Non-Negative Matrix Factorization for Semisupervised Heterogeneous Data Coclustering

Yanhua Chen, *Student Member*, *IEEE*, Lijun Wang, and Ming Dong, *Member*, *IEEE*

Abstract—Coclustering heterogeneous data has attracted extensive attention recently due to its high impact on various important applications, such us text mining, image retrieval, and bioinformatics. However, data coclustering without any prior knowledge or background information is still a challenging problem. In this paper, we propose a Semisupervised Non-negative Matrix Factorization (SS-NMF) framework for data coclustering. Specifically, our method computes new relational matrices by incorporating user provided constraints through simultaneous distance metric learning and modality selection. Using an iterative algorithm, we then perform trifactorizations of the new matrices to infer the clusters of different data types and their correspondence. Theoretically, we prove the convergence and correctness of SS-NMF coclustering and show the relationship between SS-NMF with other well-known coclustering models. Through extensive experiments conducted on publicly available text, gene expression, and image data sets, we demonstrate the superior performance of SS-NMF for heterogeneous data coclustering.

Index Terms—Non-negative matrix factorization, semisupervised clustering, heterogeneous data coclustering.

1 Introduction

Clustering or unsupervised learning is a generic name for a variety of procedures designed to find natural groupings or clusters in multidimensional data based on measured or perceived similarities among the patterns [18], [25]. The purpose of clustering is to extract useful information from unlabeled data. Applications of data clustering are found in many fields, such as text mining, Web analysis, image grouping, and bioinformatics. In general, clustering algorithms can be categorized into two groups: partitioning (flat) clustering and hierarchical clustering. Partitioning methods typically divide the data into a given number of clusters directly. Some of the popular methods in data partitioning include k-means [18] and probabilistic clustering using the Naive Bayes or Gaussian mixture model [2], [33]. Hierarchical clustering aims to obtain a hierarchy of clusters by building a tree structure that shows how the clusters are related to each other. The clustering result can be obtained by cutting the tree at a desired level [39].

Recently, spectral clustering has been widely applied in various domains [12], in which data objects are modeled as vertices of a weighted graph with edge weights representing the similarity between two data objects. Clustering is then obtained by solving an eigenvalue problem and cutting the graph vertices into different partitions. More recently, matrix-factorization-based clustering has emerged as an effective approach for clustering problems in high-dimensional spaces. In [41], it is shown that Non-negative Matrix Factorization (NMF) outperforms spectral methods in document clustering, achieving higher accuracy and efficiency.

With the fast growth of Internet and computational technologies in the past decade, many data mining applications have advanced swiftly from the simple clustering of one data type to the coclustering of multiple data types, usually involving high heterogeneity. For example, the interrelations of words, documents, and categories in text corpus, Web pages, search queries, and Web users in a Web search system, papers, keywords, authors, and conferences in a scientific publication domain can be identified through simultaneous clustering of several related data types. This is not achievable by traditional clustering methods. First, heterogeneous data contain different types of relations. Processing and interpreting them in a unified way presents a major challenge. Ad hoc integration or normalization (e.g., concatenating different features into a vector of fixed length) rarely works. Second, various data types are related to each other. Tackling each type independently will lose these interactions, which are essential to gain a full understanding of the data. Consequently, coclustering is introduced in the data mining literature, for both two data types (pairwise coclustering), [3], [13], [14], [17], [24], and multiple (more than two) data types (high-order coclustering) [4], [19], [20], [34], [35]. Through coclustering, we are able to discover a hidden global structure in the heterogeneous data, which seamlessly integrates multiple data types to provide us a better picture of the underlying data distribution, highly valuable in many real world applications.

Existing coclustering methods are mostly derived based on the graph model, which requires solving eigen-problem. Computationally, they are inefficient and inapplicable to large-scale data sets. Moreover, they are completely unsupervised. Accurately coclustering heterogeneous data without domain-dependent background information is still a challenging task. In this paper, we propose a Semisupervised NMF (SS-NMF) based framework to incorporate prior

• *The authors are with the Department of Computer Science, Wayne State University, Detroit, MI 48202.*
E-mail: {chenyanh, ljwang, mdong}@wayne.edu.

Manuscript received 21 Apr. 2008; revised 11 Oct. 2008; accepted 8 July 2009; published online 17 July 2009.
Recommended for acceptance by Y. Chen.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-04-0214.
Digital Object Identifier no. 10.1109/TKDE.2009.169.

knowledge into heterogeneous data coclustering. In the proposed SS-NMF coclustering methodology, users are able to provide constraints on data samples in the central type, specifying whether they "must" (*must-link*) or "cannot" (*cannot-link*) be clustered together. Our goal is to improve the quality of coclustering by learning a new distance metric based on these constraints. Using an iterative algorithm, we then perform trifactorizations of the new data matrices, obtained with the learned distance metric, to infer the central data clusters while simultaneously deriving the clusters of related feature modalities. The preliminary version of this work was first presented in a shortened form as conference abstracts [8], [10]. The major contribution of this work is summarized as follows:

1. We propose a novel algorithm for heterogeneous data coclustering based on NMF. Computationally, NMF coclustering is more efficient and flexible than graph-based models and can provide more intuitive clustering results.
2. To the best of our knowledge, this is the first work on the semisupervised coclustering of multiple data types. Through distance metric learning and modality selection, prior knowledge is integrated into coclustering, making *must-link* data points as tight as possible and *cannot-link* data points as loose as possible.
3. From a theoretical perspective, our approach is mathematically rigorous. The convergence and correctness are proved. In addition, we show that some well-established approaches such as probability-based coclustering, information-theoretical coclustering, and spectral coclustering can be considered as variations of our method under certain conditions.

The rest of the paper is organized as follows: We review related work in Section 2. The SS-NMF coclustering algorithm is derived in Section 3. The proof on the correctness and convergence of the proposed algorithm is presented in Section 4, in which we also build the relationship between SS-NMF with other data coclustering models. Experimental results appear in Section 5. Finally, we conclude in Section 6.

2 Related Work

In this section, we provide a review of related work. We first introduce representative coclustering algorithms in the literature. Then, we briefly overview semisupervised learning techniques.

In general, coclustering approaches can be divided into three categories: probability-based models, information-theory-based models, and graph theoretic approaches. In the first category, Hoffman and Puzicha [24] proposed the Probabilistic Latent Semantic Analysis (PLSA) model for cooccurrence data and used it for collaborative filtering. In PLSA, the data objects are embedded into a low-dimensional space using Singular Value Decomposition (SVD) for efficient pairwise coclustering. Later, PLSA was further developed into a more comprehensive generative model, Latent Dirichelt Allocation (LDA), to cluster rows and columns of data simultaneously. Within the framework of LDA, many pairwise coclustering approaches, such us Infinite Relational Model [28], Mixed Membership Blockmodel [1], and Bayesian coclustering [37], were introduced recently using different inference engines. Also recently, Long et al. proposed a high-order coclustering framework, Mixed Membership Relational Clustering (MMRC) model [35], in which parametric soft clustering results are derived using Expectation Maximization (EM) for a large number of exponential family distributions. MMRC can identify multiple cluster structures for each type of data and interactive patterns between different types of data.

Concerning the information-theory-based models, Dhillon et al. [14] presented a pairwise coclustering algorithm to maximize the mutual information between the clustered random variables subject to the constraints on the number of row and column clusters. A more general framework was presented in [3] wherein any Bregman divergence can be used as the objective function for coclustering. Later, Gao et al. [19] extended pairwise information theoretic models to high-order data coclustering. More recently, Bekkerman and Jeon [4] proposed the Combinatorial Markov Random Field (CMRF) algorithm for high-order coclustering, in which each data modality is modeled as a single combinatorial random variable in Markov Random Field. However, theoretical proof of the effectiveness and correctness of information-theory-based models is typically not presented.

Graph theoretical approaches have a well-defined objective function for data coclustering and, thus, are widely used. Spectral learning, such as Bipartite Spectral Graph Partitioning (BSGP) [13], was proposed and applied to cocluster documents and words. BSGP formulates the data matrix as a bipartite graph and seeks to find the optimal normalized cut for the graph. With a similar philosophy, Gao et al. proposed Consistent Bipartite Graph Copartitioning (CBGC) using semidefinite programming for high-order data coclustering and applied it to hierarchical text taxonomy preparation [20]. Due to the nature of graph partitioning theory, these algorithms have the restriction that clusters from different types of objects must have one-to-one associations. More recently, Long et al. [34] proposed Spectral Relational Clustering (SRC), in which they formulated heterogeneous coclustering as collective factorization on related matrices and derived a spectral algorithm to cluster multitype interrelated data objects simultaneously. SRC provides more flexibility by lifting the requirement of one-to-one association in graph-based coclustering. However, to obtain data clusters, all the aforementioned graph theoretical approaches require solving an eigen-problem, which computationally is not efficient for large-scale data sets.

In many practical learning domains, there is a large supply of unlabeled data but limited labeled data, and in most cases it can be expensive to generate large amounts of labeled data. Consequently, semisupervised learning, i.e., learning from a combination of both labeled and unlabeled data, has become a topic of significant recent interest. The framework of semisupervised learning is applicable to both classification and clustering.

In semisupervised classification, some unlabeled data are frequently exploited in addition to the category-labeled training data to improve the classification accuracy. Popular approaches include cotraining [6], Transductive Support Vector Machines (TSVM) [27], and using EM to incorporate unlabeled data into training [21]. On the other hand, semisupervised clustering uses class labels or pairwise constraints on examples to aid unsupervised clustering. It can group data using the categories of the initial labeled data as well as the unlabeled data in order to modify the existing set of categories and reflect the whole regularities

in the data. Two sources of information are usually available to a semisupervised clustering method: the similarity distance measurement in unsupervised clustering and the class labels or pairwise constraints (*must-link* or *cannot-link*) provided by users. For semisupervised clustering to be profitable, these two sources of information should not completely contradict each other. Existing methods for semisupervised clustering based on source information generally fall into two categories: *constraint-based* and *distance-based* methods. In constraint-based approaches, the clustering algorithm itself is modified so that the available labels or constraints are used to bias the search for an appropriate clustering of the data [23]. In distance-based approaches, an existing clustering algorithm that uses a distance measure is employed; however, the distance measure is first trained to satisfy the labels or constraints in the supervised data [40]. Recent research in semisupervised clustering tends to combine the constraint-based with distance-based approaches.

Noticeable efforts on semisupervised clustering include Semisupervised Kernel K-means (SS-KK) [29], Semisupervised Spectral Normalize Cuts (SS-SNC) [26], and SS-NMF [9]. SS-KK transforms the clustering distance measure by weighted kernel k-means with reward and penalty constraints to perform semisupervised clustering of data given either as vectors or as a graph. SS-SNC utilizes supervision to change the clustering distance measure with pairwise information by spectral methods. In [11], it is shown that SS-NMF provides a unified framework for semisupervised clustering. Existing algorithms, such us SS-KK and SS-SNC, can be considered as special cases of SS-NMF. In addition, experiments show that SS-NMF is able to generate significantly better clustering results by quickly learning from a few constraints.

Even though the research on data coclustering and semisupervised clustering have attracted substantial attention in the past years, to date, most semisupervised clustering models are only applicable to homogeneous data, in which *must-link* and *cannot-link* constraints are directly incorporated into the similarity matrix of homogeneous clustering. On the other hand, integrating domain knowledge into coclustering is still a largely unsolved problem due to the existence of multiple data types. Recently, Bekkerman and Sahami proposed a semisupervised CMRF model (SS-CMRF) for pairwise coclustering [5] under the information theoretic framework. However, without proof of correctness and convergence, their approach is not mathematically rigorous. In the following, we will derive a theoretically sound algorithm based on SS-NMF and apply it to heterogeneous data coclustering.

3 SS-NMF for Data Coclustering

In this section, we propose an SS-NMF model for heterogeneous data coclustering. Specifically, we will discuss 1) how to incorporate prior knowledge into data coclustering through distance metric learning and modality selection, and 2) how to efficiently infer clusters of different data types simultaneously using NMF.

