parameter Selection

In LDCC, we apply ridge regression at each local patch to capturing the intersample (or intersample) relationship, where λ appears as a regularization parameter. Besides, there is another parameter α which is used to control the importance of intersample and interfeature relationships. In the following, we examine the impacts of the two parameters on the performance of LDCC.

The performance of LDCC under different settings of λ and α is evaluated on the WebKB corpus. The number of document classes c is set to 4, and other experimental settings are the same as these in Section 4.3.1. For brevity, we just show how the accuracy of LDCC varies with the parameter λ and α. Fig. 3 plots the accuracy versus the value of λ with $\alpha = \{0.1, 1, 10\}$. As can be seen, the performance of LDCC is quite stable with respect to λ as long as it is smaller than certain threshold. Comparing the results under different settings of α, we can see that LDCC is also insensitive to α.

5 Conclusions and Future Work

In this paper, we proposed a novel coclustering algorithm named Locally Discriminative Coclustering which explores sample-feature, intersample, and interfeature relationships simultaneously. The sample-feature relationship is modeled by a bipartite graph, while the intersample and interfeature relationships are captured by local linear regressions. The results of the coclustering experiment are very promising.

The idea of local learning can be used to extend the existing coclustering algorithms, such as the information theory-based [16], [25] and matrix factorization-based algorithms [17], [18]. And we will investigate this in our future work. Although clustering is inherently an unsupervised learning problem, sometimes a small set of labeled samples (features) might be available. Thus, the extension of LDCC to incorporate prior knowledge is another research topic. Furthermore, more efficient optimization methods for LDCC will be considered.

Acknowledgments

This work was supported by Scholarship Award for Excellent Doctoral Student granted by Ministry of Education, National Key Technology R&D Program of China (2008BAH26B00), Program for New Century Excellent Talents in University (NCET-09-0685), and National Basic Research Program of China (973 Program) under Grant 2011CB302206.

REFERENCES

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning,* Springer Series in Statistics. Springer, 2009.

[2] W. Xu, X. Liu, and Y. Gong, "Document Clustering Based on Non-Negative Matrix Factorization," *Proc. 26th Ann. Int'l ACM SIGIR Conf. Research and Development in Informaion Retrieval,* pp. 267-273, 2003.

[3] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Proc. Advances in Neural Information Processing Systems,* pp. 849-856, 2002.

[4] J. McQueen, "Some Methods of Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math. Statistics and Probability,* pp. 281-297, 1967.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining,* pp. 226-231, 1996.

[6] M. Rege, M. Dong, and F. Fotouhi, "Co-Clustering Documents and Words Using Bipartite Isoperimetric Graph Partitioning," *Proc. Sixth Int'l Conf. Data Mining,* pp. 532-541, 2006.

[7] I.S. Dhillon, "Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning," *Proc. Seventh ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 269-274, 2001.

[8] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Bipartite Graph Partitioning and Data Clustering," *Proc. 10th Int'l Conf. Information and Knowledge Management,* pp. 25-32, 2001.

[9] Y. Cheng and G.M. Church, "Biclustering of Expression Data," *Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology,* pp. 93-103, 2000.

[10] Y. Kluger, R. Basri, J. Chang, and M. Gerstein, "Spectral Biclustering of Microarray Data: Coclustering Genes and Conditions," *Genome Research,* vol. 13, no. 4, pp. 703-716, 2003.

[11] S.C. Madeira and A.L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 1, no. 1, pp. 24-45, Jan.-Mar. 2004.

[12] T. George and S. Merugu, "A Scalable Collaborative Filtering Framework Based on Co-Clustering," *Proc. IEEE Fifth Int'l Conf. Data Mining,* pp. 625-628, 2005.

[13] P. Symeonidis, A. Nanopoulos, A.N. Papadopoulos, and Y. Manolopoulos, "Nearest-Biclusters Collaborative Filtering Based on Constant and Coherent Values," *Information Retrieval,* vol. 11, no. 1, pp. 51-75, 2008.

[14] N. Slonim and N. Tishby, "Document Clustering Using Word Clusters via the Information Bottleneck Method," *Proc. 23rd Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 208-215, 2000.

[15] R. El-Yaniv and O. Souroujon, "Iterative Double Clustering for Unsupervised and Semi-Supervised Learning," *Proc. 12th European Conf. Machine Learning (ECML '01),* pp. 121-132, 2001.

[16] I.S. Dhillon, S. Mallela, and D.S. Modha, "Information-Theoretic Co-Clustering," *Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 89-98, 2003.

[17] B. Long, Z.M. Zhang, and P.S. Yu, "Co-Clustering by Block Value Decomposition," *Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining,* pp. 635-640, 2005.

[18] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal Nonnegative Matrix T-Factorizations for Clustering," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 126-135, 2006.

[19] J. Han, *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers Inc., 2005.

[20] J.A. Hartigan, "Direct Clustering of a Data Matrix," *J. Am. Statistical Assoc.,* vol. 67, no. 337, pp. 123-129, 1972.

[21] A. Pothen, H.D. Simon, and K.-P. Liou, "Partitioning Sparse Matrices with Eigenvectors of Graphs," *SIAM J. Matrix Analysis and Applications,* vol. 11, no. 3, pp. 430-452, 1990.

[22] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.

[23] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, "Consistent Bipartite Graph Co-Partitioning for Star-Structured High-Order Heterogeneous Data Co-Clustering," *Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining,* pp. 41-50, 2005.

[24] N. Tishby, F.C. Pereira, and W. Bialek, "The Information Bottleneck Method," *Proc. 37th Ann. Allerton Conf. Comm., Control and Computing,* pp. 368-377, 1999.

