1-1-2010

Data Clustering And Visualization Through Matrix Factorization

Yanhua Chen
Wayne State University

DATA CLUSTERING AND VISUALIZATION THROUGH MATRIX FACTORIZATION

by

YANHUA CHEN

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2010

MAJOR: COMPUTER SCIENCE

Approved by:

_____

Advisor Date

_____

_____

_____

ACKNOWLEDGMENTS

First of all, I would like to express my special appreciation to my advisor, Dr. Ming Dong, for his guide of my professional development and an inexhaustible source of ideas through my Ph.D.program at Wayne State University. During these years, he has spent tremendous time and effort with me discussing research, teaching me to write papers, and answering my questions. Without his kind assistance and advice, this dissertation would not haven been completed.

Besides my advisor, I am very grateful to Dr. Jing Hua, Dr. Farshad Fotouhi, and Dr. William Grosky, for serving on my prospectus and dissertation committee. They gave me plenty of constructive suggestions and invaluable comments on this dissertation.

Also, I thank all the members of Machine Vision and Pattern Recognition Laboratory, past and present: Dr. Changbo Yang, Dr. Manjeet Rege, Dr. Xuanwen Luo, Yuanhong Li, and Lijun Wang, for having insightful discussions with me.

I would also like to express thanks to Sara Tipton in English Department of Wayne State University for her proofreading my dissertation.

Last but not least, I am greatly indebted to my husband, Shuqing Zeng, and my little cute daughter, Christine, for their love and support during the five-year of my Ph.D. program. Thanks also go to my parents: my father, Getao Chen, and my mother, Yuxing Wang, for giving me life in the first place, for believing in me, for educating me, for unconditional support and encouragement to pursue my interests, even when the interests went beyond their boundaries of experience, language, field, and geography. I hope I have made them as well as my whole family proud.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Data mining is the task of discovering the interesting patterns from large amounts of data. Data clustering and visualization are two important research fields in data mining. The most widely-used methods in data mining for prediction and data analysis are classification and clustering [36, 90]. Classification is a purely supervised learning model, whereas clustering is completely unsupervised. Recently, there has been a lot of interest in the combination between completely supervised and unsupervised learning in the data mining research community [93]. Frequently based on the results of classification or clustering, data visualization further provides a qualitative overview of large and complex data sets, summarizing data and assisting in identifying regions of interest [37].

In this chapter, we first give an overview of traditional supervised classification and unsupervised clustering, and then describe semi-supervised learning, which can produce considerable improvement in learning accuracy when combining classification and clustering. Second, we provide a brief introduction on data visualization. At last, we review the matrix factorization mathematical framework, which can be applied to effectively solve problems in data clustering and visualization.

1.1 Classification and Clustering

1.1.1 Classification

Classification is supervised learning, where a category label for each pattern in a training set is provided by a supervisor [36]. The goal of classification is to learn a function from the training data that gives the best predication of the class label of the test data set. Supervised classification can be divided into two main categories: *Generative model* and *Discriminative model* [92]. In the generative model, classification requires the estimate of the class-conditional

densities via the use of the Bayes rule (or probabilistic). In the discriminative framework, the focus is on optimizing certain error criterion through discriminant analysis (or geometric), where the boundaries are directly learnt from data. It can be shown that the discriminative model of classification has better generalization error than the generative model under certain assumptions, which has made discriminative classifiers such as Support Vector Machines (SVM) [110] very popular.

1.1.2 Clustering

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 [64, 36]. The purpose of clustering is to extract useful information from unlabeled data. Applications of data clustering are found in many fields, such as information discovery, text mining, web analysis, image grouping, medical diagnosis, and bioinformatics.

Homogenous Clustering

Traditional clustering algorithms focus on grouping one data type, also called as homogenous clustering. Hundreds of homogenous clustering methods can be categorized as the following two popular techniques: *Agglomerative hierarchical clustering* and *Iterative square-error partitional clustering*. Hierarchical techniques organize data in a nested sequence of groups which can be displayed in the form of a dendrogram or a tree. Square-error partitional algorithms (e.g., graph-theoretic clustering) attempt to obtain that partition which minimizes the within-cluster scatter or maximizes the between-cluster scatter. Partitional clustering techniques are used more frequently than hierarchical techniques in data mining applications since the related clustering algorithms are generally formalized as optimization problems and can be easily solved by iterative methods [25], approximation methods [63] or heuristic methods [70]. Among partitioning clustering algorithms, clustering based on spectral graph cut the-

ory has emerged as a popular method over the years with applications across various domains [38, 39, 40, 22, 103]. This method models the data objects as vertices of a weighted graph with edge weights representing the similarity between two data objects. Clustering is then obtained by "cutting" the graph vertices into different partitions. Partitioning of the graph is obtained by solving an eigenvalue problem where the clustering is inferred from the top eigenvectors.

Heterogeneous Clustering

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 co-clustering 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 gaining a full understanding of the data. Consequently, co-clustering is introduced in the data mining literature, for both two data types (i.e., *pairwise co-clustering*) [55, 26, 27, 2, 85, 81], and multiple (more than two) data types (i.e., *high-order co-clustering*) [112, 44, 45, 83, 84, 6]. Through co-clustering, 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.

1.2 Semi-supervised Learning

In many practical learning domains (e.g. text processing, bioinformatics), 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, semi-supervised learning, i.e. learning from a combination of both labeled and unlabeled data, has become a topic of significant recent interest. The framework of semi-supervised learning is applicable to both classification and clustering.

1.2.1 Semi-supervised Classification

Supervised classification has a fixed known set of categories, and category-labeled training data is used to induce a classification function. In semi-supervised setting, the training can also exploit additional unlabeled data, frequently resulting in a more accurate classification function. Several semi-supervised classification algorithms that use unlabeled data to improve classification accuracy have become popular in the past few years, which include co-training [10], transductive support vector machines [66], and using Expectation Maximization (EM) to incorporate unlabeled data into training [47, 93]. A good review of semi-supervised classification methods is given in [100].

1.2.2 Semi-supervised Clustering

Semi-supervised clustering uses class labels or pairwise constraints on data objects to aid unsupervised clustering [4, 75, 111, 114, 5, 76, 68]. It can group data using the categories of the initial labeled data as well as unlabeled data in order to modify the existing set of categories which reflect the whole regularities in the data. Two sources of information are usually available to a semi-supervised clustering method: the similarity distance measurement in unsupervised clustering and class labels or some pairwise constraints. For semi-supervised clustering to be profitable, these two sources of information should not completely contradict each other.

Existing methods for semi-supervised clustering based on source information generally fall into two categories: (1) *Semi-supervised clustering with labels*. The algorithms based

on prior knowledge is available in the form of labeled data: e.g., semi-supervised seeded or constrained k-means algorithm (SS-Seeded-Kmeans or SS-Constrained-Kmeans) enforces constraints to be satisfied during the cluster assignment in the clustering process [111]; (2) *Semi-supervised clustering with constraints*. The algorithms based on pairwise constraints are known: e.g, semi-supervised constraints partitioning k-means algorithm (SS-COP-Kmeans) initializes clusters and infers clustering based on neighborhoods derived from labeled examples [4]. Semi-supervised clustering algorithms have recently received a significant amount of attention in the machine learning and data mining communities since they can incorporate prior information about clusters into the algorithms to improve the clustering results [76].

1.3 Visualization

Data visualization is to present data and summary information using graphics, animation, 3-D displays, and other multimedia dimensional reduction tools. The main goal of data visualization is to communicate information based on learning (eg., classification or clustering) results with people clearly and effectively through graphical means.

A number of different techniques [113, 12, 24] were proposed in the literature for visualizing a large dataset, among which multidimensional projection is the most popular one. It is to map the raw data matrix into a d-dimensional space with $d = 1, 2, 3$ by employing dimensionality reduction techniques. The objective is to preserve in the projected space the distance relationships among the data in their original space. Depending on the choice of mapping functions, both linear (e.g., principle component analysis (PCA) [67]) and nonlinear (e.g., ISOMAP [107]) dimensionality reduction techniques have been proposed in the literature. However, all of above methods can not solve three real-world challenges in visualization applications, which can be summarized as follows: (1) *Scalability*: Facing the increasing amount of data, a major challenge is to develop scalable approaches that are able to process and map massive data sets in a low dimensional space. From a computational point of view, large data set significantly raises the bar on the efficiency of a processing algorithm; (2) *Accuracy*: A visualization model

for displaying data need to have a well-defined objective function and build-in mechanism to combat noises such that it can provide accurate visualization of data; (3) *Interpretability*: All data are shown in the limited visualization space. Thus, how to avoid overlaps and clearly display objects is highly important. Therefore, there is an urgent need to develop a visualization model which provides people a more efficient, effective and interpretable view of ever-growing large-scale data.

1.4 Mathematical Framework: Matrix Factorization

Recently, matrix-based method has emerged as an effective approach for analyzing data in the high dimensional space. Factorization of matrices is to decompose of a matrix into a product of other matrices, or factors, which when multiplied together give the original. There are many different matrix factorizations; each finds use among a particular class of problems. The well-known approaches include Principal Component Analysis (PCA) for multi-dimensional projection, Singular Value Decomposition (SVD) for matrix approximation and Non-negative Matrix Factorization (NMF) for data clustering.

In this dissertation, we focus on discussion of NMF. NMF was initially proposed for "parts-of-whole" decomposition [79] and later extended to provide a general framework for data clustering [29]. It can model widely varying data distributions and accomplish both hard and soft clustering simultaneously. Let $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n) \in R^{d \times n}$ be the data matrix with nonnegative elements. NMF factorizes $\mathbf{X}$ into two non-negative matrices,

$$\mathbf{X} \approx \mathbf{F}\mathbf{G}^T, \tag{1.1}$$

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 $\mathbf{x}_i$ associated with cluster k, and k is the number of clusters. The factorization is typically obtained by the least square minimization. A simple example of

NMF clustering is illustrated as follows:

$$\mathbf{X} = \begin{bmatrix} 0.185 & 0.326 & 0.761 & 2.799 & 2.375 & 2.970 & 2.585 \\ 0.508 & 0.380 & 0.884 & 2.134 & 2.374 & 2.342 & 2.524 \\ 0.452 & 0.887 & 0.457 & 2.065 & 2.484 & 2.253 & 2.163 \\ 1.486 & 1.843 & 1.858 & 0.566 & 0.103 & 0.417 & 0.269 \\ 1.496 & 1.806 & 1.610 & 0.612 & 0.158 & 0.560 & 0.784 \end{bmatrix}$$

$$\approx \mathbf{F}\mathbf{G}^T = \begin{bmatrix} 1.7621 & 0.2165 \\ 1.5164 & 0.3013 \\ 1.4388 & 0.3101 \\ 0.0000 & 1.0424 \\ 0.1327 & 0.9891 \end{bmatrix} \times$$

$$\begin{bmatrix} 0.0000 & 0.0000 & 0.0522 & 0.4740 & 0.5074 & 0.5203 & 0.4944 \\ 0.4924 & 0.6104 & 0.5686 & 0.1599 & 0.0213 & 0.1244 & 0.1419 \end{bmatrix}. \tag{1.2}$$

In Equation (1.2), based on the membership indicator $\mathbf{G}$, clearly the first three columns form one cluster, and the last four columns give another.

In [115], it is shown that NMF outperforms spectral methods, achieving higher clustering accuracy, less computation cost and more intuitive interpretability. In addition, NMF has been proved to be very useful for applications such as face recognition, text mining, multimedia analysis, and DNA gene expression grouping. Noticeable variations or extensions of NMF include block value decomposition [85], orthogonal-NMF [34], sparse-NMF [57], and convex-NMF [31]. To some extent, these methods can provide higher computational efficiency and better interpretability by imposing additional constraints such as orthogonality, sparsity or convexity in the factorization process, however, *they are still not applicable to clustering data with prior knowledge or visualize data at an extremely large scale*.

1.5 Problem Statement

In order to fuse the advantages of NMF into semi-supervised data clustering or data visualization, in this dissertation, we propose to integrate NMF-based mathematical framework into real-world data clustering and visualization applications, which leads to increased efficiency and better results. The major contribution of this dissertation can be summarized as follows:

- **How to incorporate prior or background knowledge into matrix factorization to improve the quality of clustering or co-clustering?**

  In Chapter 3, we proposed a Non-negative Matrix Factorization (NMF) [79, 78] based framework to incorporate prior knowledge into data clustering. Under the proposed Semi-Supervised NMF (SS-NMF) methodology, user is able to provide pairwise constraints on a few data objects specifying whether they "must" or "cannot" be clustered together. An iterative algorithm is derived to perform symmetric non-negative tri-factorization of the data similarity matrix.

  Moreover, in chapter 4, we proposed a Semi-Supervised NMF (SS-NMF) based framework to incorporate prior knowledge into heterogeneous data co-clustering. In the proposed SS-NMF co-clustering 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 model improves the quality of co-clustering by learning a new distance metric based on these constraints. Then tri-factorizations of the new data matrices are performed using an iterative algorithm, obtained with the learned distance metric, to infer the central data clusters while simultaneously deriving the clusters of related feature modalities and their correspondence saliency.

- **How to introduce matrix factorization into exemplar-based clustering to provide a high quality of large-scale data visualization?**

  In Chapter 5, we proposed a novel technique, Exemplar-based Visualization (EV), to visualize an extremely large data collection. Capitalizing on recent advances in matrix

approximation and factorization, EV presents a probabilistic multidimensional projection model in the low-rank subspace with a sound objective function, and the visualization is obtained through iterative optimization. By selecting the representative rows and columns, a compact approximation of the data is obtained, making the visualization highly efficient and effective. In addition, the selected exemplars neatly summarize the entire data set and greatly reduce the cognitive overload in the visualization, leading to easier interpretation of the complex data.

From a theoretical perspective, our matrix-based clustering approaches are mathematically rigorous. The convergence and correctness are proved. In addition, we show that the advantages of our models over existing ones. Experiments performed on various publicly available data sets demonstrate the superior performance of the proposed work.

The rest of this dissertation is organized as follows. Chapter 2 reviews the related work of homogeneous data clustering and heterogeneous data co-clustering, followed by semi-supervised data clustering and co-clustering. Specifically, we focus on matrix factorization method review since it is a general framework for data clustering. Then we overview some related methods on data visualization. In Chapter 3, we present a SS-NMF model for data clustering with algorithm description and experiments verification. A detailed discussion on SS-NMF for heterogenous data co-clustering model, algorithm and applications are showed in Chapter 4. Chapter 5 further introduces matrix factorization into exemplar-based clustering model and employs it into visualizing large-scale data collections. Finally, Chapter 6 summarizes and points towards the future work.

CHAPTER 2

RELATED WORK

In this chapter, we provide a review of related work. First, we describe the categories of traditional homogeneous clustering and heterogenous co-clustering methods that are well-known in the literature. Then we introduce the representative semi-supervised clustering and co-clustering models. Finally, we give an overview of data visualization techniques.

2.1 Data Clustering

2.1.1 Homogeneous Data Clustering

As explained in Section 1.1.2, there are two categories of clustering algorithms: *Hierarchical clustering* and *Partitional clustering*, depending on whether the algorithm clusters the data into a hierarchical structure or generates a flat partitioning of the data.

Hierarchical Clustering

Hierarchical clustering aims to obtain a hierarchy of clusters, called dendrogram, that shows how the clusters are related to each other. The clustering result of the data items can be obtained by cutting the dendrogram at a desired level. These methods proceed either by iteratively merging small clusters into larger ones (agglomerative algorithms) or by splitting large clusters (divisive algorithms). Based on these, it can be classified into the following categories:

- **Agglomerative Algorithm**:

  Agglomerative method creates the cluster dendrogram in a *bottom-up* agglomerative fashion, starting with each data point in its own cluster and merging clusters successively according to a similarity measure till a convergence criterion is reached, e.g., hierarchical agglomerative clustering [71], Birch [116], etc.

- **Divisive Algorithm**:

  Divisive method creates the cluster dendrogram in a *top-down* divisive fashion, where all the data points initially are in a single cluster. This cluster is then split successively according to some measurement till a convergence criterion is reached, e.g., Cobweb [41], recursive cluster-splitting using a statistical transformation [35], and PDDP (principal direction divisive partitioning) [11].

Partitional Clustering

Partitional clustering attempts to obtain a partition which minimizes the within-cluster scatter or maximizes the between-cluster scatter based on minimum square-error function. To guarantee that an optimum solution has been obtained, one has to examine all possible partitions of the n d-dimensional patterns into k clusters (for a given k), which is not computationally feasible. Therefore, various heuristic methods are used to reduce the search, however, there is no guarantee of optimality. Partitional clustering techniques are used more frequently than hierarchical techniques in the real clustering applications. It can be further divided into the following categories:

- **Density-based Algorithm**:

  These methods model clusters as dense regions and use different heuristics to find arbitrary shaped high-density regions in the input data space and group points accordingly. Well-known methods include Denclue, which tries to analytically model the overall density around a point [53], and WaveCluster, which uses wavelet-transform to find high-density regions [102]. Density-based methods typically have difficulty scaling up to very high dimensional data ($> 10,000$ dimensions), which are common in domains like text mining for instance.

- **Mixture-based Algorithm**:

  Mixture-based methods assume that the data items in a cluster are drawn from one of

several distributions (usually Gaussian) and attempt to estimate the parameters of all these distributions. The introduction of the Expectation Maximization (EM) algorithm in [25] was an important step in solving the parameter estimation problem. Mixture-based methods have a high computational complexity and make rather strong assumptions regarding the distribution of the data. Most mixture-based methods view each cluster as a single simple distribution (such as Gausssian distribution) and thus strongly constrain the shape of the clusters; this explains why we did not include these methods in the category of density-based clustering. The most popular mixture model-based clustering algorithm is k-means [88].

The k-means algorithm performs iterative relocation to partition a data set into k clusters, locally minimizing the overall distortion measurement between the data points and the cluster means (a.k.a. "centroids"). We use R^d to denote the d-dimensional real vector space; p denotes a probability density function; $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ denotes the set of n data points, where the i^{th} data point is a vector represented by $\mathbf{x}_i$ whose m^{th} component is x_{im}. The k-means algorithm creates a k-partitioning $\{\mathbf{X}_h\}_{h=1}^k$ of $\mathbf{X}$ so that if $\{\mathbf{f}_h\}_{h=1}^k$ represents the k partition centroids, then the following objective function

$$J_{k-means} = \sum_{h=1}^{k} \sum_{\mathbf{x}_i \in \mathbf{X}_h} \|\mathbf{x}_i - \mathbf{f}_h\|^2 \tag{2.1}$$

is locally minimized. Note that finding the global optima for the k-means objective function is an NP-complete problem [46]. Considering Y denotes the set of n cluster labels, where y_i is the cluster label of the i^{th} data point $\mathbf{x}_i$, that is $y_i \in \{h\}_{h=1}^k$, then an equivalent form of the k-means clustering objective function of Equation (2.1) is:

$$J_{k-means} = \sum_{\mathbf{x}_i \in \mathbf{X}} \|\mathbf{x}_i - \mathbf{f}_{y_i}\|^2. \tag{2.2}$$

- **Graph-theoretic Algorithm**:

Given an undirected graph $G = (V, E)$ which is constructed from the data set, each vertex $v_i \in V$ corresponding to a data point $\mathbf{x}_i$ and the weight of each edge $e_{ij} \in E$ corresponding to the similarity between the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ according to a domain-specific similarity measure. The k clustering problem becomes equivalent to finding the k-mincut in this graph, which is known to be a NP-complete problem for $k \geq 3$ [46]. One class of methods use heuristics to find low-cost cuts in G: methods like Rock [49] and Chameleon [69] group nodes based on the idea of defining neighborhoods using inter-connectivity of nodes in G, Metis [70] performs fast multi-level heuristics on G at multiple resolutions to give good partitions, while Opossum [104] uses a modified cut criterion to ensure that the resulting clusters are well-balanced according to a specified balancing criterion. The second class of methods for solving the graph partitioning problem take a real relaxation of the NP-complete discrete partitioning problem: these include spectral graph partitioning method that performs clustering by using the second eigenvector of the graph Laplacian to define a cut [91] and the isoperimetric graph partitioning model that performs clustering by solving a system of linear equations to find the lowest isoperimetric ratio [48].

Spectral clustering has many fundamental advantages compared to traditional mixture-based clustering algorithms such as k-means. Results obtained by spectral clustering very often outperform these traditional approaches, and it is very simple to implement and can be solved by computing eigenvalue/eigenvector problem. The most referenced algorithm proposed by Shi and Malik [103] is to minimize the Normalized Cut (NC) objective function as follows,

$$J_{NC} = \sum_{l=1}^{k} \frac{\mathbf{g}_l^T(\mathbf{D} - \mathbf{A})\mathbf{g}_l}{\mathbf{g}_l^T \mathbf{D} \mathbf{g}_l}, \tag{2.3}$$

where $\mathbf{g}$ is cluster indicator vector, $\mathbf{A}$ is pairwise similarity matrix, and $\mathbf{D}$ is diagonal

matrix.

General Clustering Framework: NMF

Nonnegative matrix factorization (NMF) is another recently developed data clustering method. It was initially proposed for "parts-of-whole" decomposition [79] and has been shown to be equivalent to the (kernel) k-means clustering and the Laplacian-based spectral clustering [29]. It can model widely varying data distributions and can do both hard and soft clustering simultaneously.