3.1 Model Formulation

NMF is initially proposed for "parts-of-whole" decomposition [31] and later extended to a general framework of data clustering [17]. It can model widely varying data distributions and do both hard and soft clustering. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in R^{d \times n}$ be the data matrix with nonnegative elements. NMF factorizes $\mathbf{X}$ into two nonnegative matrices,

$$\mathbf{X} \approx \mathbf{F}\mathbf{G}^T, \tag{1}$$

where $\mathbf{F} \in R^{d \times k}$ is the cluster centroid, $\mathbf{G} \in R^{n \times k}$ is the cluster membership indicator which corresponds to the degree object $\mathbf{x_i}$ is associated with cluster k, and k the number of clusters. The factorization is typically obtained by the least square minimization. A simple example of NMF clustering is illustrated as follows:

$$\begin{aligned}
\mathbf{X} &= \begin{bmatrix} 0.185 & 0.326 & 0.761 & 2.799 & 2.375 & 2.970 & 2.585 \\ 0.508 & 0.380 & 0.884 & 2.134 & 2.374 & 2.342 & 2.524 \\ 0.452 & 0.887 & 0.457 & 2.065 & 2.484 & 2.253 & 2.163 \\ 1.486 & 1.843 & 1.858 & 0.566 & 0.103 & 0.417 & 0.269 \\ 1.496 & 1.806 & 1.610 & 0.612 & 0.158 & 0.560 & 0.784 \end{bmatrix} \\
&\approx \mathbf{F}\mathbf{G}^T = \begin{bmatrix} 1.7621 & 0.2165 \\ 1.5164 & 0.3013 \\ 1.4388 & 0.3101 \\ 0.0000 & 1.0424 \\ 0.1327 & 0.9891 \end{bmatrix} \times \\
&\quad \begin{bmatrix} 0.0000 & 0.0000 & 0.0522 & 0.4740 & 0.5074 & 0.5203 & 0.4944 \\ 0.4924 & 0.6104 & 0.5686 & 0.1599 & 0.0213 & 0.1244 & 0.1419 \end{bmatrix}.
\end{aligned} \tag{2}$$

In (2), based on the membership indicator $\mathbf{G}$, clearly the first three columns form one cluster, and the last four columns give another.

In our model, given a Star-structured Heterogeneous Relational Data (SHRD) set, with a central data type $\mathcal{X}_c$, and l feature modalities $\mathcal{X}_1, \ldots, \mathcal{X}_p, \ldots, \mathcal{X}_l$, the goal is to cluster central data type $\mathcal{X}_c$ into k_c disjoint clusters simultaneously with feature modality $\mathcal{X}_1$ into k_1 disjoint clusters, $\ldots, \mathcal{X}_p$ into k_p disjoint clusters, $\ldots$, and $\mathcal{X}_l$ into k_l disjoint clusters. Notice that SHRD provides a very good abstraction for many real-world data mining problems. For example, it can be used to model words, documents, and categories in text mining, where the document is the central data type; authors, conferences, papers, and keywords in academic publications, where the paper is the central data type; and images, color, and texture features in image retrieval, where the image is the central data type. As such, coclustering SHRD can provide a global data structure, which shows correlations of various feature modalities, leading to a better understanding of the underlying process that generates the data. For instance, through image and low-level feature coclustering, images can be grouped together with different kinds of features. By linking certain feature modalities to a cluster of images, we can perform more efficient and effective content-based image retrieval.

To derive a solution of the coclustering problem under matrix factorization framework, we first model SHRD using a set of relation matrices. That is, a matrix $\mathbf{R}^{(cp)} \in R^{n_c \times n_p}$ is used to represent the relation between a central data type $\mathcal{X}_c$ and a feature modality $\mathcal{X}_p (1 \leq p \leq l)$. See Fig. 1a for an example of SHRD, in which the relations between the central data type and four feature modalities are modeled by relational matrices $\mathbf{R}^{(c1)}$, $\mathbf{R}^{(c2)}$, $\mathbf{R}^{(c3)}$, and $\mathbf{R}^{(c4)}$, respectively. Then, we can formulate the task of

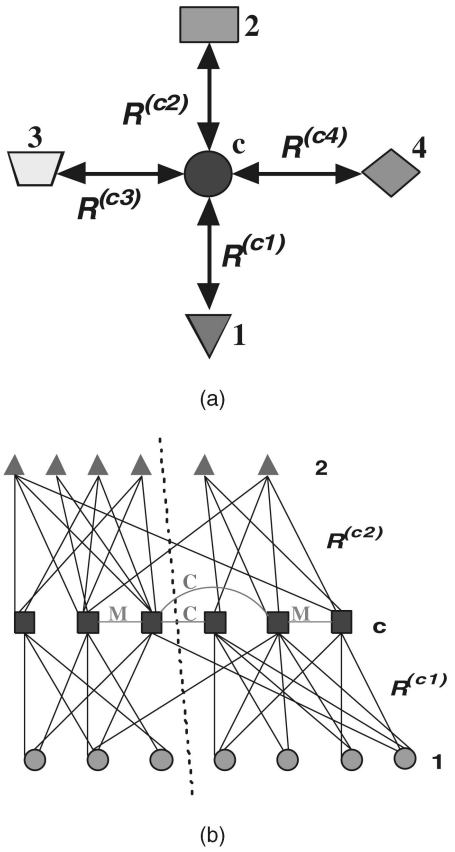


Fig. 1. (a) Heterogeneous star-structured relational data. (b) Star-structured triplet coclustering with must-link (M) and cannot-link (C) constraints.

coclustering as an optimization problem with non-negative trifactorization of $\mathbf{R}^{(cp)}$,

$$J = \min_{\mathbf{G}^{(c)}\geq\mathbf{0},\mathbf{G}^{(p)}\geq\mathbf{0},\mathbf{S}^{(cp)}\geq\mathbf{0}} \sum_{p=1}^{l} \left\|\mathbf{R}^{(cp)} - \mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\right\|^2, \quad (3)$$

where $\mathbf{G}^{(c)} \in R^{n_c\times k_c}$ and $\mathbf{G}^{(p)} \in R^{k_p\times n_p}$ are the cluster indicator matrices, and $\mathbf{S}^{(cp)} \in R^{k_c\times k_p}$ is the cluster association matrix which provides the relation between the central data type and each feature modality.

In semisupervised coclustering, we assume that the supervision is provided as two sets of pairwise constraints derived from the given labels on the central data type: *must-link* constraints $M = \{(\mathbf{x}_i, \mathbf{x}_j)\}$ and *cannot-link* constraints $C = \{(\mathbf{x_i}, \mathbf{x_j})\}$, where $(\mathbf{x}_i, \mathbf{x}_j) \in M$ implies that $\mathbf{x}_i$ and $\mathbf{x}_j$ are labeled as belonging to the same cluster, while $(\mathbf{x}_i, \mathbf{x}_j) \in C$ implies that $\mathbf{x}_i$ and $\mathbf{x}_j$ are labeled as belonging to different clusters. Note that our assumption is made based on the fact that in practice constraints are much easier to specify on the central data type (e.g., documents in document-word coclustering) than on the feature modalities (e.g., words). Fig. 1b shows a data triplet, the basic element of SHRD, with constraints on the central data type. The green edges indicate the *must-link* constraints M, while the red edges denote the *cannot-link* constraints C. The dotted line shows the optimal coclustering result.

3.2 SS-NMF for Heterogeneous Data Coclustering

In this section, we present an SS-NMF-based data coclustering algorithm. Specifically, we first discuss how constraints can be integrated into NMF-based pairwise coclustering through distance metric learning. Then, we generalize it to high-order coclustering and give the complete algorithm.

Let $\mathbf{R}^{(c1)} \in R^{n_c\times n_1}$ denote the relational matrix. The objective of pairwise coclustering is to cluster the n_c data points in the central type c along with the n_1 features in feature modality 1 while keeping the constraint violations to a minimum. In order to accomplish semisupervised coclustering, it is necessary to discover a new distance metric over the features based on the constraints provided by the users on the central data type. Specifically, given two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ of $\mathbf{R}^{(c1)}$, the Mahalanobis distance between them can be defined as

$$d\big(\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)}\big) = \sqrt{\big(\mathbf{x}_i^{(c1)} - \mathbf{x}_j^{(c1)}\big)^T \mathbf{L}^{(c1)} \big(\mathbf{x}_i^{(c1)} - \mathbf{x}_j^{(c1)}\big)}.$$

Thus, learning the distance metric $\mathbf{L}^{(c1)}$ is equivalent to finding a linear projective mapping $\sqrt{\mathbf{L}^{(c1)}}$ in the feature space [40] such that data points $(\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)}) \in M$ are moved closer to each other while $(\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)}) \in C$ are pushed further away. That is, we solve the following optimization problem:

$$\max \quad g\big(\mathbf{L}^{(c1)}\big) = \frac{\sum_{(\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)})\in C} \big\|\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)}\big\|_{\mathbf{L}^{(c1)}}}{\sum_{(\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)})\in M} \big\|\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)}\big\|_{\mathbf{L}^{(c1)}}}, \quad (4)$$

where $\|.\|$ is the Frobenius matrix norm. This maximization problem is equivalent to the generalized Semisupervised Linear Discriminate Analysis (SS-LDA) problem as follows:

$$J = \min \frac{trace\big(\mathbf{L}^{(c1)}\mathbf{W}_M^{(c1)}\big)}{trace\big(\mathbf{L}^{(c1)}\mathbf{B}_C^{(c1)}\big)}, \quad (5)$$

where $\mathbf{W}_M$ is the within-distance matrix from *must-link* constraints, $\mathbf{B}_C$ is the between-distance matrix from *cannot-link* constraints. The solution of (5) can be obtained accordingly [40].

Through learning, the distance metric $\mathbf{L}^{(c1)}$ implicitly embeds the *must-link* and *cannot-link* constraints. Thus, the original data $\mathbf{R}^{(c1)}$ is projected into a new space

$$\widetilde{\mathbf{R}}^{(c1)} = \sqrt{\mathbf{L}^{(c1)}}\mathbf{R}^{(c1)}.$$

We then perform non-negative trifactorization of the new matrix $\widetilde{\mathbf{R}}^{(c1)}$

$$J = \min_{\mathbf{G}^{(c)}\geq 0,\mathbf{G}^{(1)}\geq 0,\mathbf{S}^{(c1)}\geq 0} \left\|\widetilde{\mathbf{R}}^{(c1)} - \mathbf{G}^{(c)}\mathbf{S}^{(c1)}\mathbf{G}^{(1)}\right\|^2. \quad (6)$$

The minimization of (6) can be done by updating one factor while fixing others [17].

An example of SS-NMF for pairwise coclustering is illustrated in Fig. 2. Fig. 2a shows the relational data $\mathbf{R}^{(c1)} \in R^{30\times 2}$ with two clusters (15 asterisk points and 15 circle points), both following Gaussian distributions. The first step of SS-NMF coclustering, distance metric learning, is shown in Fig. 2c, in which a new relational data $\widetilde{\mathbf{R}}^{(c1)}$ is learned through embedding the distance metric $\mathbf{L}^{(c1)}$ into the original $\mathbf{R}^{(c1)}$. Clearly, with the *must-link* and *cannot-link* constraints, the data points within the same cluster are placed closer while points in different clusters are moved away. The result of the

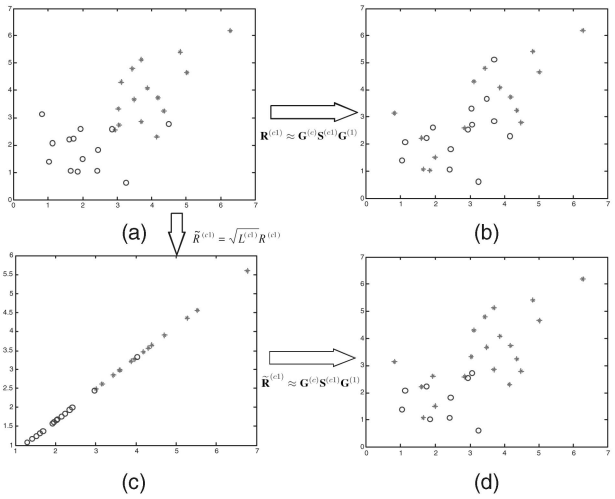


Fig. 2. An illustration of SS-NMF for data coclustering: (a) Relational data $\mathbf{R}^{(c1)}$ with two clusters. (b) Clustering result of $\mathbf{R}^{(c1)}$ with unsupervised NMF. (c) New relational data $\widetilde{\mathbf{R}}^{(c1)}$ after a linear projection with distance metric $\mathbf{L}^{(c1)}$. (d) Clustering result of $\widetilde{\mathbf{R}}^{(c1)}$ with SS-NMF.

second step, trifactorization of $\widetilde{\mathbf{R}}^{(c1)}$, is illustrated in Fig. 2d. As a comparison, we also show the result obtained by the unsupervised NMF coclustering in Fig. 2b. It is clear that the semisupervised model has a better performance.