[25] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D.S. Modha, "A Generalized Maximum Entropy Approach to Bregman Co-Clustering and Matrix Approximation," *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 509-514, 2004.

[26] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. L, and R. Harshman, "Indexing by Latent Semantic Analysis," *J. Am. Soc. for Information Science,* vol. 41, pp. 391-407, 1990.

[27] D.D. Lee and H.S. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization," *Nature,* vol. 401, no. 6755, pp. 788-791, 1999.

[28] Q. Gu and J. Zhou, "Co-Clustering on Manifolds," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 359-368, 2009.

[29] F. Pan, X. Zhang, and W. Wang, "CRD: Fast Co-Clustering on Large Datasets Utilizing Sampling-Based Matrix Decomposition," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* pp. 173-184, 2008.

[30] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf, "Learning with Local and Global Consistency," *Advances in Neural Information Processing Systems 16,* vol. 16, pp. 321-328, 2004.

[31] Y. Yang, D. Xu, F. Nie, J. Luo, and Y. Zhuang, "Ranking with Local Regression and Global Alignment for Cross Media Retrieval," *Proc. 17th Ann. ACM Int'l Conf. Multimedia,* pp. 175-184, 2009.

[32] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 5500, pp. 2323-2326, 2000.

[33] M. Wu and B. Schölkopf, "A Local Learning Approach for Clustering," *Advances in Neural Information Processing Systems 19,* vol. 19, pp. 1529-1536, 2007.

[34] F. Bach and Z. Harchaoui, "DIFFRAC: A Discriminative and Flexible Framework for Clustering," *Advances in Neural Information Processing Systems 20,* vol. 20, pp. 49-56, 2008.

[35] G. Strang, *Introduction to Linear Algebra,* third ed. Wellesley-Cambridge Press, 2003.

[36] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Advances in Neural Information Processing Systems 14,* vol. 14, pp. 585-591, 2002.

[37] G.H. Golub and C.F. Van Loan, *Matrix Computations,* third ed. Johns Hopkins Univ. Press, 1996.

[38] D. Cai, X. He, and J. Han, "Document Clustering Using Locality Preserving Indexing," *IEEE Trans. Knowledge and Data Eng.,* vol. 17, no. 12, pp. 1624-1637, Dec. 2005.

[39] L. Lováz and M.D. Plummer, *Matching Theory.* North-Holland, 1986.

[40] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science,* vol. 286, no. 5439, pp. 531-537, 1999.

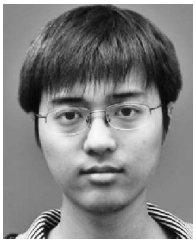
[41] J.-P. Brunet, P. Tamayo, T.R. Golub, and J.P. Mesirov, "Metagenes and Molecular Pattern Discovery Using Matrix Factorization," *Proc. Nat'l Academy of Sciences USA,* vol. 101, no. 12, pp. 4164-4169, 2004.

[42] S. Pomeroy et al., "Prediction of Central Nervous System Embryonal Tumour Outcome Based on Gene Expression," *Nature,* vol. 415, no. 6870, pp. 436-442, 2002.

Lijun Zhang received the BS degree in computer science from Zhejiang University, China, in 2007. Currently, he is working toward the PhD degree in computer science at Zhejiang University. His research interests include machine learning, information retrieval, and data mining. He is a student member of the IEEE.

Chun Chen received the BS degree in mathematics from Xiamen University, China, in 1981, and the MS and PhD degrees in computer science from Zhejiang University, China, in 1984 and 1990, respectively. Currently, he is working as a professor in the College of Computer Science, Zhejiang University. His research interests include information retrieval, data mining, computer vision, computer graphics, and embedded technology. He is a member of the IEEE.

Jiajun Bu received the BS and PhD degrees in computer science from Zhejiang University, China, in 1995 and 2000, respectively. Currently, he is a professor in the College of Computer Science, Zhejiang University. His research interests include embedded system, data mining, information retrieval, and mobile database. He is a member of the IEEE.



Zhengguang Chen received the BS degree in computer science from Zhejiang University, China, in 2009. Currently, he is working toward the MS degree in computer science at Zhejiang University. His research interests include computer vision, machine learning, and data mining.

Deng Cai received the BS and MS degrees from Tsinghua University both in automation, in 2000 and 2003, respectively, and the PhD degree in computer science from the University of Illinois at Urbana Champaign in 2009. Currently, he is working as an associate professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. His research interests include machine learning, data mining, and information retrieval. He is a member of the IEEE.

Jiawei Han is working as a professor of computer science at the University of Illinois. Currently, he is working as the director of Information Network Academic Research Center (INARC) supported by the Network Science-Collaborative Technology Alliance (NS-CTA) program of US Army Research Lab. He has chaired or served in more than 100 program committees of the major international conferences in the fields of data mining and database systems, and also served or is serving on the editorial boards for *Data Mining and Knowledge Discovery*, the *IEEE Transactions on Knowledge and Data Engineering*, the *Journal of Computer Science and Technology*, and the *Journal of Intelligent Information Systems*. He is the founding editor-in-chief of *ACM Transactions on Knowledge Discovery from Data (TKDD)*. His book *Data Mining: Concepts and Techniques* (Morgan Kaufmann) has been used worldwide as a textbook. He has received IBM Faculty Awards, HP Innovation Awards, the ACM SIGKDD Innovation Award (2004), the IEEE Computer Society Technical Achievement Award (2005), and the IEEE W. Wallace McDowell Award (2009). He is a fellow of the ACM and the IEEE.