Let $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n) \in R^{d \times n}$ be the data matrix of nonnegative elements. The NMF factorizes $\mathbf{X}$ into two non-negative matrices, $\mathbf{X} \approx \mathbf{F}\mathbf{G}^T$, where $\mathbf{F} = (\mathbf{f}_1, ..., \mathbf{f}_k) \in R^{d \times k}$ is cluster centroid and $\mathbf{G} = (\mathbf{g}_1, ..., \mathbf{g}_k) \in R^{n \times k}$ is cluster indicator, and k is a pre-specified parameter. The factorization is obtained by the least square minimization. The NMF-based clustering method is described in Algorithm 1.

Algorithm 1 Nonnegative Matrix Factorization Clustering Algorithm

INPUT: Data matrix $\mathbf{X} \in R^{d \times n}$, number k of clusters to construct
OUTPUT: Clusters $\{\mathbf{X}_h\}_{h=1}^k$ with $\mathbf{Y}_h = \{i | \mathrm{x}_i \in \mathbf{X}_h\}$
METHOD:

1. Initialize $\mathbf{F}$ and $\mathbf{G}$ with nonnegative values,
2. Iterate for each $1 \leq i \leq n$, $1 \leq m \leq d$ and $1 \leq h \leq k$ until *convergence*,
   (a) Cluster centroid:
   $$\mathbf{F}_{mh} \leftarrow \mathbf{F}_{mh} \frac{(\mathbf{X}\mathbf{G})_{mh}}{(\mathbf{F}\mathbf{G}^T\mathbf{G})_{mh}},$$
   (b) Cluster indicator:
   $$\mathbf{G}_{ih} \leftarrow \mathbf{G}_{ih} \frac{(\mathbf{X}^T\mathbf{F})_{ih}}{(\mathbf{G}\mathbf{F}^T\mathbf{F})_{ih}}.$$

Ding et al. [29] theoretically analyzes the relationships among NMF, (kernel) k-means and spectral clustering. This theoretical work gives an unified framework of clustering algorithms.

- **NMF and k-means Clustering**

  Theoretically, NMF is inherently related to (kernel) k-means clustering [82].

  Theorem 1. *Orthogonal NMF*

$$J_{NMF} = \min_{\boldsymbol{F}\geq 0, \boldsymbol{G}\geq 0} \|\boldsymbol{X} - \boldsymbol{F}\boldsymbol{G}^T\|^2, \ s.t. \ \boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}, \tag{2.4}$$

  is equivalent to (kernel) k-means clustering.

  Proof. We write $J_{NMF} = \|\mathbf{X} - \mathbf{F}\mathbf{G}^T\|^2 = \text{Tr}(\mathbf{X}^T\mathbf{X} - 2\mathbf{F}^T\mathbf{X}\mathbf{G} + \mathbf{F}^T\mathbf{F})$. The zero gradient condition $\partial J_{NMF}/\partial \mathbf{F} = -2\mathbf{X}\mathbf{G} + 2\mathbf{F} = 0$ gives $\mathbf{F} = \mathbf{X}\mathbf{G}$. Thus $J_{NMF} = \text{Tr}(\mathbf{X}^T\mathbf{X} - \mathbf{G}^T\mathbf{X}^T\mathbf{X}\mathbf{G})$. Since $\text{Tr}(\mathbf{X}^T\mathbf{X})$ is a constant, the optimization problem becomes

$$\max_{\mathbf{G}\geq 0} \text{Tr}(\mathbf{G}^T\mathbf{X}^T\mathbf{X}\mathbf{G}) \ s.t. \ \mathbf{G}^T\mathbf{G} = \mathbf{I}. \tag{2.5}$$

  In addition, the k-means clustering is to minimize the objective function as,

$$J_{k-means} = \sum_{h=1}^{k} \sum_{\mathbf{x}_i \in \mathbf{X}_h} \|\mathbf{x}_i - \mathbf{f}_h\|^2$$

  where $\mathbf{f}_h$ is the cluster centroid of the h-th cluster. More generally, the kernel k-means maps $\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i)$. Thus, the objective function of kernel k-means becomes

$$J_{kernel-kmeans} = \min \sum_{h=1}^{k} \sum_{\mathbf{x}_i \in \mathbf{X}_h} \|\phi(\mathbf{x}_i) - \overline{\phi_h}\|^2, \tag{2.6}$$

  where $\overline{\phi_h}$ is the centroid in the feature space. This can be solved via the optimization problem,

$$\max_{\mathbf{G}\geq 0} \text{Tr}(\mathbf{G}^T\mathbf{A}\mathbf{G}) \ s.t. \ \mathbf{G}^T\mathbf{G} = \mathbf{I}, \tag{2.7}$$

  where $\mathbf{G}$ are the cluster indicators and $\mathbf{A}_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. Specifically, for k-means,

$\phi(\mathbf{x}_i) = \mathbf{x}_i$, $\mathbf{A}_{ij} = \mathbf{x}_i^T\mathbf{x}_j$. Therefore, it is proved that the objective function of NMF (Equation (2.5)) is equivalent to the objective function of (kernel) k-means (Equation (2.7)). We also note that Theorem 1 holds even if $\mathbf{X}$ and $\mathbf{F}$ are not nonnegative, i.e., $\mathbf{X}$ and $\mathbf{F}$ have mixed-sign entries. The proof is completed. □

NMF has clustering capabilities which is generally better than k-means. In k-means, an exact orthogonality of columns of cluster indictor $\mathbf{G}$ implies that each row of $\mathbf{G}$ can have only one nonzero element, which implies that each data object belongs only to 1 cluster. This is *hard* clustering. While in NMF, the near-orthogonality condition of $\mathbf{G}$ relaxes this a bit, i.e, each data object could belong fractionally to more than 1 cluster. This is *soft* clustering. Thus, NMF has better clustering flexibility.

- **NMF and Spectral Clustering**

In addition, we discuss the relationship between NMF and spectral clustering. There are three popular objectives in spectral clustering: the Ratio Cut [50], the Normalized Cut [103], and the MinMax Cut [30]. We are interested in the multi-way clustering objective functions as,

$$J = \min \sum_{h=1}^{k} \frac{s(\mathbf{X}_h, \overline{\mathbf{X}_h})}{\rho(\mathbf{X}_h)}, \tag{2.8}$$

where

$$\rho(\mathbf{X}_h) = \begin{cases} |\mathbf{X}_h| & \text{for Ratio Cut} \\ \sum_{\mathbf{x}_i \in \mathbf{X}_h} \mathbf{d}_i & \text{for Normalized Cut} \\ s(\mathbf{X}_h, \mathbf{X}_h) & \text{for MinMax Cut} \end{cases},$$

$\overline{\mathbf{X}_h}$ is the complement of subset $\mathbf{X}_h$ in graph G, $s(S, \bar{S}) = \sum_{i \in S} \sum_{j \in \bar{S}} a_{ij}$, $\mathbf{d}_i = \sum_j a_{ij}$, and a_{ij} is the $(ij)^{th}$ component of similarity matrix $\mathbf{A}$.

Here, we show that the minimization of these objective functions (e.g, Normalized Cut) can be equivalently carried out via the NMF.

Theorem 2. *Normalized Cut using pairwise similarity matrix* $\boldsymbol{A}$ *is equivalent to kernel* k*-means clustering with the kernel matrix* $\widetilde{\boldsymbol{A}} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$*, where* $\boldsymbol{D} = diag(\boldsymbol{d}_1, ..., \boldsymbol{d}_n)$*.*

Theorem 3. *Normalized Cut using similarity matrix* $\boldsymbol{A}$ *is equivalent to symmetric NMF,*

$$J_{NMF} = \min_{\boldsymbol{G}\geq 0} \|\widetilde{\boldsymbol{A}} - \boldsymbol{G}\boldsymbol{G}^T\|^2, \tag{2.9}$$

where $\boldsymbol{G}$ *is indicator vector as,*

$$\boldsymbol{G} = (\boldsymbol{g}_1, ...\boldsymbol{g}_k), \boldsymbol{g}_l^T\boldsymbol{g}_h = \delta_{lh}, \boldsymbol{g}_l = (0, ..., 0, \overbrace{1, ..., 1}^{n_l}, 0, ..., 0)^T)/n_l^{1/2}, \tag{2.10}$$

where $1 \leq l \leq k$*,* n_l *is the number of vectors in the* l*-th cluster.*

Proof. Let $\mathbf{g}_l$ be cluster indicators as in Equation (2.9). One can easily see that

$$s(\mathbf{X}_h, \overline{\mathbf{X}_h}) = \sum_{\mathbf{x}_i \in \mathbf{X}_h} \sum_{\mathbf{x}_j \in \overline{\mathbf{X}_h}} a_{ij} = \mathbf{g}_l^T(\mathbf{D} - \mathbf{A})\mathbf{g}_l,$$

and

$$\sum_{\mathbf{x}_i \in \mathbf{X}_h} \mathbf{d}_i = \mathbf{g}_l^T\mathbf{D}\mathbf{g}_l.$$

Define the scaled cluster indicator vector $\mathbf{z}_l = \mathbf{D}^{1/2}\mathbf{g}_l / \|\mathbf{D}^{1/2}\mathbf{g}_l\|$, which obeys the orthogonal condition $\mathbf{z}_l^T\mathbf{z}_h = \delta_{lh}$, or $\mathbf{Z}^T\mathbf{Z} = \mathbf{I}$, where $\mathbf{Z} = (\mathbf{z}_1, ..., \mathbf{z}_k)$. Substituting into the Normalized Cut objective function, we have

$$J_{NC} = \min \sum_{l=1}^{k} \frac{\mathbf{g}_l^T(\mathbf{D} - \mathbf{A})\mathbf{g}_l}{\mathbf{g}_l^T\mathbf{D}\mathbf{g}_l} = \min \sum_{l=1}^{k} \mathbf{z}_l^T(\mathbf{I} - \widetilde{\mathbf{A}})\mathbf{z}_l. \tag{2.11}$$

Since the first term is a constant, thus, the above minimization problem becomes

$$\max_{\mathbf{Z}\geq 0} \mathrm{Tr}(\mathbf{Z}^T\widetilde{\mathbf{A}}\mathbf{Z}) \; s.t. \; \mathbf{Z}^T\mathbf{Z} = \mathbf{I}. \tag{2.12}$$

Once the solution $\widehat{\mathbf{Z}}$ is obtained, we can recover $\mathbf{G}$ by optimizing

$$\min_{\mathbf{G}\geq 0} \sum_{l} \left\| \widehat{\mathbf{z}}_l - \frac{\mathbf{D}^{1/2}\mathbf{g}_l}{\|\mathbf{D}^{1/2}\mathbf{g}_l\|} \right\|^2. \tag{2.13}$$

The exact solution is $\mathbf{g}_l = \mathbf{D}^{-1/2}\widehat{\mathbf{z}}_l$, or $\mathbf{G} = \mathbf{D}^{-1/2}\mathbf{Z}$. It means that row i of $\mathbf{Z}$ is multiplied by a constant $\mathbf{d}_i^{-1/2}$. The relative weights across different clusters in the same row remain same. Thus, $\mathbf{G}$ represents the same clustering as $\mathbf{Z}$ does. As a result, the spectral clustering objective function (Equation (2.12) is identical to the NMF clustering objective function (Equation (2.5) if let $\mathbf{Z} = \mathbf{G}$ and $\widetilde{\mathbf{A}} = \mathbf{X}^T\mathbf{X}$. The proof is completed. □

Comparing to the spectral graph model, NMF does not require the derived cluster indicator space $\mathbf{G}$ to be orthogonal, and it guarantees that each data takes only non-negative values. These two characteristics make NMF superior to spectral clustering methods because of the following reasons: First, when overlap exists among clusters, NMF can still find a direction for each cluster, while the orthogonal requirement by the eigenvector computation makes the derived directions less likely to correspond to each of the clusters. Second, as the direct benefit of the above two NMF characteristics, the cluster membership of each data can be easily identified from NMF, while indicator space derived by the spectral clustering model does not provide a direct indication of the data partitions, and consequently, traditional data clustering methods such as k-means have to be applied in this eigenvector space to find the final set of data clusters. Third, standard factorization of a data matrix uses SVD as widely used in spectral clustering. However, for many data sets such as images and text, the original data matrices are nonnegative. A factorization such as SVD contain negative entries and thus has difficulty for clustering interpretation.

To summarize, NMF, kernel k-means clustering and spectral clustering are unified in a simple way: they are different prescriptions of the same problem with slightly different

constraints.

2.1.2 Heterogeneous Data Co-clustering

In general, co-clustering approaches can be divided into three categories: probability-based analysis, information-theory based models, and graph theoretic approaches.

Co-clustering with Probability-based Model

In the first category, Hoffman et al. [55] proposed the Probabilistic Latent Semantic Analysis (PLSA) model for co-occurrence 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 co-clustering. 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 co-clustering approaches, such as Infinite Relational Model [74], Mixed Membership Blockmodel [1] and Bayesian co-clustering [101], were introduced recently using different inference engines. Also recently, Long et al. proposed a high-order co-clustering framework, Mixed Membership Relational Clustering (MMRC) model [86], 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. Generally, probabilistic techniques provide good cluster accuracy but cost more computation overhead.

Co-clustering with Information Theory Model

Concerning the information-theory based models, Dhillon et al. [27] presented a pairwise co-clustering 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 [2] wherein any Bregman divergence can be used as the objective

function for co-clustering. Later, Gao et al. [45] extended pairwise information theoretic models to high-order data co-clustering. More recently, Bekkerman and Jeon [6] proposed the Combinatorial Markov Random Field (CMRF) algorithm for high-order co-clustering, 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.

Co-clustering with Graph Theory Model

Graph theoretical approaches have a well-defined objective function for data co-clustering and, thus, are widely used. Spectral learning, such as Bipartite Spectral Graph Partitioning (BSGP) [26], was proposed and applied to co-cluster 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 Co-partitioning (CBGC) using semi-definite programming for high-order data co-clustering and applied it to hierarchical text taxonomy preparation [44]. 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 association. More recently, Long et al. [83] proposed Spectral Relational Clustering (SRC), in which they formulated heterogeneous co-clustering as collective factorization on related matrices and derived a spectral algorithm to cluster multi-type interrelated data objects simultaneously. SRC provides more flexibility by lifting the requirement of one-to-one association in graph-based co-clustering. 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.

2.1.3 Semi-supervised Clustering

We provide a review of related work on using user provided information to improve data clustering. We first discuss some algorithms in which prior knowledge is in the form of labeled

data. Next, we describe other algorithms for which pairwise constraints are required to be known *a priori*.

Semi-supervised Clustering with Labels

SS-Constrained-Kmeans [111] and SS-Seeded-Kmeans [4] are the two well-known algorithms in semi-supervised clustering with labels. The SS-Constrained-Kmeans seeds the k-means algorithm with the given labeled data and keeps that labeling unchanged throughout the algorithm. Moreover, it is appropriate when the initial seed labeling is noise-free, or if the user does not want the labels of the seed data to change. On the other hand, the SS-Seeded-Kmeans algorithm changes the given labeling of the seed data during the course of the algorithm. Also, it is applicable in the presence of noisy seeds, since it does not enforce the seed labels to remain unchanged during the clustering iterations and can therefore abandon noisy seed labels after the initialization step. Semi-supervised clustering with labels has been successfully applied to the problem of document clustering. [56] proposed incorporating background knowledge into document clustering by enriching the text features using WordNet[1]. Recently, [58] presented a probabilistic generative model to incorporate extended feedback that allows the user and the algorithm to jointly arrive at coherent clusters that capture the categories of interest to the user. [94, 10, 66] proposed methods where the user provided class labels *a priori* to some of the documents. These algorithms use the labeled data to generate seed clusters that initialize a clustering algorithm, and use constraints generated from the labeled data to guide the clustering process.

Semi-supervised Clustering with Constraints

In certain applications, supervision in the form of class labels may be unavailable. For example, complete class labels may be unknown in the context of clustering for speaker identification in

[1] http://wordnet.princeton.edu

a conversation [3], or clustering GPS data for lane-finding [111]. In some domains, pairwise constraints occur naturally, e.g., the Database of Interacting Proteins (DIP) dataset contains information about proteins co-occurring in processes, which can be viewed as *must-link* constraints during clustering. Similarly, for document clustering, user knowledge about which few documents are related or unrelated can be incorporated to improve the clustering results. Moreover, it is easier for a user to provide feedback in the form of pairwise constraints than class labels, since providing constraints does not require the user to have significant prior knowledge about the categories in the data set. Amongst the various methods proposed for utilizing user provided constraints for semi-supervised clustering, two of the well-known include the semi-supervised kernel k-means (SS-KK) [76] and semi-supervised spectral clustering with normalized cuts (SS-SNC) [65]. While, SS-KK transforms the clustering distance measure by weighted kernel k-means with reward and penalty constraints to perform semi-supervised 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.

Even though the research on semi-supervised clustering have attracted substantial attention in the past years, to date, most semi-supervised clustering models are only applicable to homogeneous data. Recently, Bekkerman and Sahami proposed a semi-supervised CMRF model (SS-CMRF) for pairwise co-clustering [7] under the information theoretic framework. However, without proof of correctness and convergence, their approach is not mathematically rigorous. Comparatively speaking, semi-supervised on heterogenous clustering has limited research in the literature until now.

2.2 Data Visualization

Visualization enables us to browse intuitively through huge amounts of data and thus could expand the human ability for comprehending complex datasets. A number of different techniques [113, 12, 24] were proposed in the literature for visualizing a large dataset, among which multidimensional projection is the most popular one. Assuming $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..\mathbf{x}_n\} \in R^{d_1 \times n}$

with a high dimension d_1 is the data matrix where columns index the objects and rows denote the features appearing in them, multidimensional projection is to find the embedding of data $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..\mathbf{y}_n\} \in R^{d_2 \times n}$ in the visualization space, usually $d_2 = \{1, 2, 3\}$ and minimize $|\delta(\mathbf{x}_i, \mathbf{x}_j) - D(f(\mathbf{x}_i), f(\mathbf{x}_j))|$, where $\delta(\mathbf{x}_i, \mathbf{x}_j)$ is the original dissimilarity distance and $D(f(\mathbf{x}_i), f(\mathbf{x}_j))$ is the Euclidean distance between the corresponding two points in the projected space, and $f : \mathbf{X} \rightarrow \mathbf{Y}$ is a mapping function [106].

In general, multidimensional projection techniques [67, 23, 107, 99, 96] can be divided into two major categories based on the function f employed: *Linear projection* methods and *Non-linear projection* methods. Linear projection creates an orthogonal linear transformation that transforms the data to a new coordinate system such that the new variable is a linear combination of the original variables. Among such techniques, the widely known is Principle Component Analysis (PCA) [67]. However, many data sets contain essential nonlinear structures that are invisible to PCA. For those cases, non-linear projection methods, using information not contained in the covariance matrix, are more appropriate. Several approaches, such us multidimensional scaling (MDS) [23] and ISOMAP [107], have been proposed for reproducing nonlinear higher-dimensional structures on a lower-dimensional display, and they differ in how the different distances are weighted and how the function are optimized.

Although multidimensional project techniques can extract a low-dimensional representation of a high-dimensional dataset, most of them take no account of the latent structure in the given data. To this end, Least Square Projection (LSP) [96] first chooses a set of control points using k-medoids method [8] based on the number of classes and then obtains the projection through the least square approximation, in which the data are projected following the geometry defined by the control points. Recently, incorporating probabilistic semantic models into analyzing large datasets has attracted great research interests [42, 61] since it can provide a higher quality (i.e., more meaningful) visualization. In Probabilistic Latent Semantic Analysis (PLSA) [54], a class is modeled as a probability distribution over features, and salient objects

are embedded together even if they do not share any features through Parametric Embedding (PE) [60]. Consequently, the objects that tend to be associated with the same class would be embedded nearby, as would class that tend to have the similar objects associated with them.

Unfortunately, all the aforementioned methods are inapplicable to visualize a large-scale dataset. When dealing with tens of thousands of objects, for example, PCA will fail to run due to insufficient memory and high computational cost of solving the eigen problem. The ever-increasing online data collection presents an unprecedented challenge for the development of highly scalable methods that can be implemented in a linear polynomial time. More recently, hierarchical-clustering based methods [43, 95] are proposed to partially solve the memory and computation problem, in which a hierarchical cluster tree is first constructed using a recursive partitioning process, and then the elements of that tree are mapped to the lower dimensional space to create a visual representation. However, these methods lack a mathematically rigorous objective function to minimize f. In addition, all determinations are strictly based on local decisions, and the deterministic nature of the hierarchy technique prevents reevaluation after points are grouped into a node of tree. Therefore, an incorrect assignment made earlier in the process cannot be corrected.

On the other side, humans can gain insight into the information embedded in data more strongly when presented in the visual system. A list of the most common multi-dimensional visualization interfaces is presented in [72], for examples, iconic displays [21], dense pixel displays [73], stacked displays [62], parallel coordinates [59], etc. To this end, it is desired to supplement clustering algorithms by visualization models to build a user-friendly system, thus providing more meaningful insights into a complex data set.