In general SHRD coclustering, the central data type has to be clustered together with all feature modalities. Again, let $\mathbf{R}^{(cp)}(1 \le p \le l)$ denote a relational matrix between a central data and each feature modality, the goal of SS-NMF coclustering is to iteratively cluster the rows and columns of each $\mathbf{R}^{(cp)}$, subject to the M and C constraints. Similar to the case of pairwise coclustering, the first step in high-order coclustering is to obtain the new matrix $\widetilde{\mathbf{R}}^{(cp)}$. In other words, we need to learn a distance metric $\mathbf{L}^{(cp)}$ for each relation based on the constraints such that the clustering result on the central type is globally optimized. Moreover, high-order coclustering introduces an additional layer of complexity: because feature modalities can play different roles in the grouping of the central data type, we have to consider the issue of modality selection. To this end, we introduce a modality importance factor, $\mathbf{a} = [\alpha^{(cp)}]$, to denote the relative weighting of each modality. Specifically, $\mathbf{a}$ is computed by solving an unconstrained linear regression problem. The solution of this problem has a close form and is easy to obtain. However, such an unconstrained least square solution may not provide satisfactory results if considering prediction accuracy and interpretation. Thus, we further apply the coefficient shrinkage technique [7] to limit $\alpha^{(cp)}$ in the range of [0, 1]. Note that the modality selection and distance metric learning are strongly dependent. This suggests that these two objectives must be achieved simultaneously. In Algorithm 1, we propose an algorithm to iteratively learn the optimal distance metric $\mathbf{L}^{(cp)}$ and modality importance factor $\mathbf{a}$. Based on these two variables, we compute a new relational data matrix $\widetilde{\mathbf{R}}^{(cp)}$. Thus, $\widetilde{\mathbf{R}}^{(cp)}$ incorporates information captured by $\mathbf{a}$ and $\mathbf{L}^{(cp)}$.

Algorithm 1. Simultaneous Distance Metric Learning and Modality Selection

INPUT: Original relational matrix $\mathbf{R}^{(cp)}(1 \le p \le l)$, central type $\mathcal{X}_c$ with must-link constraint M, and cannot-link constraint C

OUTPUT: Optimal distance metric $\mathbf{L}^{(cp)}$, modality importance factor $\mathbf{a}$, and new relational matrix $\widetilde{\mathbf{R}}^{(cp)}$

METHOD:

1. Construct the target distance vector $\widetilde{D}$ based on constraints M and C, where each element $\widetilde{d}_{ij}$ is 0 if $(\mathbf{x}_i, \mathbf{x}_j) \in M$, and 1 if $(\mathbf{x}_i, \mathbf{x}_j) \in C$
2. Obtain the initial distance metric $\mathbf{L}^{(cp)}$ by SS-LDA with constraints M and C
3. Set the number of iterations t=0
   a. Compute the new relational matrix
   $$\widetilde{\mathbf{R}}^{(cp)} = \sqrt{\mathbf{L}^{(cp)}}\mathbf{R}^{(cp)}$$
   b. Compute the distance vector $D^{(cp)}$, which contains only data points with constraints
   c. Obtain the modality importance factor through the following optimization
   $$\mathbf{a}_t^{opt} = arg\ \min_\alpha \|\widetilde{D} - \textstyle\sum_{p=1}^{l} \alpha^{(cp)} D^{(cp)}\|^2$$
   d. Let $\mathbf{R}^{(cp)} = \alpha^{(cp)}\widetilde{\mathbf{R}}^{(cp)}$, and learn the new distance metric $\mathbf{L}^{(cp)}$ by SS-LDA with constrains M and C
4. If $\mathbf{a}_{t+1} - \mathbf{a}_t > \varepsilon$, set $t = t + 1$ and repeat steps a-d; otherwise, stop, let $\widetilde{\mathbf{R}}^{(cp)} = \mathbf{R}^{(cp)}$, and output the optimal distance metric $\mathbf{L}^{(cp)}$, the modality importance factor $\mathbf{a}$, and the new relational matrix $\widetilde{\mathbf{R}}^{(cp)}$

To achieve high-order coclustering, we again need to perform non-negative trifactorization of $\widetilde{\mathbf{R}}^{(cp)}$ shown in (3). In order to obtain the (local) optimal solution for the above minimization problem, the cluster structure for each data type has to be updated iteratively. In Algorithm 2, we derive an EM style approach that iteratively performs the matrix decomposition using a set of multiplicative updating rules.

Algorithm 2. SS-NMF for High-Order Coclustering

INPUT: New relational matrix $\widetilde{\mathbf{R}}^{(cp)}$

OUTPUT: Cluster indicator matrices $\mathbf{G}^{(c)}$, $\mathbf{G}^{(p)}$, and cluster association matrix $\mathbf{S}^{(cp)}$

METHOD:

1. Initialize $\mathbf{G}^{(c)}$, $\mathbf{G}^{(p)}$, and $\mathbf{S}^{(cp)}$ with non-negative values
2. Iterate for each $i(1 \le i \le n_p)$, $h(1 \le h \le k_p)$, and $p(1 \le p \le l)$ until *convergence*
   a. Cluster indicator matrices:
   $$\mathbf{G}_{ih}^{(c)} \leftarrow \mathbf{G}_{ih}^{(c)} \frac{\sum_{p=1}^{l} \left(\widetilde{\mathbf{R}}^{(cp)}\mathbf{G}^{(p)}\mathbf{S}^{(cp)^T}\right)_{ih}}{\sum_{p=1}^{l} \left(\mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)^T}\mathbf{G}^{(p)}\mathbf{S}^{(cp)^T}\right)_{ih}}, \quad (7)$$
   $$\mathbf{G}_{ih}^{(p)} \leftarrow \mathbf{G}_{ih}^{(p)} \frac{\left(\mathbf{S}^{(cp)^T}\mathbf{G}^{(c)^T}\widetilde{\mathbf{R}}^{(cp)}\right)_{ih}}{\left(\mathbf{S}^{(cp)^T}\mathbf{G}^{(c)^T}\mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\right)_{ih}}. \quad (8)$$
   b. Cluster association matrix:
   $$\mathbf{S}_{ih}^{(cp)} \leftarrow \mathbf{S}_{ih}^{(cp)} \frac{\left(\mathbf{G}^{(c)^T}\widetilde{\mathbf{R}}^{(cp)}\mathbf{G}^{(p)^T}\right)_{ih}}{\left(\mathbf{G}^{(c)^T}\mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\mathbf{G}^{(p)^T}\right)_{ih}}. \quad (9)$$

4 Theoretical Analysis

4.1 Algorithm Convergence and Correctness

We now prove the theoretical convergence and correctness of the SS-NMF coclustering algorithm. Motivated by [17],

[32], we render the proof based on optimization theory, auxiliary function, and several matrix inequalities.

4.1.1 Correctness

First, we prove the correctness of the algorithm, which can be stated as,

Proposition 1. *If the solution converges based on the updating rules in (7)-(9), the solution satisfies the KKT optimality condition.*

Proof. Following the standard theory of constrained optimization, we introduce the Lagrangian multipliers λ_0, λ_p, and λ_{p+l} to minimize the lagrangian function,

$$\begin{aligned} &L\big(\mathbf{G}^{(c)}, \mathbf{G}^{(p)}, \mathbf{S}^{(cp)}, \lambda_0, \lambda_p, \ldots, \lambda_{p+l}\big) \\ &\quad = \sum_{p=1}^{l} \big\|\widetilde{\mathbf{R}}^{(cp)} - \mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\big\|^2 - \mathrm{Tr}\big(\lambda_0 \mathbf{G}^{(c)^T}\big) \\ &\qquad - \mathrm{Tr}\sum_{p=1}^{l}\big(\lambda_p \mathbf{S}^{(cp)^T}\big) - \mathrm{Tr}\sum_{p=1}^{l}\big(\lambda_{p+l}\mathbf{G}^{(p)^T}\big). \end{aligned} \tag{10}$$

Based on the KKT complementarity conditions $\frac{\partial L}{\partial \mathbf{G}^{(c)}}=0$, $\frac{\partial L}{\partial \mathbf{S}^{(cp)}}=0$, and $\frac{\partial L}{\partial \mathbf{G}^{(p)}}=0$, we obtain the following three equations:

$$\begin{aligned} &\sum_{p=1}^{l}\big(2\widetilde{\mathbf{R}}^{(cp)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T} - 2\mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T}\big) \\ &\quad + \lambda_0 = 0, \\ &2\mathbf{G}^{(c)^T}\widetilde{\mathbf{R}}^{(cp)}\mathbf{G}^{(p)^T} - 2\mathbf{G}^{(c)^T}\mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\mathbf{G}^{(p)^T} + \lambda_p = 0, \\ &2\mathbf{S}^{(cp)^T}\mathbf{G}^{(c)^T}\widetilde{\mathbf{R}}^{(cp)} - 2\mathbf{S}^{(cp)^T}\mathbf{G}^{(c)^T}\mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)} + \lambda_{p+l} = 0. \end{aligned}$$

We apply the Hadamard multiplication on both sides of the three equations by $\mathbf{G}^{(c)}$, $\mathbf{S}^{(cp)}$, and $\mathbf{G}^{(p)}$, respectively. Using KKT conditions of

$$\lambda_0 \odot \mathbf{G}^{(c)} = 0, \qquad \lambda_p \odot \mathbf{S}^{(cp)} = 0, \qquad \lambda_{p+l} \odot \mathbf{G}^{(p)} = 0,$$

where $\odot$ denotes the Hadamard product of two matrices. We can prove that if $\mathbf{G}^{(c)}$, $\mathbf{S}^{(cp)}$, and $\mathbf{G}^{(p)}$ are a local minimizer of the objective function in (10), the following three equations are satisfied:

$$\begin{aligned} &\left(\sum_{p=1}^{l}\big(\widetilde{\mathbf{R}}^{(cp)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T}\big) - \sum_{p=1}^{l}\big(\mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T}\big)\right) \\ &\quad \odot \mathbf{G}^{(c)} = 0, \\ &\big(\big(\mathbf{G}^{(c)^T}\widetilde{\mathbf{R}}^{(cp)}\mathbf{G}^{(p)^T}\big) - \big(\mathbf{G}^{(c)^T}\mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\mathbf{G}^{(p)^T}\big)\big) \odot \mathbf{S}^{(cp)} = 0, \\ &\big(\big(\mathbf{S}^{(cp)^T}\mathbf{G}^{(c)^T}\widetilde{\mathbf{R}}^{(cp)}\big) - \big(\mathbf{S}^{(cp)^T}\mathbf{G}^{(c)^T}\mathbf{R}^{(cp)}\big)\big) \odot \mathbf{G}^{(p)} = 0. \end{aligned}$$

Based on the above three equations, we derive the proposed updating rules of (7)-(9). If the updating rules converge, the solution satisfies the KKT optimality condition. The proof is completed.

□

4.1.2 Convergence

Next, we prove the convergence of the algorithm. In Proposition 2, we show that the objective function decreases monotonically under the three updating rules of (7)-(9). This can be done by making use of an auxiliary function similar to that used in [32].

Proposition 2. *If any two of three matrices $\mathbf{G}^{(c)}$, $\mathbf{S}^{(cp)}$, and $\mathbf{G}^{(p)}$ are fixed, $J = \sum_{p=1}^{l}\|\widetilde{\mathbf{R}}^{(cp)} - \mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(c)}\|^2$ decreases monotonically under the updating rules of (7)-(9).*

Proof. Assume $\mathbf{S}^{(cp)}$ and $\mathbf{G}^{(p)}$ are fixed matrices, a function $F\big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}\big)$ is called an auxiliary function of

$$J\big(\mathbf{G}^{(c)^{[t+1]}}\big) \text{ if it satisfies } \quad F\big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}\big) \geq J\big(\mathbf{G}^{(c)^{[t+1]}}\big)$$
$$\text{and} \quad F\big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t+1]}}\big) = J\big(\mathbf{G}^{(c)^{[t+1]}}\big)$$

for any $\mathbf{G}^{(c)^{[t+1]}}$ and $\mathbf{G}^{(c)^{[t]}}$. Define

$$\mathbf{G}^{(c)^{[t+1]}} = arg\min F\big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}\big),$$

then we can construct

$$\begin{aligned} J\big(\mathbf{G}^{(c)^{[t]}}\big) &= F\big(\mathbf{G}^{(c)^{[t]}}, \mathbf{G}^{(c)^{[t]}}\big) \geq F\big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}\big) \\ &\geq J\big(\mathbf{G}^{(c)^{[t+1]}}\big). \end{aligned}$$

Thus, $J(\mathbf{G}^{(c)^{[t]}})$ is monotonic decreasing (nonincreasing).