CHAPTER 3

SEMI-SUPERVISED DATA CLUSTERING BASED ON NMF

In this chapter, we propose a Non-negative Matrix Factorization (NMF) based framework to incorporate prior knowledge into data clustering. Under the proposed Semi-Supervised NMF (SS-NMF) methodology, user is able to provide pairwise constraints on a few data objects specifying whether they "must" or "cannot" be clustered together. We derive an iterative algorithm to perform symmetric non-negative tri-factorization of the data similarity matrix. The correctness and convergence of the algorithm is proved by showing that the solution satisfied the KKT optimality and the algorithm is guaranteed to converge. We also prove that SS-NMF is a general and unified framework for semi-supervised clustering by establishing the relationship between SS-NMF and other existing semi-supervised clustering algorithms.

In the following, we first formulate the SS-NMF model in Section 3.1 and derive the agorithm in Section 3.2. Theoretically, we prove the correctness and convergence of the algorithm in Section 3.3.1. Equivalence of SS-NMF to SS-KK and SS-SNC is proven in Section 3.3.2, followed by a discussion of advantages of SS-NMF in Section 3.3.3. Finally, experiments performed on various publicly available data sets demonstrating the superior performance of the SS-NMF for clustering are illustrated in Section 3.4.

3.1 Model Formulation

We assume the data consists of n objects, and that d features have been extracted from each of the objects. Correspondingly, the data can be represented using a matrix $\mathbf{X} \in R^{d \times n}$ where columns index the data objects to be clustered and rows denote the features. An entry x_{mi} in this matrix denotes the value of feature m for object i.

We propose a SS-NMF model for data clustering. In the proposed model, we perform

symmetric non-negative tri-factorization of the similarity matrix $\mathbf{A} = \mathbf{X}^T\mathbf{X} \in R^{n\times n}$ as,

$$\mathbf{A} \approx \mathbf{GSG}^T, \tag{3.1}$$

where $\mathbf{G} \in R^{n\times k}$ is the cluster indicator matrix. An entry g_{ih} in $\mathbf{G}$ gives the degree of association of object $\mathbf{x}_i$ with cluster h. The cluster membership of an object is given by finding the cluster with the maximum association value. $\mathbf{S} \in R^{k\times k}$ is the cluster centroid matrix that gives a compact $k \times k$ representation of $\mathbf{X}$.

Supervision is provided as two sets of pairwise constraints on the data objects: *must-link* constraints C_{ML} and *cannot-link* constraints C_{CL}. Every pair, $(\mathbf{x}_i, \mathbf{x}_j) \in C_{ML}$ implies that $\mathbf{x}_i$ and $\mathbf{x}_j$ must belong to the same cluster. Similarly, all possible pairs $(\mathbf{x}_i, \mathbf{x}_j) \in C_{CL}$ implies that the two objects should belong to different clusters. The constraints are accompanied by associated violation cost matrix $\mathbf{W}$. An entry w_{ij} in this matrix denotes the cost of violating the constraint between $\mathbf{x}_i$ and $\mathbf{x}_j$, if such a constraint exists, that is, either $(\mathbf{x}_i, \mathbf{x}_j) \in C_{ML}$ or $(\mathbf{x}_i, \mathbf{x}_j) \in C_{CL}$. The model relies on a distortion measure $D : R^d \rightarrow R$, to compute distance between the data objects. For a given k, the goal is to partition the data objects into k disjoint clusters $\{\mathbf{X}_h\}_{h=1}^k$, such that the total distortion between the objects and the corresponding cluster representatives (i.e., centroid) is (locally) minimized according to the given distortion measure D, while constraint violations are kept to a minimum.

3.2 Algorithm Derivation

We define the objective function of SS-NMF as,

$$J_{SS-NMF} = \min_{\mathbf{S}\geq 0, \mathbf{G}\geq 0} \|\widetilde{\mathbf{A}} - \mathbf{GSG}^T\|^2, \tag{3.2}$$

where $\widetilde{\mathbf{A}} = \mathbf{A} - \mathbf{W}_{reward} + \mathbf{W}_{penalty}$ is affinity or similarity matrix $\mathbf{A}$ with constraints $\mathbf{W}_{reward} = \{w_{ij} | (\mathbf{x}_i, \mathbf{x}_j) \in C_{ML}, s.t. y_i = y_j\}$ and $\mathbf{W}_{penalty} = \{w_{ij} | (\mathbf{x}_i, \mathbf{x}_j) \in C_{CL}, s.t. y_i = y_j\}$, w_{ij} is the penalty cost for violating a constraint between $\mathbf{x}_i$ and $\mathbf{x}_j$, and y_i is the cluster label of $\mathbf{x}_i$.

$\mathbf{S} \in R^{k \times k}$ is the cluster centroid, and $\mathbf{G} \in R^{n \times k}$ is the cluster indicator.

We propose an iterative procedure for the minimization of Equation (3.2) where we update one factor while fixing the others. The updating rules are

$$\mathbf{S}_{ih} \leftarrow \mathbf{S}_{ih} \sqrt[2]{\frac{(\mathbf{G}^T \widetilde{\mathbf{A}} \mathbf{G})_{ih}}{(\mathbf{G}^T \mathbf{G} \mathbf{S} \mathbf{G}^T \mathbf{G})_{ih}}}, \tag{3.3}$$

$$\mathbf{G}_{ih} \leftarrow \mathbf{G}_{ih} \sqrt[4]{\frac{(\widetilde{\mathbf{A}} \mathbf{G} \mathbf{S})_{ih}}{(\mathbf{G} \mathbf{S} \mathbf{G}^T \mathbf{G} \mathbf{S})_{ih}}}. \tag{3.4}$$

Thus, the SS-NMF algorithm for data clustering can be illustrated in Algorithm 2.

Algorithm 2 SS-NMF Algorithm

INPUT: Data similarity matrix $\mathbf{A} \in R^{d \times d}$, number of clusters k, constraint penalty matrix $\mathbf{W}_{penalty}$, and constraint reward matrix $\mathbf{W}_{reward}$
OUTPUT: Clusters $\{\mathbf{X}_h\}_{h=1}^{k}$ with $\mathbf{Y}_h = \{i | \mathbf{x}_i \in \mathbf{X}_h\}$
METHOD:

1. Initialize $\mathbf{S}$ and $\mathbf{G}$ with non-negative values,
2. Construct $\widetilde{\mathbf{A}} = \mathbf{A} - \mathbf{W}_{reward} + \mathbf{W}_{penalty}$,
3. Iterate for each i and h until *convergence*,
   (a) Cluster centroid:
   $$\mathbf{S}_{ih} \leftarrow \mathbf{S}_{ih} \sqrt[2]{\frac{(\mathbf{G}^T \widetilde{\mathbf{A}} \mathbf{G})_{ih}}{(\mathbf{G}^T \mathbf{G} \mathbf{S} \mathbf{G}^T \mathbf{G})_{ih}}},$$
   (b) Cluster indicator:
   $$\mathbf{G}_{ih} \leftarrow \mathbf{G}_{ih} \sqrt[4]{\frac{(\widetilde{\mathbf{A}} \mathbf{G} \mathbf{S})_{ih}}{(\mathbf{G} \mathbf{S} \mathbf{G}^T \mathbf{G} \mathbf{S})_{ih}}}.$$

3.3 Theoretical Analysis

3.3.1 Algorithm Correctness and Convergence

We now prove the theoretical correctness and convergence of SS-NMF. Motivated by [85, 83, 34], we render the proof based on optimization theory, auxiliary function and several matrix

inequalities.

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 Equations (3.3) and (3.4), the solution satisfies the KKT optimality condition.*

   Proof. Following the standard theory of constrained optimization, we introduce the Lagrangian multipliers λ_1 and λ_2 to minimize the lagrangian function,

$$L(\mathbf{S}, \mathbf{G}, \lambda_1, \lambda_2) = \min_{\mathbf{S}\geq 0, \mathbf{G}\geq 0} \|\widetilde{\mathbf{A}} - \mathbf{G}\mathbf{S}\mathbf{G}^T\|^2 - \text{Tr}(\lambda_1 \mathbf{S}^T) - \text{Tr}(\lambda_2 \mathbf{G}^T). \tag{3.5}$$

   Based on the KKT complementarity conditions $\frac{\partial J}{\partial \mathbf{S}} = 0$ and $\frac{\partial J}{\partial \mathbf{G}} = 0$, we obtain the following two equations,

$$2\mathbf{G}^T\widetilde{\mathbf{A}}\mathbf{G} - 2\mathbf{G}^T\mathbf{G}\mathbf{S}\mathbf{G}^T\mathbf{G} + \lambda_1 = 0, \tag{3.6}$$

$$4\widetilde{\mathbf{A}}\mathbf{G}\mathbf{S} - 4\mathbf{G}\mathbf{S}\mathbf{G}^T\mathbf{G}\mathbf{S} + \lambda_2 = 0. \tag{3.7}$$

   Applying the Hadamard multiplication on both sides of Equations (3.6) and (3.7) by $\mathbf{S}$ and $\mathbf{G}$, respectively, and using KKT conditions of

$$\lambda_1 \odot \mathbf{S}^2 = 0,$$

$$\lambda_2 \odot \mathbf{G}^4 = 0,$$

   where $\odot$ denotes the Hadamard product of two matrices, we can prove that if $\mathbf{S}$ and $\mathbf{G}$ are a local minimizer of the objective function in Equation (3.5), the following equations

are satisfied,

$$(\mathbf{G}^T\widetilde{\mathbf{A}}\mathbf{G}) \odot \mathbf{S}^2 - (\mathbf{G}^T\mathbf{G}\mathbf{S}\mathbf{G}^T\mathbf{G}) \odot \mathbf{S}^2 = 0, \tag{3.8}$$

$$(\widetilde{\mathbf{A}}\mathbf{G}\mathbf{S}) \odot \mathbf{G}^4 - (\mathbf{G}\mathbf{S}\mathbf{G}^T\mathbf{G}\mathbf{S}) \odot \mathbf{G}^4 = 0. \tag{3.9}$$

Based on Equations (3.8) and (3.9), we derive the proposed updating rules of Equations (3.3) and (3.4). If the updating rules converge, the solution satisfies the KKT optimality condition. The proof is completed. □

2. **Convergence**

Next, we prove the convergence of the algorithm. In Propositions 2 and 3, we show that the objective function decreases monotonically under the two updating rules. This can be done by making use of an auxiliary function similar to that used in [78].

Proposition 2. *If $\boldsymbol{G}$ is a fixed matrix, then $J(\boldsymbol{S}) = \|\widetilde{\boldsymbol{A}} - \boldsymbol{G}\boldsymbol{S}\boldsymbol{G}^T\|^2 = Tr(\widetilde{\boldsymbol{A}}^T\widetilde{\boldsymbol{A}} - 2\boldsymbol{G}^T\widetilde{\boldsymbol{A}}^T\boldsymbol{G}\boldsymbol{S} + \boldsymbol{G}^T\boldsymbol{G}\boldsymbol{S}\boldsymbol{G}^T\boldsymbol{G}\boldsymbol{S}^T)$ decreases monotonically under the updating rule of Equation (3.3).*

Proof. A function $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)})$ is called an auxiliary function of $L(\mathbf{S}^{(t+1)})$ if it satisfies $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)}) \geq L(\mathbf{S}^{(t+1)})$ and $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)}) = L(\mathbf{S}^{(t+1)})$ for any $\mathbf{S}^{(t+1)}$ and $\mathbf{S}^{(t)}$. Define $\mathbf{S}^{(t+1)} = arg\min_{\mathbf{S}} F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)})$. By construction, $L(\mathbf{S}^{(t)}) = F(\mathbf{S}^{(t)}, \mathbf{S}^{(t)}) \geq F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)}) \geq L(\mathbf{S}^{(t+1)})$. Thus, $L(\mathbf{S}^{(t)})$ is monotonic decreasing (non-increasing).

The key step is to find appropriate auxiliary function $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)})$. Assuming $\mathbf{G}$ is fixed, we write

$$L(\mathbf{S}^{(t+1)}) = \text{Tr}(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{A}} - 2\mathbf{G}^T\widetilde{\mathbf{A}}^T\mathbf{G}\mathbf{S} + \mathbf{G}^T\mathbf{G}\mathbf{S}\mathbf{G}^T\mathbf{G}\mathbf{S}^T) \tag{3.10}$$

and show that

$$F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)}) = \| \widetilde{\mathbf{A}} \|^2 - \sum_{ih} 2(\mathbf{G}^T \widetilde{\mathbf{A}} \mathbf{G})_{ih} \mathbf{S}_{ih}^{(t)} (1 + log \frac{\mathbf{S}_{ih}^{(t+1)}}{\mathbf{S}_{ih}^{(t)}}) + \sum_{ih} \frac{(\mathbf{G}^T \mathbf{G} \mathbf{S}^{(t)} \mathbf{G}^T \mathbf{G})_{ih} \mathbf{S}_{ih}^{2(t+1)}}{\mathbf{S}_{ih}^{(t)}} \tag{3.11}$$

is an auxiliary function of $L(\mathbf{S}^{(t+1)})$.

First, we show that the inequality $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)}) \geq L(\mathbf{S}^{(t+1)})$ holds. We can see the second term in $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)})$ (aside from the negative sign) is always smaller than the second term in $L(\mathbf{S}^{(t+1)})$ because of the inequality $\frac{\mathbf{S}_{ih}^{(t+1)}}{\mathbf{S}_{ih}^{(t)}} \geq 1 + log(\frac{\mathbf{S}_{ih}^{(t+1)}}{\mathbf{S}_{ih}^{(t)}})$, $\forall \frac{\mathbf{S}_{ih}^{(t+1)}}{\mathbf{S}_{ih}^{(t)}} > 0$. In addition, the third term in $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)})$ is always bigger than the third term in $L(\mathbf{S}^{(t+1)})$ [34]. Thus, the condition $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)}) \geq L(\mathbf{S}^{(t+1)})$ holds. Second, we show the equality $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)}) = L(\mathbf{S}^{(t+1)})$ holds. It is obvious when $\mathbf{S}^{(t+1)} = \mathbf{S}^{(t)}$, the equality $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)}) = L(\mathbf{S}^{(t+1)})$ holds.

Therefore, $F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)})$ is an auxiliary function of $L(\mathbf{S}^{(t+1)})$. Since we have

$$\mathbf{S}^{(t+1)} = arg \min_{\mathbf{S}} F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)}), \tag{3.12}$$

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

$$\frac{\partial F(\mathbf{S}^{(t+1)}, \mathbf{S}^{(t)})}{\partial \mathbf{S}_{ih}^{(t+1)}} = -\sum_{ih} 2(\mathbf{G}^T \widetilde{\mathbf{A}} \mathbf{G})_{ih} \frac{\mathbf{S}_{ih}^{(t)}}{\mathbf{S}_{ih}^{(t+1)}} + 2 \sum_{ih} \frac{(\mathbf{G}^T \mathbf{G} \mathbf{S}^{(t)} \mathbf{G}^T \mathbf{G})_{ih} \mathbf{S}_{ih}^{(t+1)}}{\mathbf{S}_{ih}^{(t)}} = 0. \tag{3.13}$$

Thus, we can derive the updating rule of Equation (3.3) as $\mathbf{S}_{ih} \leftarrow \mathbf{S}_{ih} \sqrt[2]{\frac{(\mathbf{G}^T \widetilde{\mathbf{A}} \mathbf{G})_{ih}}{(\mathbf{G}^T \mathbf{G} \mathbf{S} \mathbf{G}^T \mathbf{G})_{ih}}}$. Under this updating rule, $J(\mathbf{S})$ decreases monotonically. The proof is completed. □

Proposition 3. *If $\boldsymbol{S}$ is a fixed matrix, $J(\boldsymbol{G}) = \|\widetilde{\boldsymbol{A}} - \boldsymbol{GSG}^T\|^2 = Tr(\widetilde{\boldsymbol{A}}^T\widetilde{\boldsymbol{A}} - 2\boldsymbol{G}^T\widetilde{\boldsymbol{A}}^T\boldsymbol{GS} + \boldsymbol{G}^T\boldsymbol{GSG}^T\boldsymbol{GS}^T)$ decreases monotonically under the updating rule of Equation (3.4).*

Proof. A function $F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)})$ is called an auxiliary function of $L(\mathbf{G}^{(t+1)})$ if it satisfies $F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)}) \geq L(\mathbf{G}^{(t+1)})$ and $F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)}) = L(\mathbf{G}^{(t+1)})$ for any $\mathbf{G}^{(t+1)}$ and $\mathbf{G}^{(t)}$. Define $\mathbf{G}^{(t+1)} = arg\min_{\mathbf{G}} F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)})$. By construction, $L(\mathbf{G}^{(t)}) = F(\mathbf{G}^{(t)}, \mathbf{G}^{(t)}) \geq F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)}) \geq L(\mathbf{G}^{(t+1)})$. Thus, $L(\mathbf{G}^{(t)})$ is monotonic decreasing (non-increasing).

The key step is to find appropriate auxiliary function $F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)})$. Assuming $\mathbf{S}$ is fixed, we write

$$L(\mathbf{G}^{(t+1)}) = \text{Tr}(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{A}} - 2\widetilde{\mathbf{A}}^T\mathbf{GSG}^T + \mathbf{G}^T\mathbf{SG}^T\mathbf{GS}^T\mathbf{G}^T) \tag{3.14}$$

and show that,

$$\begin{aligned} F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)}) =& \| \widetilde{\mathbf{A}} \|^2 \\ &- \sum_{ih} 2(\widetilde{\mathbf{A}}\mathbf{G}^{(t)}\mathbf{S})_{ih}\mathbf{G}^{(t)}_{ih}(1 + 2log\frac{\mathbf{G}^{(t+1)}_{ih}}{\mathbf{G}^{(t)}_{ih}}) \\ &+ \sum_{ih} \frac{(\mathbf{G}^{(t)}\mathbf{SG}^{T(t)}\mathbf{G}^{(t)}\mathbf{S})_{ih}\mathbf{G}^{4(t+1)}_{ih}}{\mathbf{G}^{4(t)}_{ih}} \end{aligned} \tag{3.15}$$

is an auxiliary function of $L(\mathbf{G}^{(t+1)})$.

Following the proof of Proposition 2, it is not difficult to prove $F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)})$ is an auxiliary function of $L(\mathbf{G}^{(t+1)})$. Since $\mathbf{G}^{(t+1)} = arg\min_{\mathbf{G}} F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)})$, $\mathbf{G}^{(t+1)}$ is given by the minimum of $F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)})$ while fixing $\mathbf{G}^{(t)}$. The minimum value is obtained by setting $\frac{\partial F(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)})}{\partial \mathbf{G}^{(t+1)}_{ih}} = 0$. Thus, we can derive the updating rule of Equation (3.4) as $\mathbf{G}_{ih} \leftarrow \mathbf{G}_{ih}\sqrt[4]{\frac{(\widetilde{\mathbf{A}}\mathbf{GS})_{ih}}{(\mathbf{GSG}^T\mathbf{GS})_{ih}}}$. Under this updating rule, $J(\mathbf{G})$ decreases monotonically. The proof is completed. □

3.3.2 Equivalence of SS-NMF and Other Semi-supervised Clustering Methods

We now show that SS-NMF is a general and unified framework for semi-supervised clustering by establishing the relationship between SS-NMF and other well-known semi-supervised clustering algorithms, i.e., semi-supervised kernel k-means (SS-KK) [76] and semi-supervised spectral clustering with normalized cuts (SS-SNC) [65]. In fact, both these algorithms can be considered to be special cases of SS-NMF.

Proposition 4. *Orthogonal SS-NMF clustering is equivalent to SS-KK clustering.*

Proof. The SS-NMF objective function is

$$J_{SS-NMF} = \min_{\mathbf{S}\geq 0, \mathbf{G}\geq 0} \|\widetilde{\mathbf{A}} - \mathbf{G}\mathbf{S}\mathbf{G}^T\|^2. \tag{3.16}$$

The equation can be written as, $J_{SS-NMF} = \|\widetilde{\mathbf{A}} - \mathbf{G}\mathbf{S}\mathbf{G}^T\|^2 = \|\widetilde{\mathbf{A}} - \mathbf{G}'\mathbf{G}'^T\|^2 = \text{Tr}(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{A}} - 2\mathbf{G}'^T\widetilde{\mathbf{A}}\mathbf{G}' + \mathbf{G}'^T\mathbf{G}')$ if let $\mathbf{S} = \mathbf{Q}^T\mathbf{Q}$ and $\mathbf{G}' = \mathbf{G}\mathbf{Q}^T$. Since $\text{Tr}(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{A}} + \mathbf{G}'^T\mathbf{G}')$ is a constant, the minimization of J becomes a maximization problem as,

$$\max_{\mathbf{G}'\geq 0} \text{Tr}(\mathbf{G}'^T\widetilde{\mathbf{A}}\mathbf{G}') \; s.t. \; \mathbf{G}'^T\mathbf{G}' = \mathbf{I}. \tag{3.17}$$

The SS-KK [76] is to minimize the objective function as,

$$J_{SS-KK} = \sum_{h=1}^{k}\sum_{i\in\mathbf{X}_h} \|\phi(\mathbf{x}_i) - \overline{\phi_h}\|^2 - \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in C_{ML}, s.t. y_i = y_j} w_{ij} + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in C_{CL}, s.t. y_i = y_j} w_{ij}, \tag{3.18}$$

where $\phi(\cdot)$ is the kernel function and $\overline{\phi_h}$ the centroid. Let $\mathbf{E}$ be the matrix of pairwise squared Euclidean distance among the data points, $\mathbf{W}$ the constraint matrix and $\mathbf{G}$ the cluster indicator.