The key step is to find an appropriate auxiliary function

$$F\Big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}\Big).$$

Since $\mathbf{G}^{(p)}$ and $\mathbf{S}^{(cp)}$ are fixed, we write

$$\begin{aligned} J\Big(\mathbf{G}^{(c)^{[t+1]}}\Big) = \sum_{p=1}^{l}\mathrm{Tr}\big(&\mathbf{R}^{(cp)^T}\mathbf{R}^{(cp)} - 2\mathbf{R}^{(cp)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T}\mathbf{G}^{(c)^T} \\ &+ \mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T}\mathbf{G}^{(c)^T}\big), \end{aligned}$$

and show that

$$\begin{aligned} F\Big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}\Big) = \sum_{p=1}^{l}\Bigg\{ &\| \mathbf{R}^{(cp)} \|^2 \\ &- \sum_{ih} 2\big(\mathbf{R}^{(cp)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T}\big)_{ih}\mathbf{G}_{ih}^{(c)^{[t]}}\left(1 + 2log\frac{\mathbf{G}_{ih}^{(c)^{[t+1]}}}{\mathbf{G}_{ih}^{(c)^{[t]}}}\right) \\ &+ \sum_{ih}\frac{\big(\mathbf{G}^{(c)^{[t]}}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T}\big)_{ih}\mathbf{G}_{ih}^{(c)^{4[t+1]}}}{\mathbf{G}_{ih}^{(c)^{3[t]}}}\Bigg\} \end{aligned} \tag{11}$$

is an auxiliary function of $J(\mathbf{G}^{(c)^{[t+1]}})$.

First, we show that the inequality

$$F\Big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}\Big) \geq J\Big(\mathbf{G}^{(c)^{[t+1]}}\Big)$$

holds. We can see the second term in $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}})$ (aside from the negative sign) is always smaller than the second term in $J(\mathbf{G}^{(c)^{[t+1]}})$ because of the inequality

$$\frac{\mathbf{G}_{ih}^{(c)^{[t+1]}}}{\mathbf{G}_{ih}^{(c)^{[t]}}} \geq 1 + 2log\left(\frac{\mathbf{G}_{ih}^{(c)^{[t+1]}}}{\mathbf{G}_{ih}^{(c)^{[t]}}}\right), \forall \frac{\mathbf{G}_{ih}^{(c)^{[t+1]}}}{\mathbf{G}_{ih}^{(c)^{[t]}}} > 0.$$

In addition, the third term in $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}})$ is always bigger than the third term in $J(\mathbf{G}^{(c)^{[t+1]}})$ [17]. Thus, the condition

$$F\Big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}\Big) \geq J\Big(\mathbf{G}^{(c)^{[t+1]}}\Big)$$

holds. Second, we show the equality

$$F\Big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t+1]}}\Big) = J\Big(\mathbf{G}^{(c)^{[t+1]}}\Big)$$

holds. It is obvious when $\mathbf{G}^{(c)^{[t]}} = \mathbf{G}^{(c)^{[t+1]}}$, the equality

$$F\Big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t+1]}}\Big) = J\Big(\mathbf{G}^{(c)^{[t+1]}}\Big) \quad \text{holds.}$$

Therefore, $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}})$ is an auxiliary function of

$$J\Big(\mathbf{G}^{(c)^{[t+1]}}\Big).$$

Since we have

$$\mathbf{G}^{(c)^{[t+1]}} = arg\ \min_{\mathbf{G}^{(c)}} F\Big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}\Big), \mathbf{G}^{(c)^{[t+1]}}$$

is given by the minimum of $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}})$ while fixing $\mathbf{G}^{(c)^{[t]}}$. The minimum value is obtained by setting

$$\begin{aligned}
&\frac{\partial F\big(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}\big)}{\partial \mathbf{G}_{ih}^{(c)^{[t+1]}}} \\
&= \sum_{p=1}^{l} \Bigg\{ - \sum_{ih} 4\Big(\mathbf{R}^{(cp)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T}\Big)_{ih} \frac{\mathbf{G}_{ih}^{(c)^{[t+1]}}}{\mathbf{G}_{ih}^{(c)^{[t+1]}}} \\
&\quad + 4\sum_{ih} \frac{\Big(\mathbf{G}^{(c)^{[t]}}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T}\Big)_{ih}\mathbf{G}_{ih}^{(c)^{3[t+1]}}}{G_{ih}^{(c)^{3[t]}}} \Bigg\} = 0.
\end{aligned}$$

Thus, we can derive the updating rule of (7) as

$$\mathbf{G}_{ih}^{(c)} \leftarrow \mathbf{G}_{ih}^{(c)} \frac{\sum_{p=1}^{l}\big(\mathbf{R}^{(cp)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T}\big)_{ih}}{\sum_{p=1}^{l}\big(\mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)}\big)_{ih}}.$$

Under this updating rule, $J(\mathbf{G}^{(c)^{[t]}})$ decreases monotonically.

Alternatively, we can assume that $\mathbf{S}^{(cp)}$ and $\mathbf{G}^{(c)}$, or $\mathbf{G}^{(c)}$ and $\mathbf{G}^{(p)}$, are fixed matrices. In both cases, we can render a similar proof for the updating rules of (8) and (9). The proof is completed. □

4.2 Relationship with Other Data Coclustering Models

We now discuss the relationship between NMF-based coclustering and other well-known coclustering algorithms (e.g., probability based, information-theory-based, and graph-theory-based coclustering). We show that existing methods can be considered as variations of our model under certain conditions.

4.2.1 Probability-Based Coclustering

In real world data sets, objects may belong to multiple clusters with varying degrees. Consequently, probability-based coclustering models have emerged as a flexible modeling tool for complex relational data, where each row and column have a mixed (soft) membership. MMRC, a unified framework for probability-based coclustering, is proposed recently in [35]. Assuming that $\mathbf{R}^{(12)}$ is the relational matrix, with rows and columns representing two variables $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, the objective of MMRC for pairwise coclustering is to maximize the likelihood as,

$$\begin{aligned}
J_{MMRC} &= \max \prod_{i=1}^{n_1}\prod_{j=1}^{n_2} \mathbf{R}_{ij}^{(12)} log\ p(x_{1i}, x_{2j}) \\
&= \min \prod_{i=1}^{n_1}\prod_{j=1}^{n_2} \mathbf{R}_{ij}^{(12)} log \frac{\mathbf{R}_{ij}^{(12)}}{p(x_{1i}, x_{2j})},
\end{aligned} \tag{12}$$

where the joint occurrence probability is factorized as $\mathbf{R}_{ij}^{(12)} = p(x_{1i}, x_{2j}) = p(x_{1i}|z_k)p(z_k)p(x_{2j}|z_k)$, and z_k is a set of cluster indicators.

On the other hand, NMF-based pairwise coclustering using the KL-divergence (NMF-KL) as the cost function is to minimize,

$$\begin{aligned}
J_{NMF-KL} = \min &\prod_{i=1}^{n_1}\prod_{j=1}^{n_2} \mathbf{R}_{ij}^{(12)} \\
&\Bigg[log \frac{\mathbf{R}_{ij}^{(12)}}{p(x_{1i}, x_{2j})} - \mathbf{R}_{ij}^{(12)} + \big(\mathbf{G}^{(1)}\mathbf{S}^{(12)}\mathbf{G}^{(2)}\big)_{ij}\Bigg].
\end{aligned} \tag{13}$$

It can be shown that (12) is identical to (13), i.e., $J_{MMRC} = -J_{NMF-KL} + constant$, by setting $(\mathbf{G}^{(1)}\mathbf{S}^{(12)}\mathbf{G}^{(2)})_{ij} = p(x_{1i}, x_{2j})$ [16]. Thus, we have $\mathbf{G}_{ik}^{(1)} = p(x_{1i}|z_k)$, $\mathbf{G}_{jk}^{(2)} = p(x_{2j}|z_k)$, and $\mathbf{S}_{kk}^{(12)} = p(z_k)$. In other words, the coclustering solution is similar even though different inference engines are used by the two methods. The relationship between high-order coclustering using NMF-KL and MMRC can be derived similarly.

4.2.2 Information-Theory-Based Coclustering

The representative algorithms for information-theory-based coclustering include Information-Theoretic for pairwise Coclustering (ITCC) [14] and high-order coclustering [19], CMRFs for pairwise coclustering [5] and high-order coclustering [4].

ITCC was proposed in [14] to maximize the mutual information between the clustered random variables subject to the constraints on the number of row and column clusters. Let X_1 and X_2 be discrete random variables that take values in the sets $\{x_{11}, \ldots, x_{1n_1}\}$ and $\{x_{21}, \ldots, x_{2n_1}\}$, respectively, and $\hat{X}_1$ and $\hat{X}_2$ be the cluster (partition) random variables that take values in the sets $\{\hat{x}_{11}, \ldots, \hat{x}_{1n_1}\}$ and $\{\hat{x}_{21}, \ldots, \hat{x}_{2n_2}\}$, respectively. The objective of ITCC is to minimize the mutual information loss $I(X_1; X_2) - I(\hat{X}_1; \hat{X}_2)$. CMRF is to maximize the Most Probable Explanation $I(\hat{X}_1; \hat{X}_2)$ based on the basic principles in MRF graph inferences. It is clear to see that CMRF is a simplified version of ITCC, assuming that $I(X_1; X_2)$ is a constant.

In our NMF model, the joint distribution of X_1 and X_2 can be formulated as $\mathbf{R}^{(12)}$ by assigning the probability $p(x_{1n_1}, x_{2n_2})$ as the weight on the edge between the node n_1 of the central data type $\mathcal{X}_1$, and the node n_2 of the feature modality $\mathcal{X}_2$. After the trifactorization, $\mathbf{R}^{(12)}$ is decomposed into three parts: $\mathbf{S}^{(12)}$, $\mathbf{G}^{(1)}$, and $\mathbf{G}^{(2)}$. The association matrix $\mathbf{S}^{(12)}$ can be considered as the joint probability $p(\hat{x}_{s_1}, \hat{x}_{s_2})$ of hidden variables s_1 and s_2, while the indicator matrix $\mathbf{G}^{(1)}$ or $\mathbf{G}^{(2)}$ can be considered as the conditional probability of the hidden variables in $\mathbf{S}^{(12)}$: $p(x_{n_1} \mid \hat{x}_{s_2})$ or $p(x_{n_2} \mid \hat{x}_{s_2})$. Based on this formulation, we can see that the objective function of pairwise NMF is a variation of ITCC (CMRF).

If the multiinformation $I(\hat{X}_1;\ldots;\hat{X}_l)$ is introduced into ITCC (CMRF) as the combinations of several pairwise relations, it can be extended to coclustering involving more than two random variables. The similarity between high-order NMF and high-order ITCC (CMRF) can be derived accordingly.

4.2.3 Graph-Theory-Based Coclustering

Some of the well-known graph-theory-based coclustering algorithms include BSGP [13] for pairwise coclustering and SRC [34] for high-order coclustering.

BSGP was proposed for pairwise data coclustering in [13]. BSGP formulates the data as a bipartite graph; its adjacency matrix can be written as

$$\begin{bmatrix} 0 & \mathbf{R}^{(c1)} \\ \mathbf{R}^{(c1)^T} & 0 \end{bmatrix},$$

where $\mathbf{R}^{(c1)} \in R^{n_c \times n_1}$ is a relational matrix. It was shown that spectral partitioning on the bipartite graph can be converted to a partial SVD problem. That is

$$\min_{\mathbf{G}^{(c)^T}\mathbf{G}^{(c)}=\mathbf{I},\mathbf{G}^{(\mathbf{1})^\mathbf{T}}\mathbf{G}^{(1)}=\mathbf{I},\mathbf{S}^{(\mathbf{c1})}\text{is diag}} \left\|\mathbf{R}^{(c1)} - \mathbf{G}^{(c)}\mathbf{S}^{(c1)}\mathbf{G}^{(1)}\right\|^2.$$

On the other hand, NMF-based pairwise coclustering is to minimize the following objective function:

$$\min_{\mathbf{G}^{(c)}\geq 0,\mathbf{G}^{(1)}\geq 0,\mathbf{S}^{(c1)}\geq 0} \left\|\mathbf{R}^{(c1)} - \mathbf{G}^{(c)}\mathbf{S}^{(c1)}\mathbf{G}^{(1)}\right\|^2.$$

The advantage of NMF over BSGP has been discussed in [34].