Equation (3.18) becomes the minimization of the following function,

$$\min_{\mathbf{G}\geq 0} \text{Tr}(\mathbf{G}^T(\mathbf{E}-2\mathbf{W})\mathbf{G}) \; s.t. \; \mathbf{G}^T\mathbf{G}=\mathbf{I}. \tag{3.19}$$

We can convert the minimization of Equation (3.19) to a maximization of the problem as,

$$\max_{\mathbf{G}\geq 0} \text{Tr}(\mathbf{G}^T\mathbf{K}\mathbf{G}) \; s.t. \; \mathbf{G}^T\mathbf{G}=\mathbf{I}, \tag{3.20}$$

where $\mathbf{K}=\mathbf{A}+\mathbf{W}$ and $\mathbf{A}$ the similarity matrix.

It is clear that the objective function of SS-NMF (Equation (3.17)) is equivalent to that of SS-KK (Equation (3.20)) if $\mathbf{K}=\widetilde{\mathbf{A}}$. The $\mathbf{G}'$ in Equation (3.17) represents the same clustering as $\mathbf{G}$ of Equation (3.20) does. The proof is completed. □

Proposition 5. *Orthogonal SS-NMF clustering is equivalent to SS-SNC clustering.*

Proof. The objective function of SS-SNC [65] is,

$$J_{SS-SNC}=\min\sum_{h=1}^{k}\frac{\mathbf{g}_h^T(\widetilde{\mathbf{D}}-\widetilde{\mathbf{A}})\mathbf{g}_h}{\mathbf{g}_h^T\widetilde{\mathbf{D}}\mathbf{g}_h}=\min\sum_{h=1}^{k}\mathbf{z}_h^T(\mathbf{I}-\dot{\mathbf{A}})\mathbf{z}_h, \tag{3.21}$$

where $\widetilde{\mathbf{A}}=\mathbf{A}-\mathbf{W}_{reward}-\mathbf{W}_{penalty}$ is the pairwise similarity matrix with constraints, $\widetilde{\mathbf{D}}=\text{diag}(\widetilde{\mathbf{x}}_1,...,\widetilde{\mathbf{x}}_n)$ is the diagonal matrix, $\mathbf{g}_h$ is the cluster indictor, scaled cluster indicator vector $\mathbf{z}_h=\widetilde{\mathbf{D}}^{1/2}\mathbf{g}_h/\|\widetilde{\mathbf{D}}^{1/2}\mathbf{g}_h\|$, and $\dot{\mathbf{A}}=\mathbf{D}^{-1/2}\widetilde{\mathbf{A}}\mathbf{D}^{-1/2}$.

It can be shown that the minimization of Equation (3.21) becomes a maximization problem as,

$$\max_{\mathbf{Z}\geq 0} \text{Tr}(\mathbf{Z}^T\dot{\mathbf{A}}\mathbf{Z}) \; s.t. \; \mathbf{Z}^T\mathbf{Z}=\mathbf{I}. \tag{3.22}$$

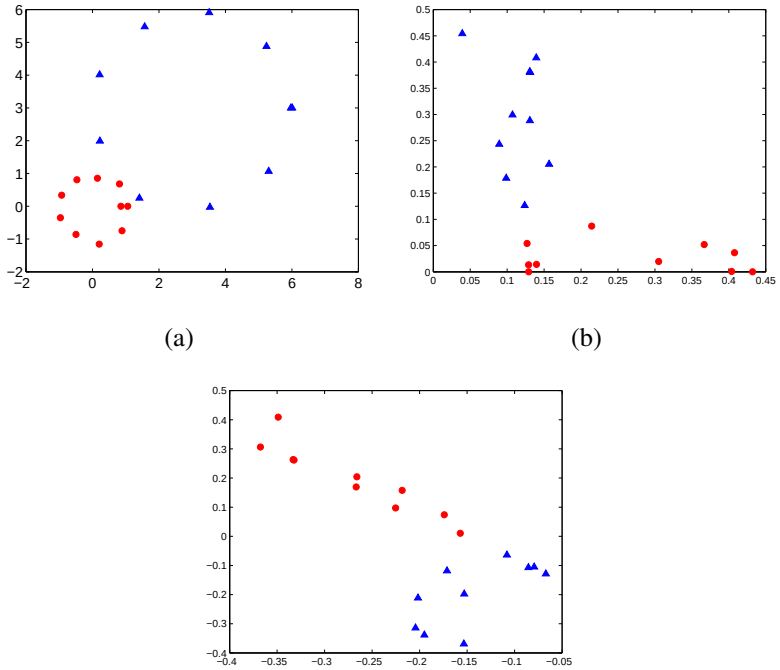
Also, it can be seen that Equation (3.17) is equivalent to Equation (3.22) if $\widetilde{\mathbf{A}}=\dot{\mathbf{A}}$. Moreover, the $\mathbf{G}'$ in Equation (3.17) represents the same clustering as $\mathbf{Z}$ of Equation (3.22) does. The proof is completed. □

From the above two proofs, we can see that SS-NMF, SS-KK, and SS-SNC are mathematically equivalent. However, notice that in SS-NMF, the matrix $\widetilde{\mathbf{A}}$ might have some negative values, which is not permitted in traditional NMF. In this case, one possible solution is to perform some normalization techniques to guarantee non-negative values. Alternatively, we can simply relax the non-negative constraint to allow negative values as in Semi-NMF [82]. In either of the approaches, the clustering result will not get affected too much. In SS-NMF, the cluster indicator $\mathbf{G}'$ is near-orthogonal and can produce soft clustering results. The cluster centroid $\mathbf{S}$ can provide good characterization of the quality of data clustering because the residue of the matrix approximation $J = \min \|\widetilde{\mathrm{A}} - \mathbf{GSG}^T\|$ is smaller than $J = \min \|\widetilde{\mathbf{A}} - \mathbf{GG}^T\|$. On the other hand, for SS-KK and SS-SNC, if input matrix is added with constraint weight $\mathbf{W}$, in order to ensure positive definiteness, certain additive constraints need to be enforced. Moreover, these constraints are difficult to be relaxed. Also, the cluster indicator $\mathbf{G}$ or $\mathbf{Z}$ is required to be orthogonal, leading to only hard clustering results. Hence, both SS-KK and SS-SNC can be viewed as special cases of SS-NMF with orthogonal space constraints. Thus, SS-NMF essentially provides a general and unified mathematical framework for semi-supervised data clustering.

3.3.3 Advantages of SS-NMF

Then, we further illustrate the advantages of SS-NMF using a toy data set shown in Figure 3.1a, which follows an extreme distribution consisting of 20 data points forming two natural clusters: two circular rings with 10 data points each. Traditional unsupervised clustering methods, such as (kernel) k-means, spectral normalized cut or NMF, are unable to produce satisfactory results on this data set. However, after incorporating knowledge from the user in the form of constraints, we are able to achieve much better results.

Unlike SS-SNC, SS-NMF maps the samples into a non-negative latent semantic space. Moreover, SS-NMF does not require the derived space to be orthogonal. Figures 3.1b and c show the data distributions in the two spaces for SS-NMF and SS-SNC, respectively. Data



(a) (b)

(c)

Figure 3.1: (a) An artificial toy dataset consisting of two natural clusters. (b) Data distribution in the SS-NMF subspace of the two column vectors of $\mathbf{G}$. The data points from the two clusters get distributed along the two axes. (c) Data distribution in the SS-SNC subspace of the first two singular vectors. There is no relationship between the axes and the clusters.

Table 3.1: Cluster indicator $\mathbf{G}$ of SS-KK and SS-NMF for the toy data set.

$\mathbf{G}$	*SS-KK*		*SS-NMF*	
$\mathbf{g}_1$	1	0	0.2778	0.0820
$\mathbf{g}_2$	1	0	0.2977	0.0486
$\mathbf{g}_3$	1	0	0.4301	0.0009
$\mathbf{g}_4$	1	0	0.1295	0.0494
$\mathbf{g}_5$	1	0	0.1377	0.0021
$\mathbf{g}_6$	1	0	0.3845	0.0000
$\mathbf{g}_7$	1	0	0.1281	0.0001
$\mathbf{g}_8$	1	0	0.1426	0.0097
$\mathbf{g}_9$	1	0	0.3119	0.0023
$\mathbf{g}_{10}$	1	0	0.4691	0.0080
$\mathbf{g}_{11}$	0	1	0.0651	0.3959
$\mathbf{g}_{12}$	0	1	0.0599	0.4449
$\mathbf{g}_{13}$	0	1	0.1161	0.4108
$\mathbf{g}_{14}$	0	1	0.0978	0.2985
$\mathbf{g}_{15}$	0	1	0.0592	0.2506
$\mathbf{g}_{16}$	1	0	0.1220	0.1233
$\mathbf{g}_{17}$	0	1	0.1047	0.1735
$\mathbf{g}_{18}$	0	1	0.1503	0.2028
$\mathbf{g}_{19}$	0	1	0.1233	0.2866
$\mathbf{g}_{20}$	0	1	0.1181	0.3800

points belonging to the same cluster are depicted by the same symbol. For SS-NMF, we plot the data points in the space of two column vectors of $\mathbf{G}$, while for SS-SNC the first two singular vectors are used. Clearly, in the SS-NMF space, every data point takes non-negative values in both the directions. Furthermore, in SS-NMF space, each axis corresponds to a cluster, and all the data points belonging to the same cluster are nicely spread along the axis. The cluster label for a data point can be determined by finding the axis with which the data point has the largest projection value. However, in the SS-SNC space, there is no direct relationship between the axes (singular vectors) and the clusters.

Table 3.1 shows the difference of cluster indicator between the hard clustering of SS-KK and soft clustering of SS-NMF. An exact orthogonality in SS-KK means that each row of cluster indicator $\mathbf{G}$ has only one nonzero element, which implies that each data object belongs to only 1 cluster. The near-orthogonality of cluster indicator $\mathbf{G}$ in SS-NMF relaxes this a bit, i.e., each data object could belong fractionally to more than 1 cluster. This can help in knowledge discovery in the cases where the data point is evenly projected along the different axes. For instance, $\mathbf{g}_{16} = \{0.1220, 0.1233\}$ indicates that this data point may belong to any one of the two clusters.

SS-NMF uses an efficient iterative algorithm instead of solving a computationally expensive constrained eigen decomposition problem as in SS-SNC. The time complexity of SS-NMF is $\mathcal{O}(tkn^2)$ where k is the number of clusters, n is the number of data objects, and t is the number of iterations. In fact, the time complexity is similar to that of the classical SS-KK clustering algorithm. However, compared to SS-KK, SS-NMF algorithm is simple as it only involves some basic matrix operations and hence can be easily deployed over a distributed computing environment when dealing with large data sets. Another advantage in favor of SS-NMF is that a partial answer can be obtained at intermediate stages of the solution by specifying a fixed number of iterations.

In Figure 3.2, we demonstrate the computational speed of SS-NMF with respect to SS-KK

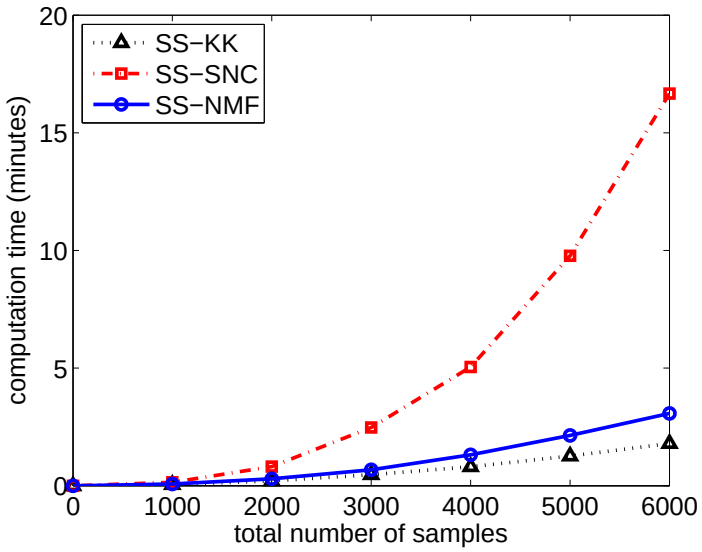


Figure 3.2: Computational speed comparison for SS-KK, SS-SNC and SS-NMF.

and SS-SNC. This experiment was performed on a machine with a 3 GHz Intel Pentium 2 processor with 2 GB RAM. As the number of data samples increase, SS-SNC turns out to be the slowest of the three algorithms. SS-KK is the quickest with SS-NMF closely following it. In the next section, we show the superior performance of SS-NMF in terms of clustering accuracy in comparison with other clustering algorithms.

3.4 Experiments and Results

In this section, we empirically demonstrate the performance of SS-NMF for data clustering. We present the details of our experiments, starting with the descriptions of the data sets (Section 3.4.1), the methodology and evaluation metrics (Section 3.4.2), followed by thorough performance comparisons with leading unsupervised and semi-supervised clustering algorithms (Section 3.4.3).

3.4.1 Data Description

We thoroughly evaluate the proposed algorithm on a variety of data sets, with number of classes ranging from 2 to 10, having between 27 to 500 data samples, and the dimensionality

(attributes) ranging from 4 to 12,600. These data sets represent applications from different domains such as text mining, image grouping and bioinformatics.

1. **Text Data Sets**

   We use eight text data sets for document clustering. In particular, we created the data sets by mixing some of the data sets used in [51] [1]. Data sets $oh0$ and $oh5$ are from OHSUMED collection [52], a subset of MEDLINE database, which contains $233,445$ documents indexed using $14,321$ unique categories. Data set $re0$ is from Reuters-21578 text categorization collection Distribution 1.0 [80]. Data set $Fbis$ is from the Foreign Broadcast Information Service data of TREC-5 [108]. For all data sets, the common words are removed and the words are stemmed using Porter's suffix-stripping algorithm [98].

   Table 3.2 shows the document data sets used in our experiments. These data sets were created as follows:

   - Classes $Graft$-$Survival$ and $Phospholipids$ from $oh5$ were mixed to form the $Graft$-$Phos$ data set.
   - Data set $England$-$Heart$ was created by mixing classes $England$ and $Heart$-$Valve$-$Prosthesis$ from $oh0$.
   - $Interest$-$Trade$ was formed by mixing $Interest$ and $Trade$ classes of $re0$ data set.
   - We randomly selected 2, 3, 4, 5, and 10 classes from *Fbis* to form data sets *Fbis2*, *Fbis3*, *Fbis4*, *Fbis5* and *Fbis10*, respectively.

   In addition, we performed feature selection on the words by retaining the top 10% of the words based on mutual information in each of the data sets.

[1] http://www.cs.umn.edu/~han/data/tmdata.tar.gz

Table 3.2: Summary of text data sets used in the experiments.

Data sets	*No. of clusters*	*No. of words*	*No. of docs*
Graft-Phos	*2*	*2432*	*293*
England-Heart	*2*	*2504*	*375*
Interest-Trade	*2*	*2682*	*438*
Fbis2	*2*	*2000*	*200*
Fbis3	*3*	*2000*	*300*
Fbis4	*4*	*2000*	*400*
Fbis5	*5*	*2000*	*500*
Fbis10	*10*	*2000*	*500*

2. **Gene Expression Data Sets**

   The five data sets used in our experiments from Kent Ridge Biomedical Data Repository[2] are: *AML/ALL*, *Colon Tumor*, *Prostate Cancer*, *ALL/MLL/AML*, and *Central Nervous System (CNS)*.

   - The *ALL/AML* data set includes two types of human tumor-acute myelogenous leukemia (*AML*, 11 samples) and acute lymphoblastic leukemia (*ALL*, 27 samples).
   - The *Colon Tumor* data set contains 62 samples collected from colon-cancer patients. Among them, 40 tumor biopsies are from tumors and 22 normal biopsies are from healthy parts of the colons of the same patients. $2,000$ out of around $6,500$ genes were selected based on the confidence in the measured expression levels.
   - The *Prostate Cancer* data set contains 52 prostate tumor samples and 50 non-tumor prostate samples with around $12,600$ genes.
   - The *ALL/MLL/AML* data set contains 57 leukemia samples which are divided into 20 *ALL*, 17 *MLL* and 20 *AML*.
   - The *Central Nervous System (CNS)* data set consists of 34 samples: 10 classic medulloblastoms, 10 malignantgliomas, 10 rhabdoids and 4 normals.

   These data sets are summarized in Table 3.3.

[2] http://datam.i2r.a-star.edu.sg/datasets/krbd/

Table 3.3: Summary of gene expression data sets used in the experiments.

Data sets	*No. of clusters*	*No. of genes*	*No. of samples*
ALL/AML	*2*	*7129*	*38*
Colon Tumor	*2*	*2000*	*62*
Prostate Tumor	*2*	*12600*	*102*
ALL/MLL/AML	*3*	*12582*	*57*
CNS	*4*	*7129*	*34*

3. **Image Data Sets**

   All the images used in our experiments come from the Corel image database [3]. Each image category has images between 100 and 300, and the total number of image is 10,000. Each Corel category is treated as a human-labeled cluster and is used as ground truth for our clustering task. Some sample images are shown in Figure 3.3. The image categories used in our experiments were *Owls*, *Roses*, *Lions*, *Elephants* and *Horses*. We refer to the categories using the first alphabet in the figures and tables as O, R, L, E, and H, respectively. The entire image database consists of $1,500$ images with 300 images in each category. For the image features, we adopted the HSV space and performed principal component analysis (PCA) along H, S and V dimensions separately. The image was then projected in the eigen vector space to get weights along the principal components. A feature vector for each image was formed by concatenating weights along the three dimensions.

4. **UCI Data Sets**

   We utilize three data sets from the UCI data repository [4]: *Iris*, *LettersIJL*, and *Soybean*.

   - *Iris* plant data contains three classes: Iris Setosa, Iris Versicolour and Iris Virginica with four attributes sepal length, sepal width, petal length and petal width.
   - *LettersIJL* is a randomly sampled subset of three letters I, J, L with 300 samples from Letters data set.

[3] http://wang.ist.psu.edu/docs/related/
[4] http://archive.ics.uci.edu/ml/

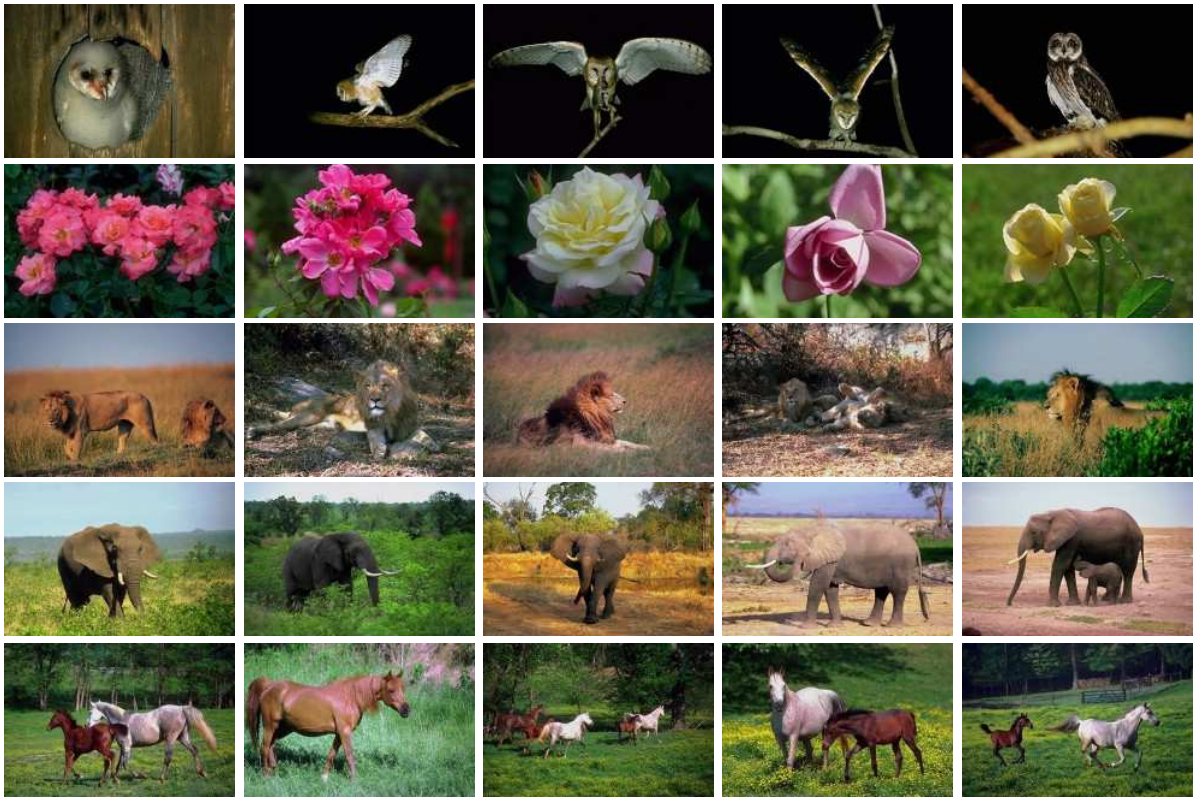


Figure 3.3: Image samples for semi-supervised clustering.