SRC is proposed in [34] for high-order data coclustering. It iteratively embeds each type of data into low-dimensional spaces and benefits through the interactions in the hidden structure of different data types. The underlying objective function is

$$\min_{\mathbf{G}^{(c)^T}\mathbf{G}^{(c)}=\mathbf{I},\mathbf{G}^{(p)^T}\mathbf{G}^{(p)}=\mathbf{I}} \sum_{p=1}^{l} \left\|\mathbf{R}^{(cp)} - \mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\right\|^2.$$

On the other hand, NMF-based high-order coclustering is to minimize the following function:

$$\min_{\mathbf{G}^{(c)}\geq 0,\mathbf{G}^{(p)}\geq 0,\mathbf{S}^{(cp)}\geq 0} \sum_{p=1}^{l} \left\|\mathbf{R}^{(cp)} - \mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\right\|^2.$$

The advantage of NMF or SS-NMF over SRC can best be illustrated using an example. We construct a synthetic data set which has 30 data points in the central type $\mathcal{X}_c$ with two feature modalities $\mathcal{X}_1$ (300 features) and $\mathcal{X}_2$ (2 features). Each data type has two clusters of equal size. That is, we build two relational matrices: $\mathbf{R}^{(c1)}$ of size 30×300 and $\mathbf{R}^{(c2)}$ of size 30×2, both binary matrices with two-by-two block structures generated by the Bernoulli distribution. Specifically, $\mathbf{R}^{(c1)}$ is generated based on the block structure

$$\begin{bmatrix} 0.8 & 0.1 \\ 0.2 & 0.9 \end{bmatrix},$$

and $\mathbf{R}^{(c2)}$ is based on the block structure

$$\begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix}.$$

Unlike SRC, NMF or SS-NMF maps the data into a nonnegative latent semantic space which is not required to be orthogonal. Panels (a)-(c), (d)-(f), and (g)-(i) in Fig. 3 show the clustering results obtained by SRC, NMF, and SS-NMF, in which the two clusters are denoted by the red stars and the blue triangles, respectively. For NMF or SS-NMF, we plot the data points in the subspace of the first two column vectors of $\mathbf{G}^{(c)}$, $\mathbf{G}^{(1)}$, and $\mathbf{G}^{(2)}$, while for SRC we use the subspace of the first two singular vectors. Note that for either NMF or SS-NMF, each data point takes a non-negative value on both axes. In the NMF subspace, each axis corresponds to a cluster, and all the data points belonging to the same cluster are nicely located close to the axis. In the SS-NMF subspace, the data points belonging to the same cluster almost spread along each axis. This indicates that SS-NMF can provide better clustering accuracy than unsupervised NMF because the cluster label for a data point is determined by finding the axis with which the data point has the largest projection value. On the other hand, in the SRC subspace, we observe no direct relationship between the axes (singular vectors) and the clusters.

5 Experiments and Results

In this section, we empirically demonstrate the performance of SS-NMF coclustering. We first conduct pairwise coclustering on documents (i.e., documents and words) and gene expressions (i.e., conditions and genes). In these experiments, we compare the performance of SS-NMF coclustering with six representative clustering algorithms, including KK, BSGP, CMRF, NMF, SS-KK, and SS-CMRF. In addition, we also compare our model with a well-known semisupervised classification method, TSVM. Then, we perform high-order coclustering for text corpus (i.e., words, documents, and categories, in which the document is the central data type) and image data (i.e., color and texture features associated with images). Similarly, on these data sets, we compare SS-NMF coclustering with four algorithms, i.e., SRC, CMRF, NMF, and SS-CMRF. Through these comparisons, we demonstrate the relative position of our method with respect to existing approaches on (semisupervised) data clustering/classification and show the benefits of integrating prior knowledge into coclustering.

5.1 Data Description and Preprocessing

5.1.1 Text Coclustering

We primarily utilize the data sets used in [22].[1] Data sets $oh5$ and $oh15$ are from OHSUMED collection, a subset of MEDLINE database, which contains 233,445 documents indexed using 14,321 unique categories. Data set WAP is from the WebACE Project, and each document corresponds to a web page listed in the subject hierarchy of Yahoo. Data set $re0$ is the *Reuters-21578* text categorization collection (distribution 1.0). We also use the Newsgroup data which contains about 2,000 articles from 20 newsgroups [30].[2] In our experiments, we intermix some of the data sets mentioned above. Tables 1 and 2 give the details of the data sets we use for pairwise (e.g., document-word)

1. http://www.cs.umn.edu/~han/data/tmdata.tar.gz.
2. http://www.cs.uiuc.edu/homes/dengcai2/Data/TextData.html.

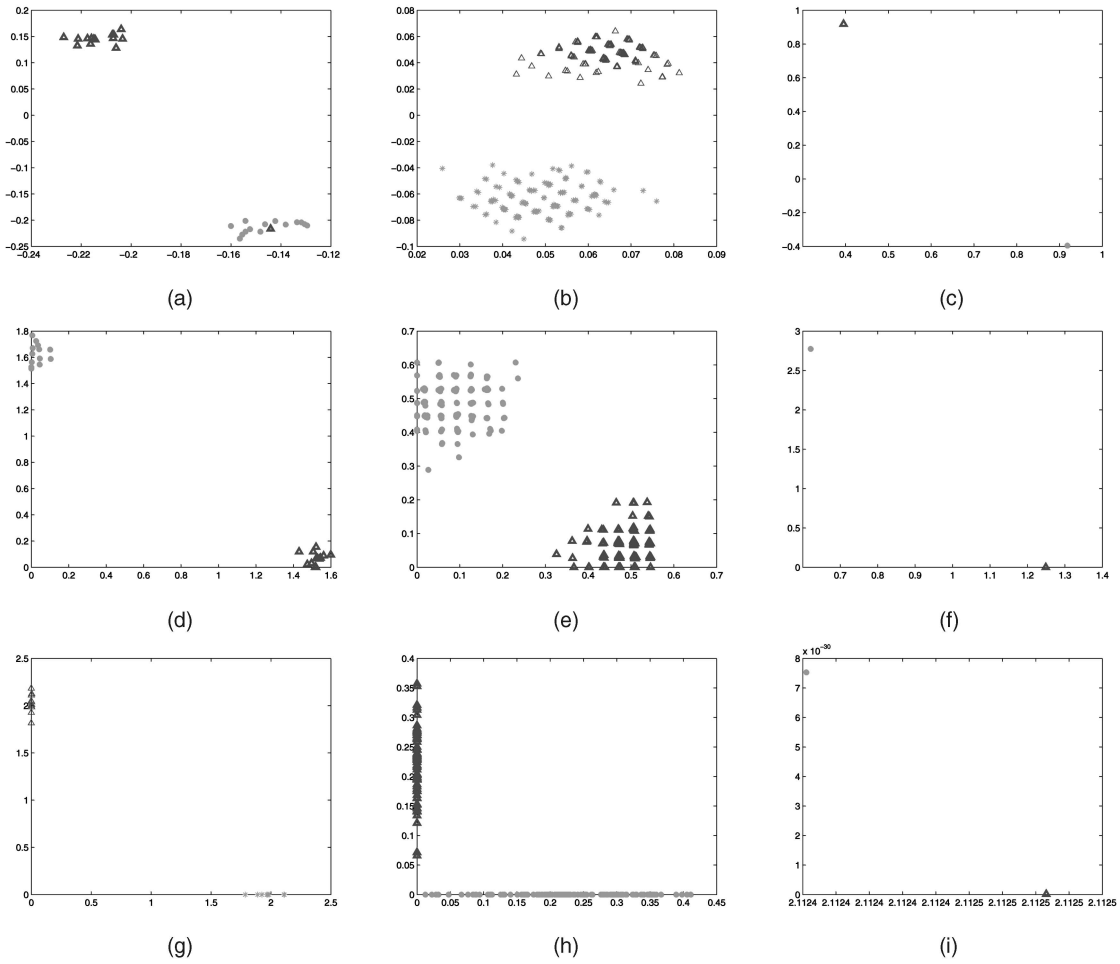


Fig. 3. (a)-(c) Clustering results by SRC in the subspace of the first two singular vectors of $\mathbf{G}^{(c)}$, $\mathbf{G}^{(1)}$, and $\mathbf{G}^{(2)}$. There is no direct relationship between the axes and the clusters. (d)-(f) Clustering results by NMF in the subspace of the first two column vectors of $\mathbf{G}^{(c)}$, $\mathbf{G}^{(1)}$, and $\mathbf{G}^{(2)}$. The data points from the two clusters are distributed closely to the two axes. (g)-(i) Clustering results by SS-NMF (with 5 percent constraints) in the subspace of the first two column vectors of $\mathbf{G}^{(c)}$, $\mathbf{G}^{(1)}$, and $\mathbf{G}^{(2)}$. The data points from the two clusters are distributed exactly along the two axes.

and high-order (e.g., word-document-category) coclustering, respectively.

We use the term frequency to build a *document-word* matrix. To compare the algorithms on the same ground and make our results consistent with others [17], [34], we carry out feature selection to choose the top 1,000 words by descending values of the mutual information between a word w and a document label y:

$$I(W,Y) = \sum_{Y}\sum_{W} p(w,y) log\left(\frac{p(w,y)}{p_1(w)p_2(y)}\right),$$

where W and Y are random variables, denoting word and document labels, respectively. The *Document-category* matrix is constructed by computing the probability of each document belonging to each category. The following technique is used: 1) For each class of documents, select the top 1,000 words based on mutual information. 2) For each document, if any of the top 1,000 word occurs, the amount of occurrence is 1, otherwise 0. 3) The probability of one document belonging to a category is the ratio of the sum of occurrence of the top 1,000 words in this document to 1,000. Thus, every element of *document-category* matrix is in the range $[0, 1]$. In addition, for semisupervised clustering, we

TABLE 1
Data Sets for Text Pairwise (Document-Word) Coclustering

Name	*Data sets*	*Data structure*	*No. of clusters*	*No. of documents*
CT1	*oh15*	*Adenosine-Diphosphate, Blood-Vessels*	*2*	*154*
CT2	*oh15*	*Aluminum,Blood-Coagulation-Factors*	*2*	*122*
CT3	*re0*	*interest,reserves*	*2*	*261*
CT4	*re0*	*housing,jobs*	*2*	*55*
CT5	*re0*	*housing,interest,jobs*	*3*	*274*
CT6	*oh15*	*Aluminum,Blood-Vessels,Leucine*	*3*	*207*
CT7	*re0*	*cpi, housing, ipi, lei,retail*	*5*	*144*
CT8	*re0*	*bop,cpi,gnp,housing,interest,ipi,jobs,lei,money,reserves*	*10*	*1150*

TABLE 2
Data Sets for Text High-Order (Word-Document-Category) Coclustering

Name	Data sets	Data structure	No. of categories	No. of clusters	No. of documents
HT1	oh15,re0	{Adenosine-Diphosphate,Aluminum,Cell-Movement}, {cpi,money}	2	5	899
HT2	oh15,re0	{Blood-Coagulation-Factors,Enzyme-Activation,Staphylococcal-Infections}, {jobs,reserves}	2	5	461
HT3	oh15,re0	{Aluminum,Blood-Coagulation-Factors,Blood-Vessels} {housing,retail}	2	5	256
HT4	oh5,re0	{Aluminum,Cell-Movement,Staphylococcal-Infections}, {cpi,wpi}	2	5	391
HT5	WAP,re0	{media,film,music}, {cpi,jobs}	2	5	404
HT6	Newsgroup	{rec.sport.baseball,rec.sport.hockey}, {talk.politics.guns,talk.politics.mideast,talk.politics.misc}	2	5	500
HT7	Newsgroup	{comp.graphics,comp.os.ms-windows.misc}, {rec.autos,rec.motorcycles}, {sci.crypt,sci.electronics}	3	6	300
HT8	Newsgroup	{comp.graphics,comp.os.ms-windows.misc}, {sci.electronics, sci.med},	2	4	3932
HT9	Newsgroup	{rec.autos,rec.motorcycles, rec.sport.baseball}, {sci.crypt,sci.electronics, sci.space},	2	6	5942

TABLE 3
Data Sets for Gene Expression Pairwise (Condition-Gene) Coclustering

Name	Data sets	Data structure	No. of clusters	No. of conditions
BT1	ALL/AML	ALL,AML	2	72
BT2	BreastCancer	Relapse, Non-relapse	2	97
BT3	CentralNervous	Class1, Class2	2	60
BT4	ColonTumor	Positive,Negative	2	62
BT5	LungCancer	MPM,ADCA	2	181
BT6	OvarianCancer	Cancer,Normal	2	253
BT7	ALL/MLL/AML	ALL,MLL,AML	3	72

define the percentage (percent) of pairwise constraints with respect to all the possible document pairs, which is $\binom{\text{total docs}}{2}$. The document constraints are generated by randomly selecting documents from each class of the data set. Other data sets also use similar defined constraints for the central data type.

5.1.2 Gene Expression Coclustering

We utilize seven data sets from Kent Ridge Biomedical Data Repository[3] for gene expression coclustering, including *ALL/AML Leukemia*, *Breast Cancer*, *Central Nervous System*, *Colon Tumor*, *Lung Cancer*, *Ovarian Cancer*, and *ALL/MLL/AML Leukemia*. In our experiment, we compute the first principal component u_1 based on Principal Component Analysis. Since u_1 is a linear combination of genes, the magnitude of $u_1(i)$ is indicative of the variance of gene i [15]. We sort all genes in a descending order based on the variances and retain only the top 2,000 genes. The details of these data sets are given in Table 3.