- *Soybean* comes from Soybean Small data with 4 classes: D1, D2, D3 and D4.

The data sets are summarized in Table 3.4.

Table 3.4: Summary of UCI data sets used in the experiments.

Data sets	*No. of clusters*	*No. of attributes*	*No. of samples*
Iris	*3*	*4*	*150*
LettersIJL	*3*	*16*	*300*
Soybean	*4*	*35*	*47*

3.4.2 Methodology and Evaluation Metrics

We compare the performance of SS-NMF model on all data sets with the following six clustering methods: (1) k-means, (2) kernel k-means, (3) spectral normalized cuts, (4) NMF, (5) SS-KK, (6) SS-SNC. The first four methods are the most popular unsupervised data clustering methods, whereas SS-KK and SS-SNC are the representative semi-supervised ones. Through these comparison studies, we demonstrate the relative position of SS-NMF with respect to unsupervised and semi-supervised approaches in real-world data clustering.

We evaluate the clustering results using confusion matrix and the accuracy metric AC. Each entry (i, j) in the confusion matrix represents the number of objects in cluster i that belong to true class j. The AC metric measures how accurately a learning method assigns labels $\hat{y}_i$ to the ground truth y_i, and is defined as,

$$AC = \frac{\sum_{i=1}^{n} \delta(y_i, \hat{y}_i)}{n}, \tag{3.23}$$

where n denotes the total number of objects in the experiment, and δ is the delta function that equals one if $\hat{y}_i = y_i$; otherwise it is zero. Since 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 one trial with a minimal objective value. In reality, usually a few number of trials is sufficient. In the case of NMF and k-means, for a given k, we conduct 20 test runs. 3 trials

are performed in each of the 20 test runs and final accuracy value is the average of all the test runs.

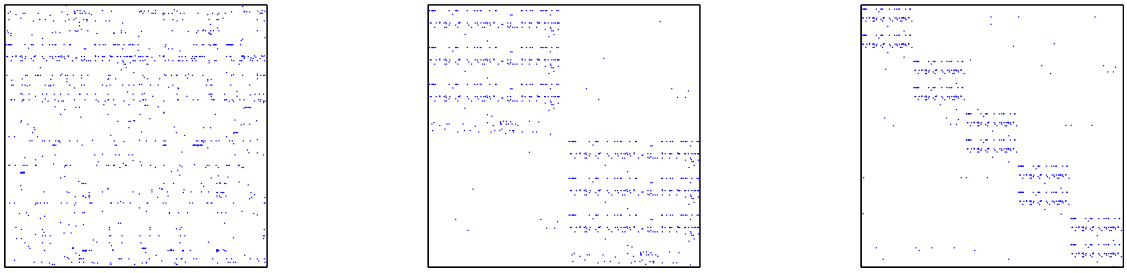
3.4.3 Results

1. **Document Clustering**

   We first perform comparison of the four unsupervised clustering approaches with SS-NMF having pairwise constraints on only 3% pairs of all the possible document pairs, which is $\binom{\text{total docs}}{2}$. Each of the constraints were generated by randomly selecting a pair of documents. Other datasets also use similar defined constraints in the following experiments. If both the documents have the same class label (*must-link*) , then the constraint is assigned maximum weight in the document-document similarity matrix. On the other hand, if they belong to different classes (*cannot-link*), then the minimum weight in the similarity matrix is used for the constraint. For kernel k-means, we used a Gaussian (exponential) kernel $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2/2\sigma^2)$, with variance $\sigma = 0.00001$ for 2 clusters and $\sigma = 0.01$ for more than 2 clusters. In Table 3.5, we compare the algorithms on all the text data sets using AC values. The performance of the first three methods is similar with NMF proving to be the best amongst the unsupervised methods. However, the accuracy of NMF greatly deteriorates and is unable to produce meaningful results on data sets having more than 2 clusters. On the other hand, the superior performance of SS-NMF is evident across all the data sets. We can see that in general a semi-supervised method can greatly enhance the document clustering results by benefitting from the user provided knowledge. Moreover, SS-NMF is able to generate significantly better results by quickly learning from the few pairwise constraints provided. Table 3.6 demonstrates the performance of SS-NMF when varying amounts of pairwise constraints are available *a priori*. We reported the results in terms of the confusion matrix $\mathbf{C}$ and the cluster centroid matrix $\mathbf{S}$. As the available prior knowledge increases from 0% to 5%, we can make the following two key observations. Firstly, the confusion matrices tend to become per-

Table 3.5: Comparison of document clustering accuracy between k-means, kernel k-means, spectral normalized cuts (SNC), NMF and, SS-NMF with 3% constraints.

Dataset	*Graft-Phos*	*England-Heart*	*Interest-Trade*	*Fbis2*	*Fbis3*	*Fbis4*	*Fbis5*	*Fbis10*
k-means	*0.6849*	*0.7108*	*0.7228*	*0.5650*	*0.4728*	*0.4620*	*0.4180*	*0.2320*
kernel k-means	*0.7986*	*0.7147*	*0.7420*	*0.5700*	*0.5533*	*0.5525*	*0.5140*	*0.3780*
SNC	*0.6553*	*0.6320*	*0.7032*	*0.9900*	*0.6367*	*0.5975*	*0.5420*	*0.3920*
NMF	*0.8157*	*0.7840*	*0.9566*	*0.9950*	*0.6533*	*0.6125*	*0.5900*	*0.4160*
SS-NMF	*0.9932*	*0.9973*	*1.0000*	*1.0000*	*0.8833*	*0.8775*	*0.7520*	*0.6740*



(a) (b) (c)

Figure 3.4: (a) Typical document-document matrix (shown here *England-Heart*) before clustering. (b) *England-Heart* similarity matrix after clustering with SS-NMF. (c) *Fbis5* similarity matrix after clustering with SS-NMF.

fectly diagonal indicating higher clustering accuracy. Second observation pertains to the cluster centroid matrix $\mathbf{S}$ which represents the similarity or distance between the clusters. Increasing values of the diagonal elements of $\mathbf{S}$ indicate higher inter-cluster similarities. As expected, when the amount of prior knowledge available is more, the performance of the algorithm clearly gets better.

In Figure 3.4a, the sparsity pattern of a typical document-document matrix $\mathbf{A} = \mathbf{X}^T\mathbf{X}$ (*England-Heart* in the figure) before clustering is shown. The SS-NMF algorithm is applied to the modified similarity matrix $\tilde{\mathbf{A}}$. Document clustering leads to re-ordering of the rows and columns of the matrix. Figures 3.4b and c, show the $\tilde{\mathbf{A}}$ matrices for *England-Heart* and *Fbis5* data sets after clustering with 5% pairwise constraints. Document clusters are indicated by the dense sub-matrices in these matrices.

Table 3.6: The comparison of confusion matrix **C** and cluster centroid matrix **S** of SS-NMF for different percentages of document pairs constrained.

% of const.	*Comp. matrix*	*Graft-Phos dataset*		*England-Heart dataset*		*Interest-Trade dataset*		*Fbis5 dataset*				
0%	**C**	116	21	181	81	215	15	1	1	4	1	4
		33	123	0	113	4	204	84	95	0	0	1
								14	1	11	1	0
								0	0	0	96	3
								1	3	85	2	92
	S	0.7771	0	1.0364	0	2.2788	0	1.0695	0	0	0	0
		0	0.7733	0	1.1500	0	2.0855	0	0.8690	0	0	0
								0	0	1.0392	0	0
								0	0	0	0.87	0
								0	0	0	0	1.0416
1%	**C**	130	3	181	31	216	1	92	17	0	8	0
		19	141	0	163	3	218	0	0	22	0	0
								0	0	64	0	1
								0	0	1	89	0
								8	83	13	3	99
	S	0.9143	0	1.2164	0	2.6920	0	2.5203	0	0	0	0
		0	0.9442	0	1.5346	0	2.4075	0	2.4751	0	0	0
								0	0	2.4251	0	0
								0	0	0	2.6532	0
								0	0	0	0	2.8233
3%	**C**	147	0	193	0	219	0	55	0	0	7	0
		2	144	1	181	0	219	33	99	0	0	0
								0	0	0	0	0
								0	0	90	89	0
								72	1	10	4	100
	S	1.2317	0	2.5813	0	3.3250	0	4.2578	0	0	0	0
		0	1.3005	0	2.7989	0	3.7290	0	4.6787	0	0	0
								0	0	4.2349	0	0
								0	0	0	4.0898	0
								0	0	0	0	4.0951
5%	**C**	149	0	194	0	219	0	100	0	0	0	0
		0	144	0	181	0	219	0	100	0	0	0
								0	0	100	0	0
								0	0	0	100	0
								0	0	0	0	100
	S	1.6094	0	3.4279	0	4.1829	0	6.5171	0	0	0	0
		0	1.5981	0	2.5649	0	4.5167	0	6.3111	0	0	0
								0	0	6.0427	0	0
								0	0	0	6.7312	0
								0	0	0	0	5.9222

We also compare SS-NMF with the other two semi-supervised clustering approaches. As before, for SS-KK, a Gaussian kernel was used. In Figure 3.5, we plot the AC values against increasing percentage of pairwise constraints available, for the algorithms on all the data sets. On the whole, all three algorithms perform better as the percentage of pairwise constraints increases. While the performance of SS-KK is close to that of SS-SNC on the data sets in Figures 3.5a-3.5c, it is clearly left out of the race completely in Figures 3.5d-3.5h. This is mainly because of the fact that SS-KK is unable to maintain its accuracy when producing more than 2 clusters. While the performance of SS-SNC is head-to-head with SS-NMF on *Fbis2* and *Fbis3*, it is consistently outperformed by SS-NMF on the rest of the data sets. Another noticeable fact is that the curve for SS-KK and SS-SNC might take a slow rise in some cases indicating that they need more amount of prior knowledge to improve the performance. Comparatively, SS-NMF gets better accuracy than the other two algorithms even for minimum percentage of pairwise constraints.

2. **Gene Expression Clustering**

Second, we present the comparison of SS-NMF with the other algorithms on real-world gene expression data sets. We first compare the four unsupervised clustering approaches with SS-NMF having pairwise constraints on only 3% pairs of all the possible sample pairs. For kernel k-means, we used a Gaussian (exponential) kernel, with variance $\sigma = 0.00001$ for *ALL/AML* and *Colon Tumor* data sets and a polynomial kernel $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1 * \mathbf{x}_2')^p$ with polynomial parameter $p = 1$ for the other gene expression data sets. In Table 3.7, we compare the algorithms on all the five gene expression data sets with AC values. As is the case with document clustering, SS-NMF performs to be the best across all the data sets. It is evident that the algorithm learns quickly in spite of having few constraints. Table 3.8 demonstrates the performance of SS-NMF improves when the number of pairwise constraints on the gene expression data sets increase from 0% to 5%.

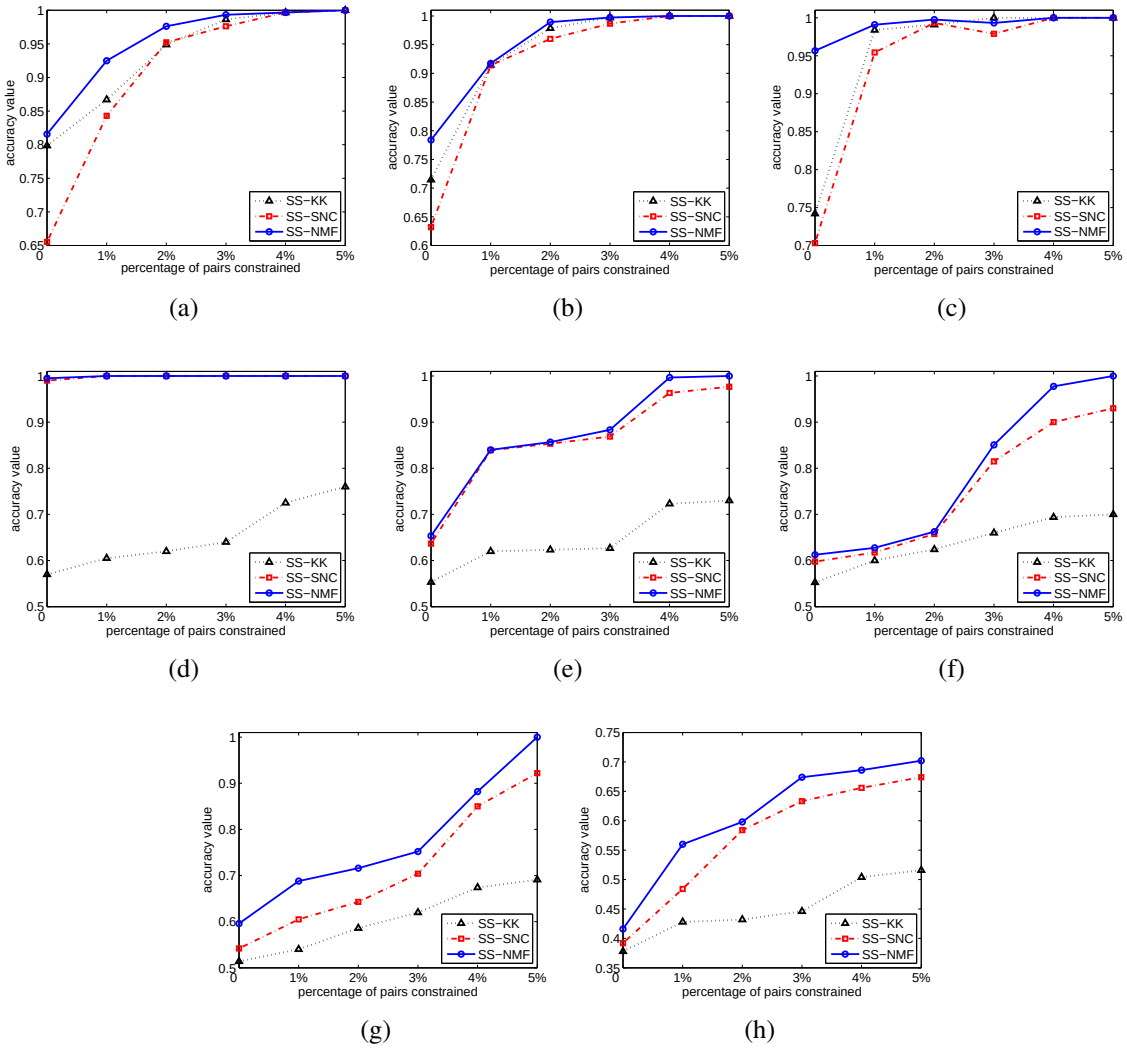


Figure 3.5: Comparison of document clustering accuracy between SS-KK, SS-SNC, and SS-NMF for different percentages of document pairs constrained (a) *Graft-Phos*, (b) *England-Heart*, (c) *Interest-Trade*, (d) *Fbis2*, (e) *Fbis3*, (f) *Fbis4*, (g) *Fbis5*, and (h) *Fbis10* dataset.

These results are reported in terms of the confusion matrix **C** and the normalized cluster centroid matrix **S** as before.

Table 3.7: Comparison of gene expression clustering accuracy between k-means, kernel k-means, spectral normalized cuts (SNC), NMF and, SS-NMF with 3% constraints.

Dataset	*ALL/AML*	*Colon Tumor*	*Prostate Cancer*	*ALL/MLL/AML*	*CNS*
k-means	*0.5263*	*0.6290*	*0.5784*	*0.6316*	*0.5294*
kernel k-means	*0.5263*	*0.5323*	*0.6078*	*0.6491*	*0.6765*
SNC	*0.6316*	*0.5968*	*0.5980*	*0.5601*	*0.6553*
NMF	*0.6842*	*0.6613*	*0.6471*	*0.6667*	*0.7674*
SS-NMF	*0.7632*	*0.7581*	*0.6667*	*0.7368*	*0.8529*

Next, we compare SS-NMF with the other two semi-supervised clustering approaches on the gene expression data sets. Figure 3.6 shows a plot of the AC values against increasing percentage of pairwise constraints for the three semi-supervised algorithms on all the five data sets. All three algorithms perform better as the percentage of pairwise constraints increases. SS-NMF performs significantly better than the other two algorithms with any percentage of constraints when distinguishing between tumor and non-tumor samples, as in Figures 3.6b-3.6c. Also, for clustering subtypes of tumors, although the differences are small, SS-NMF outperforms the other two algorithms as seen from Figures 3.6a, 3.6d-3.6e.

3. **Image Clustering**

We perform comparison of three popular unsupervised image clustering methods: kernel k-means (KK), spectral normalized cuts (SNC), and NMF, with the proposed algorithm. For SS-NMF, we provide only 3% pairwise constraints out of a total of possible image pairs. For KK, we use a polynomial kernel $K = (1 + \mathbf{X}_1^T \mathbf{X}_2)^p$ with $p = 1$. In Table 3.9, we compare the algorithms using AC values. Amongst the unsupervised methods, NMF proves to be the best one. However, the accuracy of all the unsupervised methods greatly deteriorates and is unable to produce meaningful results on data sets having more than 2 clusters. This is because these methods only rely on the visual features of the images

Table 3.8: The comparison of confusion matrix **C** and cluster centroid matrix **S** of SS-NMF for different percentages of gene expression sample pairs constrained.

% of constraints	*Comparison matrix*	*ALL/AML dataset*		*ALL/MLL/AML dataset*			*CNS dataset*			
0%	**C**	17	2	17	6	0	7	1	2	0
		10	9	1	6	5	0	7	0	0
				2	5	15	2	0	8	0
							1	2	0	4
	S	1.3679	0	1.3501	0	0	3.7739	0	0	0
		0	1.3063	0	1.0768	0	0	4.6714	0	0
				0	0	1.3498	0	0	2.9214	0
							0	0	0	3.3950
1%	**C**	18	1	17	3	0	9	1	1	1
		9	10	1	8	6	1	7	0	0
				2	6	14	0	0	8	0
							0	2	1	3
	S	1.3735	0	1.3886	0	0	3.8979	0	0	0
		0	1.3145	0	1.0815	0	0	4.6859	0	0
				0	0	1.3506	0	0	2.9348	0
							0	0	0	3.4125
3%	**C**	19	1	16	1	7	9	1	1	1
		8	10	4	13	0	1	8	0	0
				0	3	13	0	0	9	0
							0	1	0	3
	S	1.3824	0	1.3614	0	0	4.0468	0	0	0
		0	1.3277	0	1.1008	0	0	5.0569	0	0
				0	0	1.3575	0	0	3.1888	0
							0	0	0	3.5181
5%	**C**	21	1	15	3	0	10	0	1	0
		6	10	5	14	1	0	9	0	0
				0	0	19	0	0	8	0
							0	1	1	4
	S	1.3917	0	1.3915	0	0	4.7369	0	0	0
		0	1.3331	0	1.122	0	0	5.2554	0	0
				0	0	1.3582	0	0	3.3510	0
							0	0	0	3.6125

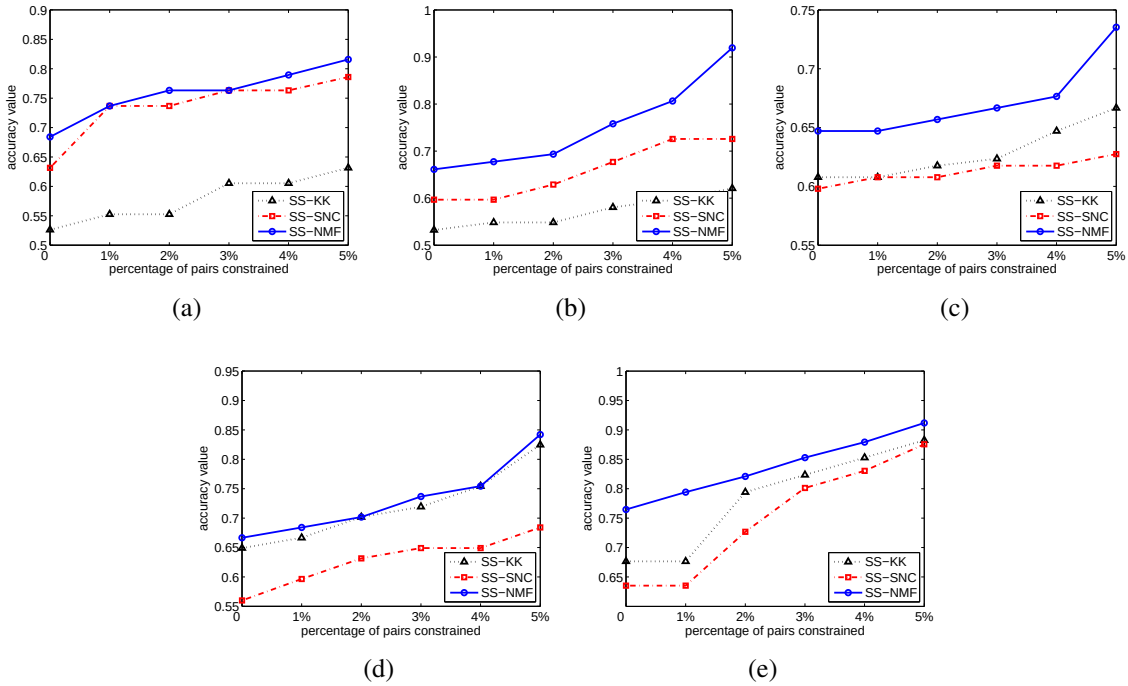


Figure 3.6: Comparison of gene expression clustering accuracy between SS-KK, SS-SNC, and SS-NMF for different percentages of sample pairs constrained (a) *ALL/AML*, (b) *Colon Tumor*, (c) *Prostate Cancer*, (d) *ALL/MLL/AML*, and (e) *CNS* dataset.

to perform the clustering. On the other hand, the superior performance of SS-NMF is evident across all the image data sets. From this experiment, we can infer that in general a semi-supervised method can greatly enhance the image clustering results by benefitting from the user provided knowledge. Moreover, SS-NMF is able to generate significantly better results by quickly learning from the few pairwise constraints provided. Table 3.10 demonstrates the performance of SS-NMF when varying amounts of pairwise constraints are available *a priori*. The results in terms of the confusion matrix **C** and the cluster centroid matrix **S** are reported as before. It is clear to see that the performance of the SS-NMF algorithm clearly gets better as the available prior knowledge increases from 0% to 5%.

Table 3.9: Comparison of image clustering accuracy between KK, SNC, NMF and, SS-NMF with only 3% pairwise constraints.

Dataset	*O-R*	*L-H*	*R-L*	*O-R-L*	*O-R-L-E*	*O-L-E-H*
KK	*0.6933*	*0.6553*	*0.8600*	*0.6750*	*0.6012*	*0.5775*
SNC	*0.8300*	*0.7900*	*0.8750*	*0.7092*	*0.6150*	*0.5975*
NMF	*0.8400*	*0.7950*	*0.8950*	*0.7167*	*0.6550*	*0.6525*
SS-NMF	*0.9400*	*0.8500*	*0.9300*	*0.8833*	*0.7125*	*0.7095*

In addition, we compare SS-NMF with the two semi-supervised clustering approaches: SS-KK and SS-SNC. As before, for SS-KK, a polynomial kernel was used. In Figure 3.7, we plot the AC values against increasing percentage of pairwise constraints available for different combinations of the image categories. On the whole, all the three algorithms perform better as the percentage of pairwise constraints increases. SS-KK is unable to achieve decent clustering results and is clearly the weakest of the three algorithms. The performance of SS-SNC is head-to-head on few categories such as in Figures 3.7(d) and (h), however it is consistently outperformed by SS-NMF in the rest of the results. For experiments involving the image categories *Lions*, *Elephants* and *Horses*, the accuracy of SS-NMF improves significantly when the percentage of pairwise constraints available are around 4% to 5%. This is because there is a considerable resemblance in the images

Table 3.10: The comparison of confusion matrix **C** and cluster centroid matrix **S** of SS-NMF for different percentages of image pairs constrained.

% of constraints	Comparison matrix	*O-R data set*		*L-E-H dataset*		
0%	**C**	81	13	48	20	16
		19	87	46	41	3
				6	39	81
	S	45.659	0	42.276	0	0
		0	48.837	0	50.797	0
				0	0	64.711
1%	**C**	84	11	49	20	16
		16	89	45	42	3
				6	38	81
	S	46.029	0	45.455	0	0
		0	48.837	0	55.531	0
				0	0	68.019
3%	**C**	89	1	50	10	16
		11	99	45	60	3
				5	30	81
	S	47.071	0	45.603	0	0
		0	48.931	0	57.330	0
				0	0	68.390
5%	**C**	94	2	56	21	13
		6	98	15	74	5
				29	5	82
	S	48.289	0	48.289	0	0
		0	49.491	0	61.883	0
				0	0	71.650

of these categories, and hence more prior knowledge is required. Another noticeable fact is that the curve for SS-KK and SS-SNC might take a slow rise in some cases indicating that they need more amount of prior knowledge to improve the performance. Comparatively, SS-NMF gets better accuracy than the other two algorithms even for minimum percentage of pairwise constraints.

4. **UCI Data Clustering**

Table 3.11 shows the comparison of SS-NMF with the four unsupervised clustering algorithms on three UCI data sets. As before, for kernel k-means, we use a Gaussian (exponential) kernel with variance $\sigma = 1$ for *Iris* data and polynomial kernel with polynomial parameter $p = 1$ for the other data sets. As can be seen, with just 5% constraints, SS-NMF yields significantly better results than the unsupervised approaches. For instance, on *Soybean* data, SS-NMF improves the accuracy over 25%. Similar trends can also be observed for other two data sets.

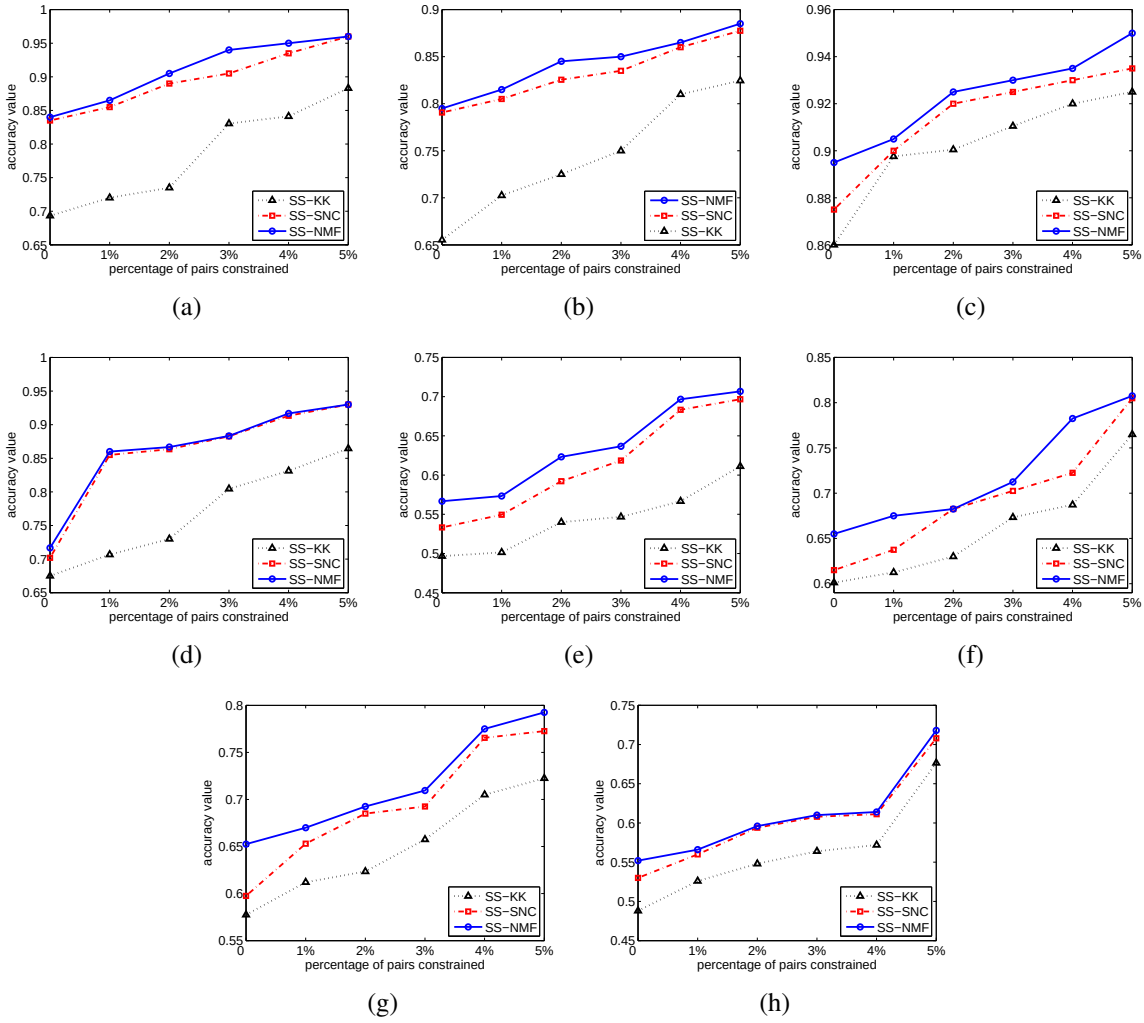


Figure 3.7: Comparison of image clustering accuracy between SS-KK, SS-SNC, and SS-NMF for different percentages of image pairs constrained (a) *O-R*, (b) *L-H*, (c)*R-L*, (d) *O-R-L*, (e) *L-E-H*, (f) *O-R-L-E*, (g)*O-L-E-H* and (h) *O-R-L-E-H* dataset.

Table 3.11: Comparison of UCI data clustering accuracy between k-means, kernel k-means, spectral normalized cuts (SNC), NMF and, SS-NMF with 5% constraints.

Dataset	*Iris*	*LettersIJL*	*Soybean*
k-means	*0.8263*	*0.5167*	*0.7234*
kernel k-means	*0.6933*	*0.5167*	*0.7021*
SNC	*0.6667*	*0.4467*	*0.7234*
NMF	*0.6733*	*0.5200*	*0.7447*
SS-NMF	*0.9267*	*0.6300*	*0.9149*

Figure 3.8 illustrates the performance of SS-NMF and the two semi-supervised algorithms for increasing number of pairwise constraints on the UCI data sets. We can observe that SS-NMF clustering always produces best accuracy performance when the dimensionality of the data sets is high (Figure 3.8b-3.8c). However, it is unable to achieve quality clustering on low dimensionality data sets for fewer constraints. For *Iris* data set which has dimensionality of 4 (Figure 3.8a), SS-NMF yields low accuracy initially and tends to slowly catch up with SS-KK as the percentage of pairwise constraints increase. This shows that SS-NMF is a viable proposition for low-dimensional data as well but needs higher percentage of constraints.

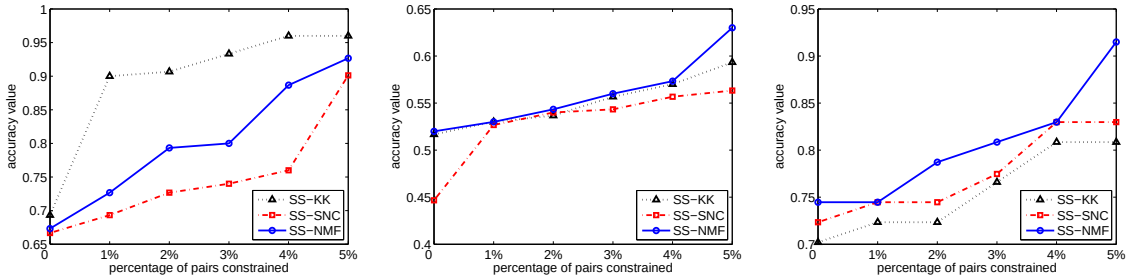


Figure 3.8: Comparison of UCI data clustering accuracy between SS-KK, SS-SNC, and SS-NMF for different percentages of sample pairs constrained (a) *Iris*, (b) *LettersIJL*, and (c) *Soybean* dataset.

3.5 Summary

We present SS-NMF: a semi-supervised approach for clustering based on non-negative matrix factorization. In the proposed framework, users are able to provide supervision in terms of *must-link* and *cannot-link* pairwise constraints on the data objects. We derive an iterative algorithm to perform symmetric tri-factorization of the data similarity matrix. We mathematically show the correctness and convergence of SS-NMF. Moveover, we prove that SS-NMF provides a general and unified framework for semi-supervised data clustering. Existing approaches can be considered as special cases of it. Empirically, we show that SS-NMF out-

performs well-established unsupervised and semi-supervised clustering methods in grouping publicly available datasets.

CHAPTER 4

SEMI-SUPERVISED DATA CO-CLUSTERING BASED ON NMF

In the previous chapter, we show that SS-NMF provides a general and unified framework for semi-supervised clustering. Here, we extend the SS-NMF model to incorporate prior knowledge into heterogeneous data co-clustering. In the proposed SS-NMF co-clustering 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 co-clustering by learning a new distance metric based on these constraints. Using an iterative algorithm, we then perform tri-factorizations 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.

In the following, we first present SS-NMF co-clustering model in Section 4.1 and derive its solution in Section 4.2. Then, we prove the correctness and convergence of the algorithm in Section 4.3.1 and show the relationship between SS-NMF with other well-known co-clustering models in Section 4.3.2 from a theoretical prospective. Empirically, the details of experimental evaluations are given in Section 4.4.

4.1 Model Formulation

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

In our model, given a Star-structured Heterogeneous Relational Data (SHRD) set, with a central data type $\mathcal{X}_c$, and l feature modalities $\mathcal{X}_1$, ..., $\mathcal{X}_p$, ..., $\mathcal{X}_l$, the goal is to cluster central

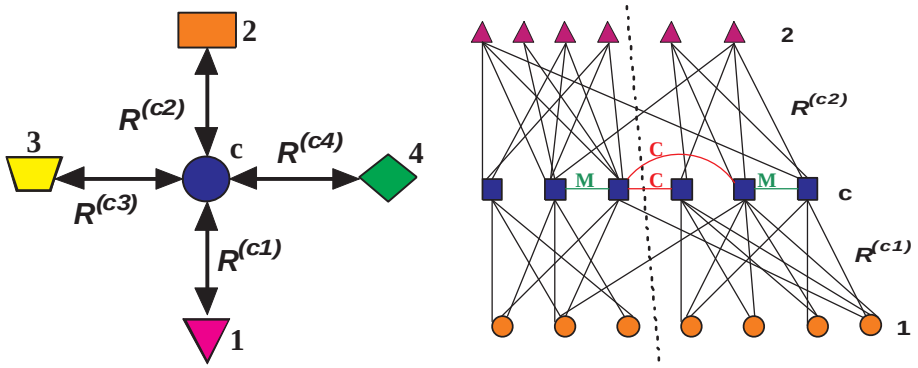
data type $\mathcal{X}_c$ into k_c disjoint clusters simultaneously with feature modality $\mathcal{X}_1$ into k_1 disjoint clusters, ..., $\mathcal{X}_p$ into k_p disjoint clusters, ... , and $\mathcal{X}_l$ into k_l disjoint clusters. Notice that SHRD provides a very good abstraction for many real-world 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, co-clustering 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 co-clustering, 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 co-clustering 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 Figure 4.1(a) 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 co-clustering as an optimization problem with nonnegative tri-factorization of $\mathbf{R}^{(cp)}$,

$$J = \min_{\mathbf{G}^{(c)} \geq 0, \mathbf{G}^{(p)} \geq 0, \mathbf{S}^{(cp)} \geq 0} \sum_{p=1}^{l} \|\mathbf{R}^{(cp)} - \mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\|^2, \tag{4.1}$$

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 semi-supervised co-clustering, we assume that the supervision is provided as two sets of



(a) (b)

Figure 4.1: (a) Heterogeneous star-structured relational data. (b) Star-structured triplet co-clustering with must-link (*M*) and cannot-link (*C*) constraints.

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 co-clustering) than on the feature modalities (e.g., words). Figure 4.1(b) 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 co-clustering result.

4.2 Algorithm Derivation

Let $\mathbf{R}^{(c1)} \in R^{n_c \times n_1}$ denote the relational matrix. The objective of pairwise co-clustering 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 semi-supervised co-clustering, 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 $d(\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)}) =$

$\sqrt{(\mathbf{x}_i^{(c1)} - \mathbf{x}_j^{(c1)})^T \mathbf{L}^{(c1)} (\mathbf{x}_i^{(c1)} - \mathbf{x}_j^{(c1)})}$. Thus, learning the distance metric $\mathbf{L}^{(c1)}$ is equivalent to finding a linear projective mapping $\sqrt{\mathbf{L}^{(c1)}}$ in the feature space [114] 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,

$$\max \quad g(\mathbf{L}^{(c1)}) = \frac{\sum_{(\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)}) \in C} ||\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)}||_{\mathbf{L}^{(c1)}}}{\sum_{(\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)}) \in M} ||\mathbf{x}_i^{(c1)}, \mathbf{x}_j^{(c1)}||_{\mathbf{L}^{(c1)}}}, \tag{4.2}$$

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

$$J = \min \frac{trace(\mathbf{L}^{(c1)} \mathbf{W}_M{}^{(c1)})}{trace(\mathbf{L}^{(c1)} \mathbf{B}_C^{(c1)})}, \tag{4.3}$$

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 Equation (4.3) can be obtained accordingly [114].

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 tri-factorization of the new matrix $\widetilde{\mathbf{R}}^{(c1)}$ as,

$$J = \min_{\mathbf{G}^{(c)} \geq 0, \mathbf{G}^{(1)} \geq 0, \mathbf{S}^{(c1)} \geq 0} \|\widetilde{\mathbf{R}}^{(c1)} - \mathbf{G}^{(c)} \mathbf{S}^{(c1)} \mathbf{G}^{(1)}\|^2. \tag{4.4}$$

The minimization of Equation (4.4) can be done by updating one factor while fixing others [34].

An example of SS-NMF for pairwise co-clustering is illustrated in Figure 4.2. Figure 4.2(a) 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 co-clustering, distance metric learning, is shown in Figure 4.2(c), 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 second step, tri-factorization of $\widetilde{\mathbf{R}}^{(c1)}$, is illustrated in Figure 4.2(d). As a comparison, we also show the result obtained by the unsupervised NMF co-clustering in Figure 2(b). It is clear that the semi-supervised model has a better performance.

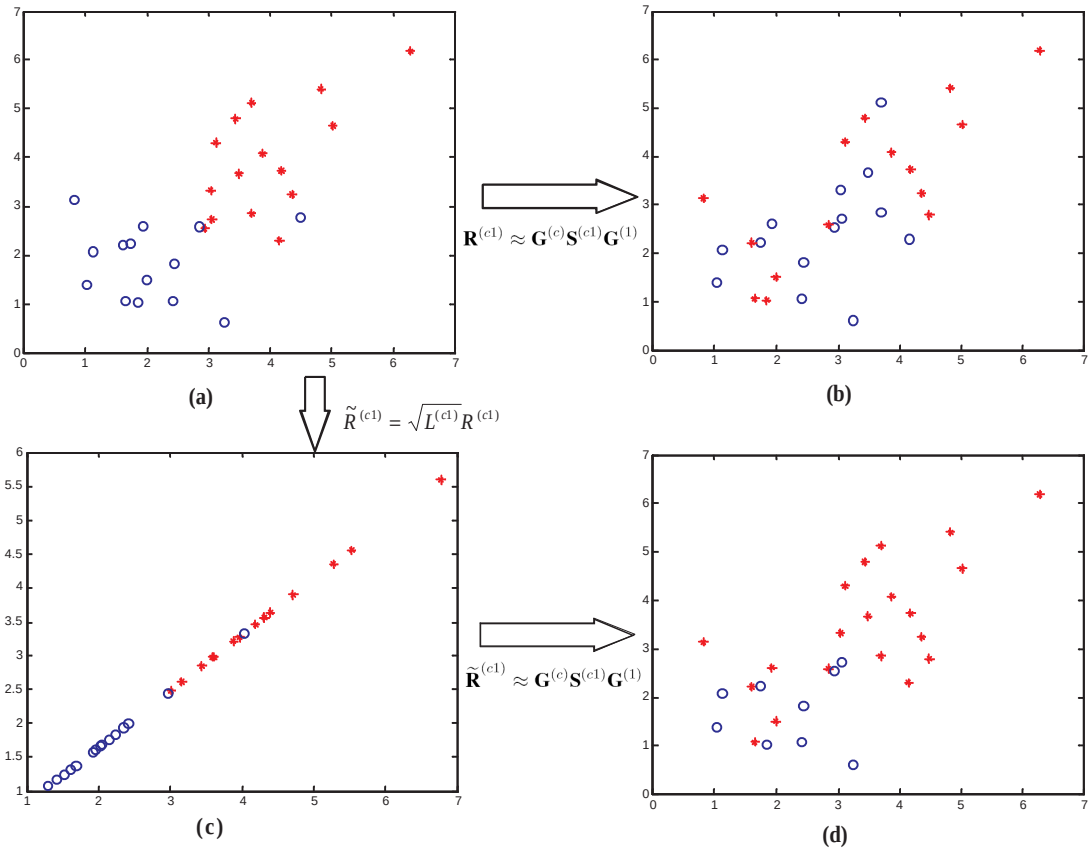


Figure 4.2: An illustration of SS-NMF for data co-clustering: (a) Relational data $\mathbf{R}^{(c1)}$ with two clusters. (b) Clustering result of $\mathbf{R}^{(c1)}$ with unsupervised NMF. (c) New relational data $\widetilde{\mathbf{R}}^{(c1)}$ after a linear projection with distance metric $\mathbf{L}^{(c1)}$. (d) Clustering result of $\widetilde{\mathbf{R}}^{(c1)}$ with SS-NMF.

In general SHRD co-clustering, the central data type has to be clustered together with all feature modalities. Again, let $\mathbf{R}^{(cp)}$ $(1 \leq p \leq l)$ denote a relational matrix between a central

data and each feature modality, the goal of SS-NMF co-clustering 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 co-clustering, the first step in high-order co-clustering 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 co-clustering 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, **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 [13] 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 3, we propose an algorithm to iteratively learn the optimal distance metric $\mathbf{L}^{(cp)}$ and modality importance factor **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 **a** and $\mathbf{L}^{(cp)}$.

To achieve high-order co-clustering, we again need to perform non-negative tri-factorization of $\widetilde{\mathbf{R}}^{(cp)}$ shown in Equation (4.1). 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 4, we derive an EM style approach that iteratively performs the matrix decomposition using a set of multiplicative updating rules.