5.1.3 Image Coclustering

The image data used in our experiments is chosen from Corel CDs, which contains 31,438 general-purpose images of various contents, such as plants, animals, buildings, human society, etc. To evaluate our algorithm, we construct a data set with 1,000 images from 10 categories: "eggs," "decoys," "firearms," "cards," "buses," "abstract," "foliage," "dawn," "texture," and "wave." Some examples from each category are shown in Fig. 4. In our experiment, we mix up some of the aforementioned categories with details in Table 4.

For image coclustering, a large number of visual contents are extracted from each image [36], [38], belonging to two modalities: color and texture. Specifically, color features include color channels (RGB, 9 features, including mean, variance, and skewness of R, G, and B channels), color histogram (CH, 12 features), and color coherence vector (CCV, 24 features). Texture features include Gabor wavelet-based texture (Gab, 24 features), edge direction histogram (EDH, 9 features), and edge direction coherence vector (EDCV, 9 features). Based on the extracted visual features, we build two relational matrices *image-color* and *image-texture*, and each element in the matrices is normalized into

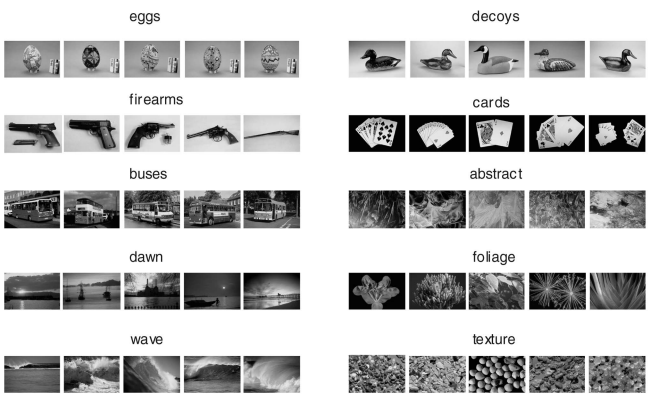


Fig. 4. Image samples for high-order coclustering.

3. http://datam.i2r.a-star.edu.sg/datasets/krbd/.

TABLE 4
Data Sets for Image High-Order (Color-Image-Texture) Coclustering

Name	Data structure	No. of modalities	No. of clusters	No. of images
IT1	eggs,decoys	3	2	200
IT2	dawn,foliage	3	2	200
IT3	decoys,dawn	3	2	200
IT4	decoys,firearms,cards,buses	3	4	400
IT5	abstract,dawn,foliage,waves	3	4	400
IT6	eggs,decoys,dawn,foliage	3	4	400
IT7	eggs,decoys,buses,abstract,texture,dawn	3	6	600

the range $[0, 1]$. Coclustering is then performed on images, color features (45 dimensions), and texture features (42 dimensions) simultaneously.

5.2 Evaluation Method

We evaluate the clustering results using the accuracy rate AC, which measures how accurately a learning method assigns label $\hat{y}_i$ to a data point with the ground truth y_i. The AC metric is defined as

$$AC = \frac{\sum_{i=1}^{n} \delta(y_i, \hat{y}_i)}{n}, \tag{14}$$

where n denotes the total number of data points or features in the experiment and δ is the delta function that equals one if $\hat{y}_i = y_i$; otherwise, it is zero. Since an iterative algorithm is not guaranteed to find the global minimum, it is beneficial to run the algorithm several times with different initial values and choose the average of all the test runs as the final accuracy value. In our experiments, for each given cluster number k, we conduct 10 test runs, and the final AC value is the average of all runs.

5.3 Pairwise Coclustering

5.3.1 Text Pairwise Coclustering

First, we conduct pairwise coclustering experiments on the text data sets with *document-word* matrices and compare the performance of SS-NMF with the following six clustering methods:

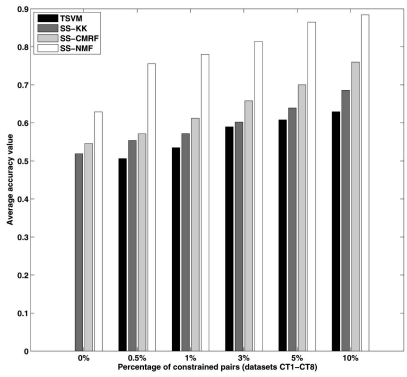
1. KK [29],
2. BSGP [13],
3. CMRF [5],
4. NMF (i.e., SS-NMF with 0 percent constraints),
5. SS-KK [29], and
6. SS-CMRF [5].

The first four are popular unsupervised methods, whereas SS-KK and SS-CMRF are representative semisupervised ones. Moreover, we also compare with a well-known semisupervised classification method: TSVM [27].

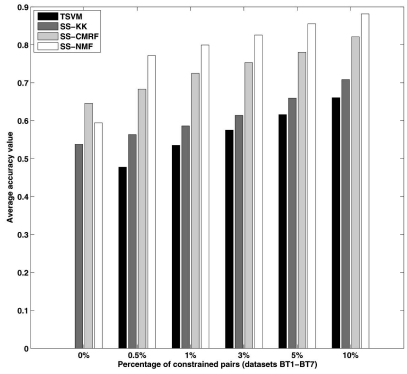
The top half of Table 5 shows the AC values of document clustering obtained by unsupervised methods: KK, BSGP, CMRF, and NMF, the semisupervised classification method: TSVM, and three semisupervised clustering methods: SS-KK, SS-CMRF, and SS-NMF. All of semisupervised methods are reported based on incorporating 10 percent constraints into the central data. Averaged AC values over all eight data sets are also computed. In the four unsupervised approaches, KK has the lowest average AC. This is mainly due to the fact that the document-word relation is not formulated and utilized in one-way KK clustering. AC values of BSGP or CMRF, on average, are about 10 percent lower than NMF, which is the best among the unsupervised methods. However, all unsupervised methods get a low AC value (around 30 percent) for the data set CT8, which has a large number of clusters ($k = 10$). That is, no meaningful clustering results are produced. Table 5 also shows that semisupervised clustering methods provide at least a 15 percent increase on the average AC values when compared with the corresponding unsupervised ones. This indicates that a semisupervised clustering method can generally benefit from additional

TABLE 5
Comparison of Accuracy among Unsupervised Clustering KK, BSGP, CMRF, NMF, Semisupervised Classification TSVM, and Semisupervised Clustering SS-KK, SS-CMRF, SS-NMF with 10 Percent Constraints on Text (Document-Word) Data Sets (CT1-CT8) and Gene Expression (Condition-Gene) Data Sets (BT1-BT7)

Name	KK	BSGP	CMRF	NMF	TSVM	SS-KK	SS-CMRF	SS-NMF
CT1	0.7897	0.4870	0.5545	0.8052	0.6270	0.9610	0.7984	0.8606
CT2	0.5164	0.6148	0.6582	0.6475	0.6542	0.7541	0.9041	0.9902
CT3	0.6820	0.7510	0.7264	0.7586	0.8243	0.7588	0.8682	0.8774
CT4	0.5355	0.7190	0.5419	0.4635	0.7163	0.7455	0.8310	0.8248
CT5	0.4652	0.6148	0.4974	0.6364	0.8502	0.6606	0.7682	0.9818
CT6	0.4638	0.5072	0.5585	0.6763	0.4783	0.6618	0.7585	0.9101
CT7	0.4236	0.2778	0.5000	0.6667	0.4665	0.5000	0.7261	0.9944
CT8	0.2857	0.2330	0.3327	0.3774	0.4268	0.4478	0.4667	0.6343
Average	0.5191	0.5256	0.5462	0.6290	0.6293	0.6862	0.7600	0.8842
BT1	0.6050	0.8194	0.8238	0.6111	0.6513	0.8606	0.9538	0.9444
BT2	0.6189	0.5155	0.6156	0.5258	0.6583	0.7320	0.7426	0.7732
BT3	0.5000	0.6000	0.5250	0.5833	0.6491	0.6233	0.7147	0.7667
BT4	0.5000	0.7258	0.6452	0.6613	0.6291	0.7613	0.8400	0.8710
BT5	0.6570	0.5138	0.9118	0.8785	0.8467	0.8569	1.0000	1.0000
BT6	0.5099	0.6522	0.5167	0.4704	0.6650	0.6403	0.7393	0.9960
BT7	0.3750	0.5417	0.4829	0.4306	0.5266	0.4861	0.6778	0.8194
Average	0.5380	0.6241	0.6459	0.5944	0.6609	0.7086	0.8212	0.8815



(a)



(b)

Fig. 5. Comparison of average accuracy for semisupervised classification TSVM, and pairwise coclustering SS-KK, SS-CMRF, and SS-NMF, with different amounts of constraints on (a) text data and (b) gene expression data.

constraints, and thus greatly improve the clustering results. Moreover, SS-NMF outperforms SS-KK and SS-CMRF, especially in the data sets with more than two clusters, i.e., data sets CT5-CT8. It is also worth noting that the AC values of SS-NMF are as high as 99 percent on the data sets CT2, CT5, and CT7. In other words, SS-NMF provides near perfect clustering results on these data sets. Another important observation is that all the semisupervised clustering approaches outperform TSVM on average AC due to very limited background knowledge (up to 10 percent). In these cases, the known labels are simply too few to initiate a good classifier training. Overall, the superior performance of SS-NMF is evident in terms of the average accuracy.

In Fig. 5a, we plot the average AC value on all eight data sets against the increasing percentage of pairwise constraints for TSVM, SS-KK, SS-CMRF, and SS-NMF. We clearly see that SS-NMF significantly outperforms TSVM, SS-KK, and SS-CMRF in all cases, gaining at least 12 percent higher clustering accuracy. Another important observation is that the average accuracy of all four methods consistently increases with the gradual increase of the pairwise constraints (from 0.5 to 10 percent). Particularly, SS-NMF is able to generate significantly better results (over 10 percent) by quickly learning from just a few constraints (0.5 percent). Therefore, document clustering performance can be greatly improved even with very limited prior knowledge.

5.3.2 Gene Expression Pairwise Coclustering

Second, we conduct coclustering on gene expressions with *condition-gene* matrix and compare the performance of SS-NMF with the same set of algorithms used in Section 5.3.1.

The bottom half of Table 5 shows the AC values of condition clustering obtained by both unsupervised methods and semisupervised ones with 10 percent constraints. Overall, it is evident that SS-NMF provides the best clustering result on average when compared with other unsupervised or semisupervised methods. As the results demonstrate, the clustering accuracy gain of SS-NMF over unsupervised methods is over 20 percent on most data sets even though unsupervised NMF is not the best among unsupervised approaches. This clearly indicates the outstanding benefits brought by the partial supervision integrated in SS-NMF. It is also worth pointing out that the AC values of SS-NMF are (nearly) 100 percent on the data sets BT5 and BT6.

Fig. 5b illustrates the average AC values against the increasing percentage of pairwise constraints for semisupervised condition clustering/classification. Overall, SS-NMF provides the highest accuracy among the four semisupervised methods. Not surprisingly, we see that more constraints on the patient conditions lead to higher accuracy for all four approaches. Again, substantial performance improvement is achieved by SS-NMF, up to 20 percent accuracy increase, with very limited prior knowledge (e.g., 0.5 percent constraints).

5.4 High-Order Coclustering

5.4.1 Text High-Order Coclustering

First, we conduct experiments to cocluster words, documents, and categories and compare the performance of SS-NMF with three unsupervised approaches and one semisupervised method, namely,

1. SRC [34],
2. CMRF [4],
3. NMF (i.e., SS-NMF with 0 percent constraints), and
4. SS-CMRF (the high-order SS-CMRF is directly extended from the prior work in [4] and [5]).

Coclustering accuracy. The top half of Table 6 shows document coclustering accuracy obtained by SRC, CMRF, NMF, SS-CMRF, and SS-NMF (both with 15 percent constraints). Averaged AC values over all nine data sets are also reported. In our experiment, we observe that the relations among multiple data types in some text data sets are highly complicated (e.g., HT8 and HT9). To achieve reasonable clustering results, more domain knowledge is required. Thus, up to 15 percent constraints are used in high-order coclustering experiments (recall that we use up to 10 percent constraints in pairwise coclustering).