4.3 Theoretical Analysis

4.3.1 Algorithm Correctness and Convergence

We now prove the theoretical convergence and correctness of the SS-NMF co-clustering algorithm. Motivated by [78, 34], we render the proof based on optimization theory, auxiliary function and several matrix inequalities.

1. **Correctness**

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

   Proposition 6. *If the solution converges based on the updating rules in Equations (4.5)-(4.7), 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,

$$
\begin{aligned}
& L(\mathbf{G}^{(c)}, \mathbf{G}^{(p)}, \mathbf{S}^{(cp)}, \lambda_0, \lambda_p, ..., \lambda_{p+l}) \\
&= \sum_{p=1}^{l} \|\widetilde{\mathbf{R}}^{(cp)} - \mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\|^2 \\
&\quad - \mathrm{Tr}(\lambda_0 \mathbf{G}^{(c)^T}) - \mathrm{Tr}\sum_{p=1}^{l}(\lambda_p \mathbf{S}^{(cp)^T}) - \mathrm{Tr}\sum_{p=1}^{l}(\lambda_{p+l}\mathbf{G}^{(p)^T}).
\end{aligned} \tag{4.8}
$$

   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,

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

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 Equation (4.8), the following three equations are satisfied,

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

Based on the above three equations, we derive the proposed updating rules of Equations (4.5)-(4.7). If the updating rules converge, the solution satisfies the KKT optimality condition. The proof is completed. □

2. **Convergence**

Next, we prove the convergence of the algorithm. In Proposition 7, we show that the objective function decreases monotonically under the three updating rules of Equations (4.5)-(4.7) . This can be done by making use of an auxiliary function similar to that used in [78].

Proposition 7. *If any two of three matrices $\boldsymbol{G}^{(c)}$, $\boldsymbol{S}^{(cp)}$ and $\boldsymbol{G}^{(p)}$ are fixed, $J = \sum_{p=1}^{l} \|\widetilde{\boldsymbol{R}}^{(cp)} - \boldsymbol{G}^{(c)}\boldsymbol{S}^{(cp)}\boldsymbol{G}^{(c)}\|^2$ decreases monotonically under the updating rules of Equations (4.5)-*

(4.7).

Proof. Assume $\mathbf{S}^{(cp)}$ and $\mathbf{G}^{(p)}$ are fixed matrices, a function $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}})$ is called an auxiliary function of $J(\mathbf{G}^{(c)^{[t+1]}})$ if it satisfies $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}) \geq J(\mathbf{G}^{(c)^{[t+1]}})$ and $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t+1]}}) = J(\mathbf{G}^{(c)^{[t+1]}})$ for any $\mathbf{G}^{(c)^{[t+1]}}$ and $\mathbf{G}^{(c)^{[t]}}$. Define $\mathbf{G}^{(c)^{[t+1]}}$ $= arg\min_{\mathbf{G}^{(c)}} F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}})$, then we can construct

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

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

The key step is to find an appropriate auxiliary function $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}})$. Since $\mathbf{G}^{(p)}$ and $\mathbf{S}^{(cp)}$ are fixed, we write

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

and show that

$$\begin{aligned} F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}) = &\sum_{p=1}^{l}\{\| \mathbf{R}^{(cp)} \|^2 \\ &- \sum_{ih} 2(\mathbf{R}^{(cp)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T})_{ih}\mathbf{G}_{ih}^{(c)^{[t]}}(1 + 2log\frac{\mathbf{G}_{ih}^{(c)^{[t+1]}}}{\mathbf{G}_{ih}^{(c)^{[t]}}}) \\ &+ \sum_{ih} \frac{(\mathbf{G}^{(c)^{[t]}}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\mathbf{G}^{(p)^T}\mathbf{S}^{(cp)^T})_{ih}\mathbf{G}_{ih}^{(c)^{4[t+1]}}}{\mathbf{G}_{ih}^{(c)^{3[t]}}}\} \end{aligned} \tag{4.9}$$

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

First, we show that the inequality $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}) \geq J(\mathbf{G}^{(c)^{[t+1]}})$ 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 $\frac{\mathbf{G}_{ih}^{(c)^{[t+1]}}}{\mathbf{G}_{ih}^{(c)^{[t]}}} \geq 1 + 2log(\frac{\mathbf{G}_{ih}^{(c)^{[t+1]}}}{\mathbf{G}_{ih}^{(c)^{[t]}}})$, $\forall \frac{\mathbf{G}_{ih}^{(c)^{[t+1]}}}{\mathbf{G}_{ih}^{(c)^{[t]}}} > 0$. In addition, the third term in $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}})$ is always bigger than the third term in $J(\mathbf{G}^{(c)^{[t+1]}})$ [34]. Thus, the condition $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}}) \geq J(\mathbf{G}^{(c)^{[t+1]}})$ holds. Second, we show the equality $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t+1]}}) = J(\mathbf{G}^{(c)^{[t+1]}})$ holds. It is obvious when $\mathbf{G}^{(c)^{[t]}} = \mathbf{G}^{(c)^{[t+1]}}$, the equality $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t+1]}}) = J(\mathbf{G}^{(c)^{[t+1]}})$ holds.

Therefore, $F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}})$ is an auxiliary function of $J(\mathbf{G}^{(c)^{[t+1]}})$. Since we have $\mathbf{G}^{(c)^{[t+1]}} = arg\min_{\mathbf{G}^{(c)}} F(\mathbf{G}^{(c)^{[t+1]}}, \mathbf{G}^{(c)^{[t]}})$, $\mathbf{G}^{(c)^{[t+1]}}$ 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

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

Thus, we can derive the updating rule of Equation (4.5) as $\mathbf{G}_{ih}^{(c)} \leftarrow \mathbf{G}_{ih}^{(c)} \frac{\sum_{p=1}^{l} (\mathbf{R}^{(cp)} \mathbf{G}^{(p)^T} \mathbf{S}^{(cp)^T})_{ih}}{\sum_{p=1}^{l} (\mathbf{G}^{(c)} \mathbf{S}^{(cp)} \mathbf{G}^{(p)} \mathbf{G}^{(p)^T} \mathbf{S}^{(cp)})_{ih}}$. 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 Equations (4.6) and (4.7). The proof is completed. □

4.3.2 Relationship with Other Data Co-clustering Models

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

model under certain conditions.

Probability based co-clustering

In real world data sets, objects may belong to multiple clusters with varying degrees. Consequently, probability based co-clustering 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 co-clustering, is proposed recently in [86]. 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 co-clustering is to maximize the likelihood as,

$$\begin{aligned} J_{MMRC} &= \max \prod_{i=1}^{n_1} \prod_{j=1}^{n_2} \mathbf{R}_{ij}^{(12)} log p(x_{1i}, x_{2j}) \\ &= \min \prod_{i=1}^{n_1} \prod_{j=1}^{n_2} \mathbf{R}_{ij}^{(12)} log \frac{\mathbf{R}_{ij}^{(12)}}{p(x_{1i}, x_{2j})}, \end{aligned} \tag{4.10}$$

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 co-clustering using the KL-divergence (NMF-KL) as the cost function is to minimize

$$\begin{aligned} J_{NMF-KL} = \min \prod_{i=1}^{n_1} \prod_{j=1}^{n_2} \mathbf{R}_{ij}^{(12)} \\ [log \frac{\mathbf{R}_{ij}^{(12)}}{p(x_{1i}, x_{2j})} - \mathbf{R}_{ij}^{(12)} + (\mathbf{G}^{(1)}\mathbf{S}^{(12)}\mathbf{G}^{(2)})_{ij}]. \end{aligned} \tag{4.11}$$

It can be shown that Equation (4.10) is identical to Equation (4.11), i.e., $J_{MMRC} = -J_{NMF-KL}$ $+constant$, by setting $(\mathbf{G}^{(1)}\mathbf{S}^{(12)}\mathbf{G}^{(2)})_{ij} = p(x_{1i}, x_{2j})$ [32]. 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 co-clustering solution is similar even

though different inference engines are used by the two methods. The relationship between high-order co-clustering using NMF-KL and MMRC can be derived similarly.

Information-theory based co-clustering

The representative algorithms for information-theory based co-clustering include Information-Theoretic for pairwise Co-Clustering (ITCC) [27] and high-order co-clustering [45], Combinatorial MRFs (CMRF) for pairwise co-clustering [7] and high-order co-clustering [6].

ITCC was proposed in [27] 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 $\{x_{11}, ..., x_{1n_1}\}$ and $\{x_{21}, ..., x_{2n_1}\}$, respectively, and $\hat{X}_1$ and $\hat{X}_2$ be the cluster (partition) random variables that take values in the sets $\{\hat{x}_{11}, ..., \hat{x}_{1n_1}\}$ and $\{\hat{x}_{21}, ..., \hat{x}_{2n_2}\}$, 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 tri-factorization, $\mathbf{R}^{(12)}$ 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 multi-information $I(\hat{X}_1; ...; \hat{X}_l)$ is introduced into ITCC (CMRF) as the combinations of several pairwise relations, it can be extended to co-clustering involving more than two random variables. The similarity between high-order NMF and high-order ITCC (CMRF) can

be derived accordingly.

Graph-theory based co-clustering

Some of the well-known graph-theory based co-clustering algorithms include Bipartite Spectral Graph Partitioning (BSGP) [26] for pairwise co-clustering and Spectral Relational Clustering (SRC) [83] for high-order co-clustering.

BSGP was proposed for pairwise data co-clustering in [26]. 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 singular value decomposition (SVD) problem. That is,

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

On the other hand, NMF-based pairwise co-clustering is to minimize the following objective function,

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

The advantage of NMF over BSGP has been discussed in [83].

SRC is proposed in [83] for high-order data co-clustering. 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

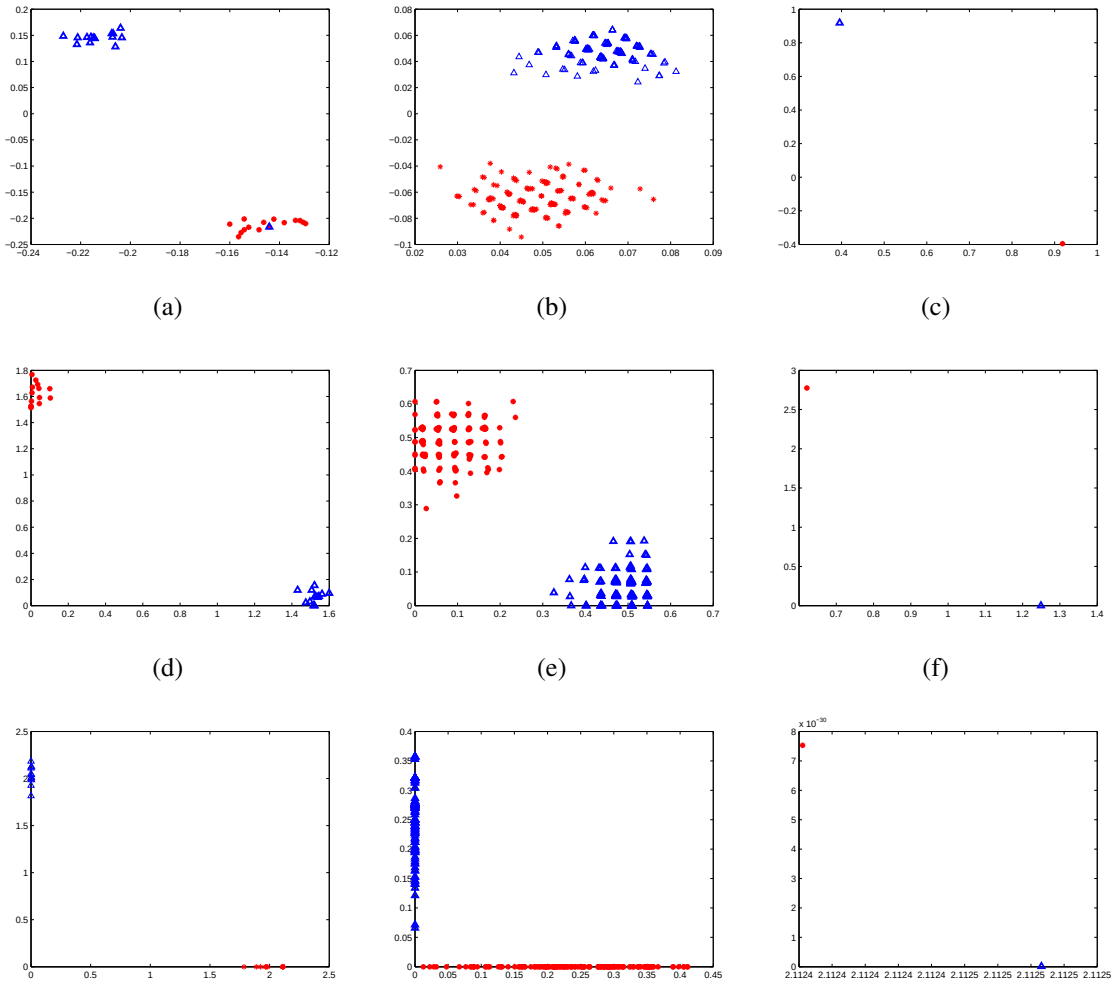
$$\min_{\mathbf{G}^{(c)^T}\mathbf{G}^{(c)}=\mathbf{I},\mathbf{G}^{(p)^T}\mathbf{G}^{(p)}=\mathbf{I}} \sum_{p=1}^{l} \|\mathbf{R}^{(cp)} - \mathbf{G}^{(c)}\mathbf{S}^{(cp)}\mathbf{G}^{(p)}\|^2.$$

On the other hand, NMF-based high-order co-clustering is to minimize the following function,

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

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 2-by-2 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 non-negative latent semantic space which is not required to be orthogonal. Panels (a)-(c), (d)-(f) and (g)-(i) in Figure 4.3 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.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.3: (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% 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.

4.4 Experiments and Results

In this section, we empirically demonstrate the performance of SS-NMF for data clustering. we present the details of our experiments, starting with the descriptions of the data sets (Section 3.4.1), the methodology and evaluation metrics (Section 3.4.2), followed by thorough performance comparisons with leading unsupervised and semi-supervised clustering algorithms (Section 3.4.3).

We first conduct pairwise co-clustering 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 co-clustering with six representative clustering algorithms, including Kernel Kmeans (KK), BSGP, CMRF, NMF, SS-KK, and SS-CMRF. In addition, we also compare our model with a well-known semi-supervised classification method, TSVM. Then, we perform high-order co-clustering 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 co-clustering 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 (semi-supervised) data clustering/classification and show the benefits of integrating prior knowledge into co-clustering.

4.4.1 Data Description and Preprocessing

Text Co-clustering

We primarily utilize the data sets used in [51] [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

[1] http://www.cs.umn.edu/∼han/data/tmdata.tar.gz

data which contains about 2000 articles from 20 newsgroups [77] [2]. In our experiments, we intermix some of the data sets mentioned above. Table 4.1 and Table 4.2 give the details of the data sets we use for pairwise (e.g., document-word) and high-order (e.g., word-document-category) co-clustering, 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 [83, 34], we carry out feature selection to choose the top 1000 words by descending values of the mutual information between a word w and a document label y:

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

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 1000 words based on mutual information. (2) For each document, if any of the top 1000 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 1000 words in this document to 1000. Thus, every element of *document-category* matrix is in the range $[0, 1]$. In addition, for semi-supervised clustering, we define the percentage ($\%$) 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.

[2] http://www.cs.uiuc.edu/homes/dengcai2/Data/TextData.html

Table 4.1: Data sets for text pairwise (document-word) co-clustering.

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 4.2: Data sets for text high-order (word-document-category) co-clustering.

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*

Gene Expression Co-clustering

We utilize seven data sets from Kent Ridge Biomedical Data Repository [3] for gene expression co-clustering, 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 [28]. We sort all genes in a descending order based on the variances and retain only the top 2000 genes. The details of these data sets are given in Table 4.3.

Table 4.3: Data sets for gene expression pairwise (condition-gene) co-clustering.

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*

Image Co-clustering

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 1000 images from ten categories: “eggs”, “decoys”, “firearms”, “cards”, “buses”, “abstract”, “foliage”, “dawn”, “texture” and “wave”. Some examples from each category are shown in Figure 4.4. In our experiment, we mix up some of the aforementioned categories with details in Table 4.4.

For image co-clustering, a large number of visual contents are extracted from each image [109, 87], belonging to two modalities: color and texture. Specifically, color features include color channels (RGB, 9 features, including mean, variance and skewness of R, G, B channels),

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

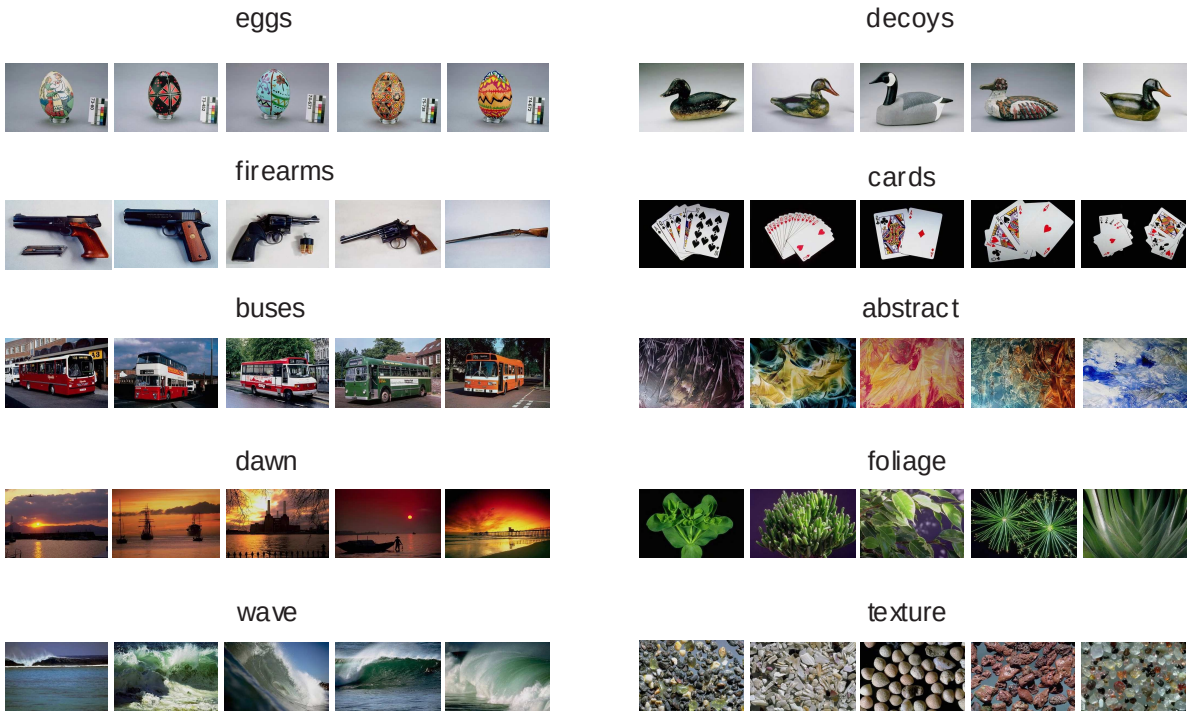


Figure 4.4: Image samples for high-order co-clustering.

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

Table 4.4: Data sets for image high-order (color-image-texture) co-clustering.

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*

4.4.2 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

$$AC = \frac{\sum_{i=1}^{n} \delta(y_i, \hat{y}_i)}{n}, \tag{4.12}$$

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 ten test runs, and the final AC value is the average of all runs.

4.4.3 Pairwise Co-clustering

Text Pairwise Co-clustering

First, we conduct pairwise co-clustering 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 [76], (2) BSGP [26], (3) CMRF [7], (4) NMF (i.e., SS-NMF with 0% constraints), (5) SS-KK [76], and (6) SS-CMRF [7]. The first four are popular unsupervised methods, whereas SS-KK and SS-CMRF are representative semi-supervised ones. Moreover, we also compare with a well-known semi-supervised classification method: TSVM [66].

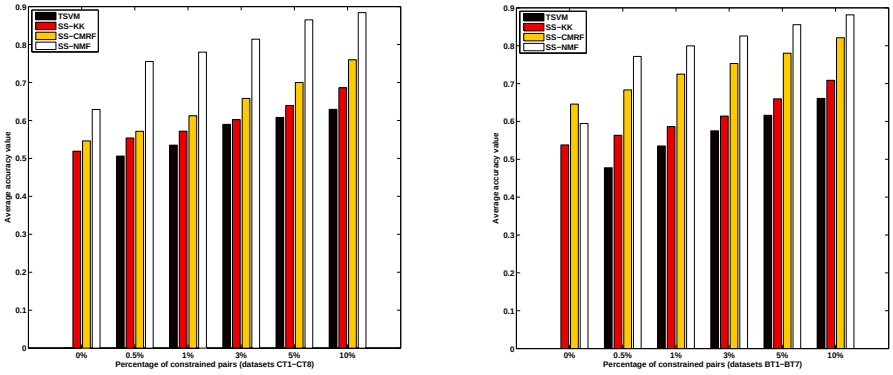
The top half of Table 4.5 shows the AC values of document clustering obtained by unsupervised methods: KK, BSGP, CMRF, and NMF, the semi-supervised classification method: TSVM, and three semi-supervised clustering methods: SS-KK, SS-CMRF and SS-NMF. All of semi-supervised methods are reported based on incorporating 10% 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% lower than NMF, which is the best among the unsupervised methods. However, all unsupervised methods get a low AC value (around 30%) for the data set CT8, which has a large number of clusters ($k = 10$). That is, no meaningful clustering results are produced. Table 4.5 also shows that semi-supervised clustering methods provide at least a 15% increase on the average AC values when compared with the corresponding unsupervised ones. This indicates that a semi-supervised clustering method can generally benefit from additional constraints 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 to CT8. It is also worth noting that the AC values of SS-NMF are as high as 99% 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 semi-supervised clustering approaches outperform TSVM on average AC due to very limited background knowledge (up to 10%). 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.