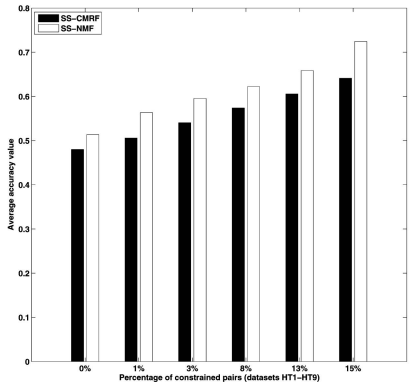
From Table 6, it is obvious that NMF outperforms other unsupervised methods in six out of nine text data sets. In general, SRC performs the worst among the three unsupervised ones. Specifically, its accuracy on the data set HT7 with three categories and six document clusters is only 19 percent. Also from Table 6, semisupervised methods provide significantly better results than the corresponding unsupervised ones. The average AC of SS-CMRF increases

TABLE 6
Comparison of Clustering Accuracy between Unsupervised SRC, CMRF, NMF, and Semisupervised SS-CMRF, SS-NMF with 15 Percent Constraints on Text High-Order (Word-Document-Category) Coclustering (Data Sets HT1-HT9) and Image High-Order (Color-Image-Texture) Coclustering (Data Sets IT1-IT7)

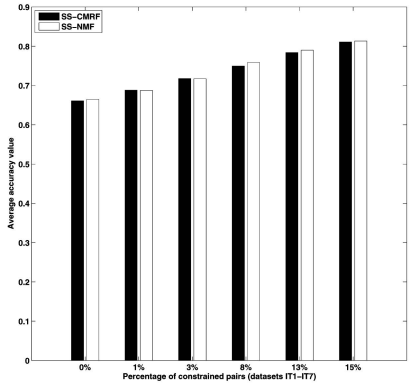
Name	*SRC*	*CMRF*	*NMF*	*SS-CMRF*	*SS-NMF*
HT1	*0.4772*	*0.5362*	*0.5250*	*0.7072*	*0.8509*
HT2	*0.4989*	*0.5785*	*0.6529*	*0.7344*	*0.8243*
HT3	*0.3359*	*0.3820*	*0.5391*	*0.5779*	*0.6875*
HT4	*0.4450*	*0.5992*	*0.5601*	*0.7481*	*0.8261*
HT5	*0.6411*	*0.6171*	*0.6386*	*0.7266*	*0.8267*
HT6	*0.4989*	*0.6014*	*0.5780*	*0.6877*	*0.8620*
HT7	*0.1900*	*0.3593*	*0.4333*	*0.5288*	*0.6467*
HT8	*0.2538*	*0.3226*	*0.3533*	*0.4863*	*0.5244*
HT9	*0.2243*	*0.3238*	*0.3389*	*0.4600*	*0.4697*
Average	*0.3961*	*0.4800*	*0.5132*	*0.6410*	*0.7243*
IT1	*0.7500*	*0.7920*	*0.8275*	*0.9823*	*0.9850*
IT2	*0.8050*	*0.8130*	*0.8200*	*0.9389*	*0.9450*
IT3	*0.8200*	*0.8300*	*0.8230*	*0.9772*	*0.9900*
IT4	*0.5100*	*0.6558*	*0.6175*	*0.7701*	*0.7225*
IT5	*0.5650*	*0.5771*	*0.5810*	*0.7147*	*0.6950*
IT6	*0.5850*	*0.5350*	*0.5625*	*0.7053*	*0.7125*
IT7	*0.4210*	*0.4250*	*0.4231*	*0.5879*	*0.6433*
Average	*0.6366*	*0.6611*	*0.6649*	*0.8109*	*0.8133*

15 percent over CMRF, while up to 20 percent is gained by SS-NMF over NMF. We also observe that SS-NMF can achieve high clustering accuracy (over 80 percent) in five out of the nine data sets. The average AC of SS-NMF is 72.43 percent, about 10 percent higher than that of SS-CMRF. In Fig. 6a, we plot the average AC values against increasing percentage of pairwise constraints for SS-CMRF and SS-NMF. Again, when more prior knowledge is available, the performance of SS-CMRF and SS-NMF clearly gets better. It is also obvious that on average SS-CMRF is consistently outperformed by SS-NMF with varying amounts of constraints.

In the left panel of Table 7, we report the accuracy of text categorization by SRC, CMRF, NMF, SS-CMRF, and SS-NMF. In six out of nine text data sets, the AC value of SS-NMF either ranks the best or the second with exceptions on the data sets: HT3, HT8, and HT9. This result shows



(a)



(b)

Fig. 6. Comparison of average clustering accuracy between SS-CMRF and SS-NMF with different amounts of constraints for (a) text high-order coclustering and (b) image high-order coclustering.

that even though the original *document-category* matrix is biased in the distance metric learning toward the constraints on the documents, SS-NMF still can provide competitive results on category clustering.

In high-order coclustering, we also obtain the clusters of words simultaneously with the clusters of documents and

TABLE 7
Text Categorization: Clustering Accuracy of Categories and Text Representation: Top 10 Words for Each Category

Name	*SRC*	*CMRF*	*NMF*	*SS-CMRF*	*SS-NMF*	*Representative words for each category*
HT1	*0.8*	*0.8*	*0.8*	*0.8*	*0.8*	{*via,coverag,calcium,purif,modifi,increm,identif,receiv,explant,delta*} {*market,pct,bank,rate,monei,billion,dollar,mln,dlr,currenc*}
HT2	*0.8*	*0.8*	*0.6*	*0.8*	*0.8*	{*studi,activ,patient,suggest,protein,increas,result,effect,treat,infect*} {*januari,pct,februari,resero,unemploy,billion,bank,fell,mln,rose*}
HT3	*0.4*	*0.7*	*0.8*	*0.8*	*0.6*	{*increas,patient,activ,perform,suggest,studi,effect,examin,result,factor*} {*februari,adjust,fall,sale,depart,retail,fell,season,level,month*}
HT4	*0.4*	*0.8*	*0.8*	*1.0*	*0.8*	{*cell,treatment,determin,site,bone,neutrophil,single,anim,change,differ*} {*consum,statist,index,inflat,rise,compar,base,month,increas,rose*}
HT5	*0.8*	*0.6*	*0.4*	*0.8*	*0.8*	{*pm,star,film,hollywood,set,releas,octob,director,time,million*} {*rise,price,rose,statist,unemploy,inflat,compar,consum,januari,increas*}
HT6	*0.8*	*0.8*	*0.6*	*0.8*	*0.8*	{*disregard,jai,pyramid,winner,aaron,baltimor,dean,leaf,ban,stanlei*} {*sahak,ohanus,melkonian,appressian,serazuma,armenian,serdar,escap, turkish,sdpa*}
HT7	*0.8*	*0.5*	*0.5*	*0.7*	*0.7*	{*mac,color,al,push,bit,sse,lower,size,traffic,screen*} {*licenc,egreeneast,clipper,drink,claim,biker,safeti,clean,dod,motorcycl*} {*vga,univ,pub,servic,educ,bill,robert,school,technic,game*}
HT8	*1.0*	*0.9*	*0.5*	*1.0*	*0.6*	{*intellect,chastiti,n3jxp,dsl,gebcadr,surrend,gebc,pitt,bank,shame*} {*ground,amp,heat,circuit,hot,increas,gif,voltag,factor,typic*}
HT9	*1.0*	*0.9*	*0.5*	*1.0*	*0.6*	{*strnlghtnetcom,sternlight,arm,escrow,clinton,clipper,wiretap,nsa,kei,tap*} {*flight,shuttl,launch,solar,moon,satellit,space,prbaccess,sky,planet*}

TABLE 8
Modality Importance for Text High-Order Coclustering: Word versus Category and for Image High-Order Coclustering: Color versus Texture

Name	*document-word*	*document-category*	*Name*	*image-color*	*image-texture*
HT1	*0.9996*	*0.3884*	*IT1*	*0.0001*	*0.2189*
HT2	*0.9999*	*0.4331*	*IT2*	*0.1890*	*0.0002*
HT3	*0.6837*	*0.9949*	*IT3*	*0.2188*	*0.0005*
HT4	*0.7607*	*0.7233*	*IT4*	*0.0088*	*0.2357*
HT5	*0.2479*	*0.9998*	*IT5*	*0.3040*	*0.0002*
HT6	*0.9999*	*0.1751*	*IT6*	*0.0001*	*0.2007*
HT7	*0.2390*	*0.9990*	*IT7*	*0.1102*	*0.0486*
HT8	*0.9996*	*0.5136*			
HT9	*0.9990*	*0.6577*			

categories. However, for text representation, there is no ground truth available to compute an AC value. Here, we select the "top" 10 words based on mutual information for each word cluster associated with a category cluster and list them in the right panel of Table 7. These words can be used to represent the underlying "concept" of the corresponding category cluster.

Modality selection. As described in Section 3.2, distance metric and modality importance are learned iteratively in Algorithm 1. First, modality selection can provide additional information on the relative importance of various relations (e.g., "word" and "category") for grouping the central data type (e.g., "document"). Moreover, from a technical point of view, it also acts like feature selection when computing the new relational data matrix. The left panel of Table 8 lists the modality importance for the two relations: *document-word* and *document-category* in SS-NMF with 1 percent constraints. A higher value in the table indicates more importance. It is clear that the significance of "word" and "category" are quite different in different data sets. Specifically, the *document-word* relation seems to play a more important role for document coclustering in all the data sets except HT3, HT5, and HT7, while the *document-category* relation is more important in the remainder. This information provides a better understanding of the underlying process that generates the document clusters.

5.4.2 Image High-Order Coclustering

Second, we present the experimental results on coclustering image data.

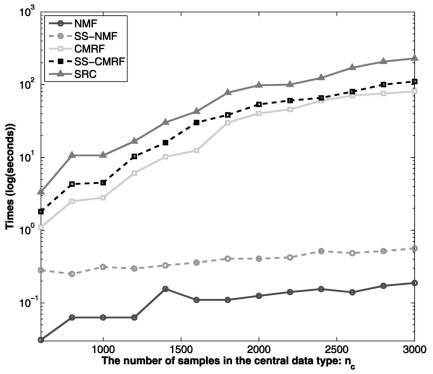
Coclustering accuracy. The bottom half of Table 6 lists image clustering accuracy obtained by SRC, CMRF, NMF, SS-CMRF, and SS-NMF (both with 15 percent constraints) for each data set, together with averaged AC value over all seven data sets. Among the three unsupervised approaches, on average NMF achieves slightly better results. Moreover, both of the semisupervised methods obtain 20 percent accuracy gain when compared with the corresponding unsupervised ones, and they perform equally well on most of the data sets. SS-NMF is slightly better than SS-CMRF on average. Fig. 6b shows that the quality of the clustering improves when the amount of constraints increases. Note that while we observe better performance of SS-NMF over SS-CMRF in text data sets, it is clear to see that the performance of SS-CMRF and SS-NMF is very close in image data sets regardless of the amount of constraints. This is mainly due to better performance of NMF in clustering high-dimensional data. The highest feature dimension is 1,000 in the text data, and only 45 for the image data.

Modality selection. The semantic gap between the low-level features and the high-level semantic concepts poses great challenge in content-based image retrieval. To this end, modality selection in coclustering is particularly beneficial because it not only provides the clusters of images, but also shows why certain images are grouped together. That is, important visual features are identified through simultaneous grouping with images. Specifically, the modality factor obtained by SS-LDA in our algorithm reflects the relative importance of various feature modalities such as color, texture, and shape in image grouping. The right panel of Table 8 lists the weights associated with color and texture given by SS-NMF with 3 percent constraints. Usually, images in the categories *eggs*, *decoys*, *buses*, *firearms*, and *cards* have strong edges. This visual observation is confirmed by our results, showing a larger weight for the texture features (e.g., Gab, EDH, and EDCV) than colors (e.g., RGB, CH, and CCV) in the data sets IT1 and IT4. On the other hand, we observe that colors may be better suited for clustering images in *dawn*, *foliage*, *wave*, *abstract*, and *texture*. In these categories, colors are relatively constant. For example, *dawn* usually has a red hue, while *foliage* has a dominate green hue. In these cases (data sets IT2 and IT5), the modality factors are also consistent with our visual judgment, with a larger value for color. Moreover, when we have many categories mixed together (e.g., data set IT7), we obtain relative balanced weights between color and texture. The result indicates that both modalities are important. If additional information regarding the image clusters is desired, it can be gained by examining the corresponding feature clusters obtained in the coclustering.

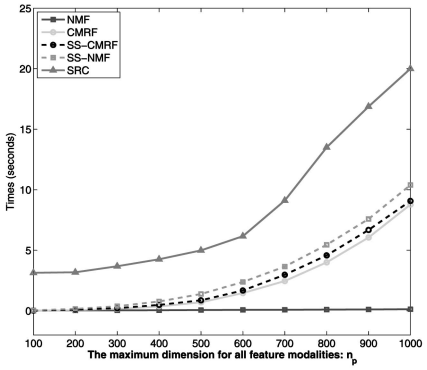
5.5 Time Complexity

Finally, we compare the computational speed of three unsupervised approaches: SRC, CMRF, and NMF, and two semisupervised approaches: SS-CMRF and SS-NMF. In a nutshell, the time complexity of SRC is $\mathcal{O}(tl(\max(n_c, n_p)^3 + kn_cn_p))$, unsupervised CMRF and SS-CMRF are $\mathcal{O}(tl(max(n_c^3, n_p^3)))$, SS-NMF is $\mathcal{O}(tl(n_p^3 + kn_cn_p))$, and unsupervised NMF is $\mathcal{O}(tlkn_cn_p)$, where t is the number of iterations, l is the number of data types, $k = \max(k_c, k_p)$ is the maximum number of clusters in all data types, n_c is the number of samples in the central data type, and n_p is the maximum feature dimension for all feature modalities. So, given t, l, and k, the actual computational speed is usually determined by n_c or n_p. Fig. 7a illustrates the computational speed for all five methods with increasing number of samples in the central data type n_c for a fixed n_p, while Fig. 7b shows the computational speed with increasing feature dimensions n_p for a fixed n_c. The experiments are performed on a machine with Dual 3 GHz Intel Xeon processors and 2 GB RAM. All algorithms are implemented using MATLAB 7.0.

In both cases, unsupervised NMF is the quickest among the five approaches as it uses an efficient iterative algorithm to compute the cluster indicator and cluster association matrices. SS-NMF ranks second as n_c increases while close to CMRF and SS-CMRF when n_p increases. The difference between SS-NMF and unsupervised NMF



(a)



(b)

Fig. 7. Comparison of computational speed between unsupervised approaches (SRC, CMRF, and NMF) and semisupervised approaches (SS-CMRF and SS-NMF). The time required by each of the algorithms are displayed (a) in log(seconds) for increasing n_c and (b) in seconds for increasing n_p.

is mainly due to the additional computation required to learn the new distance metric through SS-LDA, in which we need to solve a generalized eigen-problem. We observe that in Fig. 7a, the computing time for SS-NMF is close to unsupervised NMF because both methods have a linear complexity of n_c when n_p is fixed. On the other hand, as shown in Fig. 7b, time for SS-NMF increases more quickly ($\mathcal{O}(tln_p^3)$) when n_c is fixed. In addition, the speed of CMRF and SS-CMRF is between NMF and SRC. The computing time of these two algorithms increases quickly in both cases since their complexity is either ($\mathcal{O}(n_c^3)$) or ($\mathcal{O}(n_p^3)$) when the other is fixed. Moreover, we observe that SRC is the slowest in both cases. Even though SRC is completely unsupervised, it needs to solve a computationally more expensive constrained eigen-decomposition problem and requires additional postprocessing (k-means) to infer the clusters. From these results, it is obvious that SS-NMF provides an efficient way for semisupervised data coclustering.

6 Conclusions

In this paper, we present a novel semisupervised approach for data coclustering: SS-NMF. In the proposed SS-NMF coclustering model, users are able to provide supervision in terms of *must-link* and *cannot-link* constraints on the central data type, which are used to derive new relational matrices through iterative distance metric learning and modality selection. Trifactorizations of the new matrices are then performed to obtain the simultaneous grouping of central data type and multiple feature modalities. Theoretically, we prove the convergence and correctness of the proposed coclustering algorithm and show the relationship between SS-NMF with other data coclustering models. Our experimental results on publicly available data sets in text mining, bioinformatics, and image grouping show the superior performance of SS-NMF over existing methods for heterogeneous data coclustering.

Acknowledgments

This research was partially funded by the US National Science Foundation under grants IIS-0713315 and CNS-0751045, and by the 21st Century Jobs Fund Award, State of Michigan, under grant 06-1-P1-0193. The authors would like to express thanks to Sara Tipton in the English Department of Wayne State University for proofreading their paper. They also thank the anonymous reviewers for their constructive comments that greatly improved the paper.

References

[1] E.M. Airoldi, D.M. Blei, S.E. Fienberg, and E.P. Xing, "Mixed Membership Stochastic Blockmodels," *J. Machine Learning Research*, vol. 9, pp. 1981-2014, 2008.

[2] L. Baker and A. McCallum, "Distributional Clustering of Words for Text Classification," *Proc. 21st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 96-103, 1998.

[3] A. Banerjee, I.S. 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.

[4] R. Bekkerman and J. Jeon, "Multi-Modal Clustering for Multimedia Collections," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[5] R. Bekkerman and M. Sahami, "Semi-Supervised Clustering Using Combinatorial MRFs," *Proc. 23rd Int'l Conf. Machine Learning (ICML) Workshop Learning in Structured Output Spaces*, 2006.

[6] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. 11th Ann. Conf. Computational Learning Theory*, pp. 92-100, 1998.

[7] D. Cai, Z. Shao, X. He, X. Yan, and J. Han, "Mining Hidden Community in Heterogeneous Social Networks," *Proc. Workshop Link Discovery: Issues, Approaches and Applications*, pp. 58-65, 2005.

[8] Y. Chen, M. Dong, and W. Wang, "Image Co-Clustering with Multi-Modality Features from User Feedbacks," *Proc. ACM Int'l Conf. Multimedia*, 2009.

[9] Y. Chen, M. Rege, M. Dong, and J. Hua, "Incorporating User Provided Constraints into Document Clustering," *Proc. Seventh IEEE Int'l Conf. Data Mining*, pp. 103-112, 2007.

[10] Y. Chen, L. Wang, and M. Dong, "A Matrix-Based Approach for Semi-Supervised Document Co-Clustering," *Proc. 17th ACM Conf. Information and Knowledge Management*, pp. 1523-1524, 2008.

[11] Y. Chen, M. Rege, M. Dong, and J. Hua, "Non-Negative Matrix Factorization for Semi-Supervised Data Clustering," *J. Knowledge and Information Systems*, vol. 17, pp. 355-379, 2008.

[12] F.R.K. Chung, *Spectral Graph Theory*. Am. Math. Soc., 1997.

[13] 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.

[14] 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.

[15] C. Ding, "Unsupervised Feature Selection via Two-Way Ordering in Gene Expression Analysis," *Bioinformatics*, vol. 19, no. 10, pp. 1259-1266, 2003.

[16] C. Ding, T. Li, and W. Peng, "On the Equivalence between Non-Negative Matrix Factorization and Probabilistic Latent Semantic Indexing," *Computational Statistics and Data Analysis,* vol. 52, no. 8, pp. 3913-3927, 2008.
[17] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal Nonnegative Matrix Tri-Factorizations for Clustering," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 126-135, 2006.
[18] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* John Wiley & Sons, Inc., 2001.
[19] B. Gao, T.-Y. Liu, and W.-Y. Mao, "Star-Structured High-Order Heterogenous Data Co-Clustering Based on Consistent Information Theory," *Proc. Sixth IEEE Int'l Conf. Data Mining,* pp. 880-884, 2006.
[20] B. Gao, T.-Y. Liu, G. Feng, T. Qin, Q.-S. Cheng, and W.-Y. Ma, "Hierarchical Taxonomy Preparation for Text Categorization Using Consistent Bipartite Spectral Graph Copartitioning," *IEEE Trans. Knowledge and Data Eng.,* vol. 17, no. 9, pp. 1263-1273, Sept. 2005.
[21] Z. Ghahramani and M.I. Jordan, "Supervised Learning from Incomplete Data via the Em Approach," *Proc. Advances in Neural Information Processing Systems,* pp. 120-127, 1994.
[22] E.-H. Han and G. Karypis, "Centroid-Based Document Classification: Analysis and Experimental Results," *Proc. Fourth European Conf. Principles of Knowledge Discovery,* pp. 424-431, 2000.
[23] M. Hiu, C. Law, A. Topchy, and A. Jain, "Model-Based Clustering with Probabilistic Constraints," *Proc. Fifth SIAM Int'l Conf. Data Mining,* pp. 641-645, 2005.
[24] T. Hoffman and J. Puzicha, "Latent Class Models for Collaborative Filtering," *Proc. 16th Int'l Joint Conf. Artificial Intelligence,* pp. 688-693, 1999.
[25] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys,* vol. 31, no. 3, pp. 264-323, 1999.
[26] X. Ji and W. Xu, "Document Clustering with Prior Knowledge," *Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 405-412, 2006.
[27] T. Joachims, "Transductive Inference for Text Classification Using Support Vector Machines," *Proc. 16th Int'l Conf. Machine Learning,* pp. 200-209, 1999.
[28] C. Kemp, J.B. Tenenbaum, T.L. Griffiths, T. Yamada, and N. Ueda, "Learning Systems of Concepts with an Infinite Relational Model," *Proc. 21st Nat'l Conf. Artificial Intelligence,* 2006.
[29] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-Supervised Graph Clustering: A Kernel Approach," *Proc. 22nd Int'l Conf. Machine Learning,* pp. 457-464, 2005.
[30] K. Lang, "Newsweeder: Learning to Filter Netnews," *Proc. 12th Int'l Conf. Machine Learning,* pp. 331-339, 1995.
[31] D. Lee and H. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization," *Nature,* vol. 401, pp. 788-791, 1999.
[32] D. Lee and H. Seung, "Algorithms for Non-Negative Matrix Factorization," *Proc. 13th Advances in Neural Information Processing Systems,* pp. 556-562, 2001.
[33] X. Liu, Y. Gong, W. Xu, and S. Zhu, "Document Clustering with Cluster Refinement and Model Selection Capabilities," *Proc. 25th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 191-198, 2002.
[34] B. Long, X. Wu, Z. Zhang, and P.S. Yu, "Spectral Clustering for Multi-Type Relational Data," *Proc. 23rd Int'l Conf. Machine Learning,* pp. 585-592, 2006.
[35] B. Long, Z. Zhang, and P.S. Yu, "A Probabilistic Framework for Relational Clustering," *Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 470-479, 2007.
[36] W.-Y. Max and H. Zhang, "Benchmarking of Image Features for Content-Based Retrieval," *Proc. 32nd Asilomar Conf. Signals, Systems and Computers,* pp. 253-257, 1998.
[37] H. Shan and A. Banerjee, "Bayesian Co-Clustering," *Proc. 13th IEEE Int'l Conf. Data Mining,* pp. 530-539, 2008.
[38] A. Vailaya, A. Jain, and H. Zhang, "On Image Classification: City Images vs. Landscapes," *Pattern Recognition,* vol. 31, no. 12, pp. 1921-1935, 1998.
[39] P. Willett, "Recent Trends in Hierarchic Document Clustering: A Critical Review," *Int'l J. Information Processing and Management,* vol. 24, no. 5, pp. 577-597, 1988.
[40] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell, "Distance Metric Learning, with Application to Clustering with Side Information," *Proc. 16th Neural Information Processing Systems,* pp. 505-512, 2002.
[41] 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 Information Retrieval,* pp. 267-273, 2003.

Yanhua Chen received the MS degree in computer science and engineering from Michigan State University, East Lansing, in 2004. She is currently a PhD candidate in the Machine Vision and Pattern Recognition Laboratory of the Department of Computer Science, Wayne State University, Detroit, Michigan. Her research interests are in the areas of pattern recognition, machine learning, data mining, graph theory, and information retrieval. She is a student member of the IEEE.

Lijun Wang received her MS degree from the School of Computer Science and Technology, Harbin Institute of Technology, P.R. China. She is currently a PhD student in the Machine Vision and Pattern Recognition Laboratory of the Department of Computer Science, Wayne State University, Detroit, Michigan. Her research interests lie in the areas of data mining, information retrieval, and multimedia analysis.

Ming Dong received the BS degree from Shanghai Jiao Tong University, P.R. China, in 1995, and the PhD degree from the University of Cincinnati, Ohio, in 2001, both in electrical engineering. He is currently an associate professor of computer science and the director of the Machine Vision and Pattern Recognition Laboratory. His research interests include pattern recognition, data mining, and multimedia analysis. His research is currently supported by the US National Science Foundation (NSF) and the State of Michigan. He has published more than 70 technical articles, many in premium journals and conferences such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Visualization and Computer Graphics*, the *IEEE Transactions on Neural Networks*, the *IEEE Transactions on Computers*, the *IEEE Transactions on Fuzzy Systems*, the IEEE ICDM, the IEEE CVPR, the ACM Multimedia, and the WWW. He is currently an associate editor of *IEEE Transactions on Neural Networks, Pattern Analysis and Applications* (Springer) and was on the editorial board of the *International Journal of Semantic Web and Information Systems*, 2005-2006. He also serves as a program committee member for many related conferences. He is a member of the IEEE.