Table 4.5: Comparison of accuracy among unsupervised clustering KK, BSGP, CMRF, NMF, semi-supervised classification TSVM, and semi-supervised clustering SS-KK, SS-CMRF, SS-NMF with 10% 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*

In Figure 4.5(a), we plot the average AC value on all eight text 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% 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%). Particularly, SS-NMF is able to generate significantly better results (over 10%) by quickly learning from just a few constraints (0.5%). Therefore, document clustering performance can be greatly improved even with very limited prior knowledge.



(a) (b)

Figure 4.5: Comparison of average accuracy for semi-supervised classification TSVM, and pairwise co-clustering SS-KK, SS-CMRF and SS-NMF, with different amounts of constraints on (a) text data, and (b) gene expression data.

Gene Expression Pairwise Co-clustering

Second, we conduct co-clustering on gene expressions with *condition-gene* matrix and compare the performance of SS-NMF with the same set of algorithms used in Section 4.4.3.

The bottom half of Table 4.5 shows the AC values of condition clustering obtained by both unsupervised methods and semi-supervised ones with 10% constraints. Overall, it is evident that SS-NMF provides the best clustering result on average when compared with other unsuper-

vised or semi-supervised methods. As the results demonstrate, the clustering accuracy gain of SS-NMF over unsupervised methods is over 20% 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% on the data sets BT5 and BT6.

Figure 4.5(b) illustrates the average AC values against the increasing percentage of pairwise constraints for semi-supervised condition clustering/classification. Overall, SS-NMF provides the highest accuracy among the four semi-supervised 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% accuracy increase, with very limited prior knowledge (e.g., 0.5% constraints).

4.4.4 High-order Co-clustering

Text High-order Co-clustering

First, we conduct experiments to co-cluster words, documents and categories and compare the performance of SS-NMF with three unsupervised approaches and one semi-supervised method, namely, (1) SRC [83], (2) CMRF [6], (3) NMF (i.e., SS-NMF with 0% constraints), and (4) SS-CMRF (the high-order SS-CMRF is directly extended from the prior work in [6] and [7]).

Co-clustering Accuracy: The top half of Table 4.6 shows document co-clustering accuracy obtained by SRC, CMRF, NMF, SS-CMRF and SS-NMF (both with 15% constraints). Averaged AC values over all nine text 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% constraints are used in high-order co-clustering experiments (recall that we use up to 10% constraints in pairwise co-clustering).

From the top half of Table 4.6, it is obvious that NMF outperforms other unsupervised methods in six out of nine text data sets. In general, SRC performs the worst amongst the three

unsupervised ones. Specifically, its accuracy on the data set HT7 with three categories and six document clusters is only 19%. Also from the top half of Table 4.6, semi-supervised methods provide significantly better results than the corresponding unsupervised ones. The average AC of SS-CMRF increases 15% over CMRF, while up to 20% is gained by SS-NMF over NMF. We also observe that SS-NMF can achieve high clustering accuracy (over 80%) in five out of the nine data sets. The average AC of SS-NMF is 72.43%, about 10% higher than that of SS-CMRF. In Figure 4.6(a), 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.

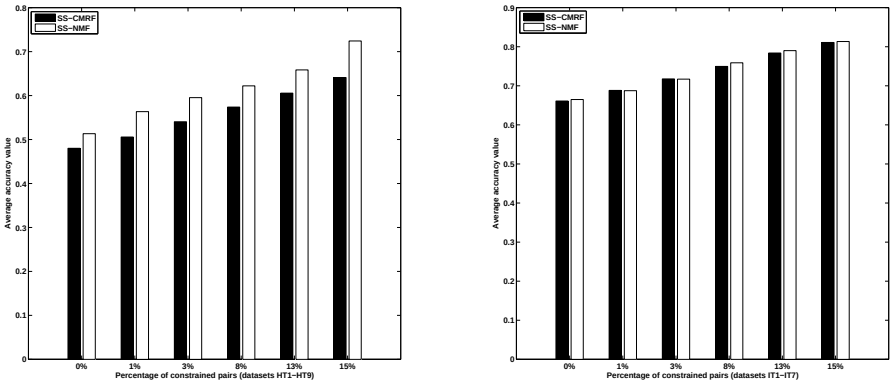
In the left panel of Table 4.7, we report the accuracy of text categorization by SRC, CMRF, NMF, SS-CMRF and SS-NMF. In six out of nine text data sets, the AC value of SS-NMF either ranks the best or the second with exceptions on the data sets: HT3, HT8 and HT9. This result shows that even though the original *document-category* matrix is biased in the distance metric learning towards the constraints on the documents, SS-NMF still can provide a competitive results on category clustering.

In high-order co-clustering, we also obtain the clusters of words simultaneously with the clusters of documents and 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 4.7. These words can be used to represent the underlying "concept" of the corresponding category cluster.

Modality Selection: As described in Section 4.2, distance metric and modality importance are learnt iteratively in Algorithm 3. First, modality selection can provide additional information on the relative importance of various relations (e.g., "word" and "category") for grouping

Table 4.6: Comparison of clustering accuracy between unsupervised SRC, CMRF, NMF, and semi-supervised SS-CMRF, SS-NMF with 15% constraints on text high-order (word-document-category) co-clustering (data sets HT1 - HT9) and image high-order (color-image-texture) co-clustering (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*



(a) (b)

Figure 4.6: Comparison of average clustering accuracy between SS-CMRF and SS-NMF with different amounts of constraints for (a) text high-order co-clustering, and (b) image high-order co-clustering.

Table 4.7: Text categorization: clustering accuracy of categories and Text representation: top ten 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,reserv,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*}

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 4.8 lists the modality importance for the two relations: *document-word* and *document-category* in SS-NMF with 1% 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 co-clustering in the all 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.

Image High-order Co-clustering

Second, we present the experimental results on high-order co-clustering image data.

Co-clustering Accuracy: The bottom half of Table 4.6 lists image clustering accuracy obtained by SRC, CMRF, NMF, SS-CMRF and SS-NMF (both with 15% constraints) for each data set, together with averaged AC value over all seven data sets. Among the three un-

Table 4.8: Modality importance for text high-order co-clustering: word v.s. category and for image high-order co-clustering: color v.s. 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*			

supervised approaches, on average NMF achieves slightly better results. Moreover, both of the semi-supervised methods obtain 20% 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. Figure 4.6(b) 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.

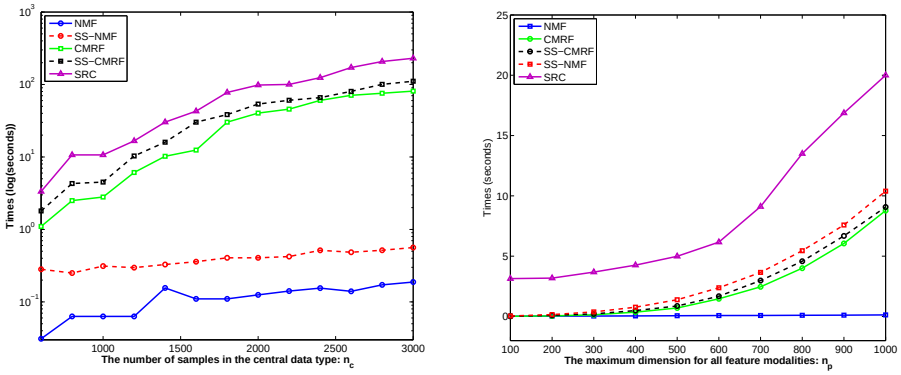
Modality Selection: The semantic gap between the low-level features and the high-level semantic concepts poses great challenge in content-based image retrieval. To this end, modality selection in co-clustering is particularly beneficial because it not only provides the clusters of images, but also shows 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 4.8 lists the weights associated with color and texture given by SS-NMF with 3% 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 judgement, 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 co-clustering.

4.4.5 Time Complexity

Finally, we compare the computational speed of three unsupervised approaches: SRC, CMRF, and NMF, and two semi-supervised 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. Figure 4.7(a) 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 Figure 4.7(b) shows the computational speed with increasing feature dimensions n_p for a fixed n_c. The experiments are performed on a machine with Dual 3GHz Intel Xeon processors and 2GB 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.



(a) (b)

Figure 4.7: Comparison of computational speed between unsupervised approaches (SRC, CMRF, and NMF) and semi-supervised 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.

The difference between SS-NMF and unsupervised NMF 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 Figure 4.7(a), 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 Figure 4.7(b), 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 post-processing (k-means) to infer the clusters. From these results, it is obvious that SS-NMF provides an efficient way for semi-supervised data co-clustering.

4.5 Summary

In this chapter, we present a novel semi-supervised approach for data co-clustering: SS-NMF. In the proposed SS-NMF co-clustering 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. Tri-factorizations 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 co-clustering algorithm and show the relationship between SS-NMF with other data co-clustering 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 co-clustering.

Algorithm 3 Simultaneous Distance Metric Learning and Modality Selection

INPUT: Original relational matrix $\mathbf{R}^{(cp)}(1 \leq p \leq l)$, 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

   $$\mathbf{a}_t^{opt} = arg\min_{\alpha} \|\widetilde{D} - \sum_{p=1}^{l} \alpha^{(cp)} D^{(cp)}\|^2,$$

   (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 $\widetilde{\mathbf{R}}^{(cp)}$.

Algorithm 4 SS-NMF for High-order Co-Clustering

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 $i(1 \leq i \leq n_p)$, $h(1 \leq h \leq k_p)$ and $p(1 \leq p \leq l)$ until *convergence*,
   (a) Cluster indicator matrices:

$$\mathbf{G}^{(c)}_{ih} \leftarrow \mathbf{G}^{(c)}_{ih} \frac{\sum_{p=1}^{l} (\widetilde{\mathbf{R}}^{(cp)} \mathbf{G}^{(p)^T} \mathbf{S}^{(cp)^T})_{ih}}{\sum_{p=1}^{l} (\mathbf{G}^{(c)} \mathbf{S}^{(cp)} \mathbf{G}^{(p)} \mathbf{G}^{(p)^T} \mathbf{S}^{(cp)^T})_{ih}}, \tag{4.5}$$

$$\mathbf{G}^{(p)}_{ih} \leftarrow \mathbf{G}^{(p)}_{ih} \frac{(\mathbf{S}^{(cp)^T} \mathbf{G}^{(c)^T} \widetilde{\mathbf{R}}^{(cp)})_{ih}}{(\mathbf{S}^{(cp)^T} \mathbf{G}^{(c)^T} \mathbf{G}^{(c)} \mathbf{S}^{(cp)} \mathbf{G}^{(p)})_{ih}}. \tag{4.6}$$

   (b) Cluster association matrix:

$$\mathbf{S}^{(cp)}_{ih} \leftarrow \mathbf{S}^{(cp)}_{ih} \frac{(\mathbf{G}^{(c)^T} \widetilde{\mathbf{R}}^{(cp)} \mathbf{G}^{(p)^T})_{ih}}{(\mathbf{G}^{(c)^T} \mathbf{G}^{(c)} \mathbf{S}^{(cp)} \mathbf{G}^{(p)} \mathbf{G}^{(p)^T})_{ih}}. \tag{4.7}$$

CHAPTER 5

EXEMPLAR-BASED VISUALIZATION OF LARGE DATA COLLECTIONS

Visualization enables us to browse intuitively through huge amounts of data and thus could expand the human ability for comprehending complex data sets. In this chapter, we focus on studying one of important applications in data visualization: large text corpus visualization.

With the rapid growth of the World Wide Web and electronic information services, text corpus is becoming available on-line at an incredible rate. No one has time to read everything, yet in many applications we often have to make critical decisions based on our understanding of large document collections. For example, when a physician prescribes a specific drug, he frequently needs to identify and understand a comprehensive body of published literature describing an association between the drug of interest and an adverse event of interest. Thus, text mining, a technique of deriving high-quality knowledge from text, has recently drawn great attention in the research community. Research topics in text mining include, but not limited to, language identification, document clustering, summarization, text indexing and visualization. In particular, text visualization refers to the technology that displays text data or mining results in a logical layout (e.g., color graphs) so that one can view and analyze documents easily and intuitively. It presents a direct way to observe the documents as well as understand the relationship between them. In addition, text visualization allows people to explore the inside logic of the model and offers users a chance to interact with the text mining model so that questions can be answered.

In general, it is convenient to transform document collections into a data matrix [24], where the columns represent documents and the row vectors denote keyword counting after pre-processing. Thus, text data sets have a very high dimensionality. A common way of visu-

alizing text corpus is to map the raw data matrix into a d-dimensional space with $d = 1, 2, 3$ by employing dimensionality reduction techniques. The objective is to preserve in the projected space the distance relationships among the documents in their original space. Depending on the choice of mapping functions, both linear (e.g., principle component analysis (PCA) [67]) and nonlinear (e.g., ISOMAP [107]) dimensionality reduction techniques have been proposed in the literature. Facing the ever-increasing amount of available documents, a major challenge of text visualization is to develop scalable approaches that are able to process tens of thousands of documents. First, from a computational point of view, large text corpus significantly raises the bar on the efficiency of an algorithm. For a collection of more than ten thousand documents, typical data projection methods, such as PCA, will fail to run due to insufficient memory. Second, since all documents are shown at once in the resulting space, overlaps of highly related documents are inevitable. Hierarchical clustering-based methods [43, 95] can partially solve the memory problem and produce a tree structure for document exploration. However, these algorithms run extremely slow. More important, they are not mathematically rigorous due to lacking a well defined objective function. Finally, knowledge or information is usually sparsely encoded in document collections. Thus, latent sematic structures (i.e., main topics of a text corpus) are included into the projection techniques for text visualization, such as Probabilistic Latent Semantic Analysis (PLSA) [54] and Least Square Projection (LSP)[96]. Generally, these models can provide a higher quality (i.e., more meaningful) visualization.

Therefore, we propose an Exemplar-based approach to Visualize (EV) extremely large text corpus. Capitalizing on recent advances in matrix approximation and decomposition, our method provides a means to visualize tens of thousands of documents with high accuracy (in retaining neighbor relations), high efficiency (in computation), and high flexibility (through the use of exemplars). Specifically, we first computes a representative text data subspace $\mathbf{C}$ and a low-rank approximation $\tilde{\mathbf{X}}$ by applying the low-rank matrix approximation method. Next, documents are clustered through the matrix decomposition: $\tilde{\mathbf{X}} = \mathbf{CWG}^T$, where $\mathbf{W}$ is the

weight matrix, and $\mathbf{G}$ is the cluster indicator matrix. To reduce the clutter in the visualization, the exemplars in each cluster are first visualized through Parameter Embedding (PE) [60], providing an overview of the distribution of the entire document collection. When desired, on the clicking of an exemplar, documents in the associated cluster or in a user-selected neighborhood are shown to provide further details. In addition, hierarchical data exploration can also be implemented by recursively applying EV in an area of interest.

In the following, we first present the EV model and derive the algorithm in Section 5.1. Then, we give some theoretical results in Section 5.2, including the correctness and convergence of the algorithm, time and space complexity analysis, and advantages of EV when compared with other visualization models. Finally, in Section 5.3, we provide thorough experimental evaluation.

5.1 Model Formulation and Algorithm

The proposed EV model takes a three-step approach to visualize large-scale text corpus. First, low rank matrix approximation is employed to select the representative subspaces and generate the compact approximation of the word-document matrix $\mathbf{X} \in R^{d_1 \times n}$. Among various matrix approximation methods, near-optimal low-rank approximation has gained increasing popularity in recent years due to its great computational and storage efficiency. The representative ones include Algorithm 844 [9], CUR [97] and CMD [105]. Typically, a near-optimal low-rank approximation algorithm first selects a set of columns $\mathbf{C}$ and a set of rows $\mathbf{R}$ as the left and right matrices of the approximation. Then, the middle matrix $\mathbf{U}$ is computed by minimizing $\|\mathbf{X} - \mathbf{CUR}\|_F^2$. Thus, at the end of the first step, we obtain the low-rank approximation $\tilde{\mathbf{X}} = \mathbf{CUR}$, the representative subspaces $\mathbf{C}$ (data exemplar set) and $\mathbf{R}$ (feature set).

In the second step, we use matrix factorization to obtain the "soft" cluster indicators in the low-rank exemplar subspace, representing the probability of each document proportion to the

topics in the topic model [33]. We formulate this task as an optimization problem,

$$\begin{aligned} J &= \min_{\mathbf{W}\geq 0, \mathbf{G}\geq 0} \|\tilde{\mathbf{X}} - \mathbf{C}\mathbf{W}\mathbf{G}^T\|_F^2 \\ &= \text{Tr}(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} - \tilde{\mathbf{X}}^T\mathbf{C}\mathbf{W}\mathbf{G}^T - \mathbf{G}\mathbf{W}^T\mathbf{C}^T\tilde{\mathbf{X}} + \mathbf{G}\mathbf{W}^T\mathbf{C}^T\mathbf{C}\mathbf{W}\mathbf{G}^T), \end{aligned} \tag{5.1}$$

where $\mathbf{W}$ is the weight matrix and $\mathbf{G}$ is the cluster indicator matrix with each element $g_{ih} \in [0, 1]$, indicating the probability distribution over topics for a particular document. In the optimization process, we propose an iterative algorithm to get non-negative $\mathbf{W}$ and $\mathbf{G}$ while fixing arbitrarily signed $\mathbf{C}$ and $\tilde{\mathbf{X}}$. The updating rules are obtained by using the auxiliary functions and the optimization theory as,

$$\mathbf{W}_{ih} \leftarrow \mathbf{W}_{ih}\sqrt{\frac{(\mathbf{A}_1{}^+\mathbf{G})_{ih} + (\mathbf{A}_3{}^-\mathbf{W}\mathbf{G}^T\mathbf{G})_{ih}}{(\mathbf{A}_1{}^-\mathbf{G})_{ih} + (\mathbf{A}_3{}^+\mathbf{W}\mathbf{G}^T\mathbf{G})_{ih}}}, \tag{5.2}$$

$$\mathbf{G}_{ih} \leftarrow \mathbf{G}_{ih}\sqrt{\frac{(\mathbf{A}_2{}^+\mathbf{W})_{ih} + (\mathbf{G}\mathbf{W}^T\mathbf{A}_3{}^-\mathbf{W})_{ih}}{(\mathbf{A}_2{}^-\mathbf{W})_{ih} + (\mathbf{G}\mathbf{W}^T\mathbf{A}_3{}^+\mathbf{W})_{ih}}}, \tag{5.3}$$

where $\mathbf{A}_1 = \mathbf{C}^T\tilde{\mathbf{X}}$, $\mathbf{A}_2 = \tilde{\mathbf{X}}^T\mathbf{C}$ and $\mathbf{A}_3 = \mathbf{C}^T\mathbf{C}$.

The third step is to use PE [60] to embed documents into a low-dimensional Euclidean space such that the input probabilities $\mathbf{G} = p(L_h|\mathbf{x}_i)$ (where L is the topic label of a document) are approximated as closely as possible by the embedding-space probabilities $p(L_h|\mathbf{y}_i)$. The objective is to minimize the difference between input probabilities and the corresponding embedding-space probabilities using a sum of Kullback-Leibler (KL) divergences for each document: $\sum_{i=1}^{n} KL(p(L_h|\mathbf{x}_i)\|p(L_h|\mathbf{y}_i))$. Minimizing this sum $\sum_{h=1}^{z} p(L_h|\mathbf{y}_i)$ is equivalent to minimizing the following sum of KL divergences,

$$E(\mathbf{y}_i, \phi_h) = -\sum_{i=1}^{n}\sum_{h=1}^{z} p(L_h|\mathbf{x}_i)\log p(L_h|\mathbf{y}_i). \tag{5.4}$$

The unknown parameters, a set of coordinates of documents $\mathbf{y}_i$ and coordinates of topics ϕ_h in

the embedding space, can be obtained with a gradient-based numerical optimization method. The gradients of Equation (5.4) with respect to $\mathbf{y}_i$ and ϕ_h are

$$\frac{\partial E}{\partial \mathbf{y}_i} = \sum_{h=1}^{z} (p(L_h|\mathbf{x}_i) - p(L_h|\mathbf{y}_i))(\mathbf{y}_i - \phi_h), \tag{5.5}$$

$$\frac{\partial E}{\partial \phi_h} = \sum_{i=1}^{n} (p(L_h|\mathbf{x}_i) - p(L_h|\mathbf{y}_i))(\phi_h - \mathbf{y}_i). \tag{5.6}$$

Thus, we can find the locally optimal solution for embedding coordinates $\mathbf{y}_i$ for each document given ϕ_h.

The complete EV algorithm is given in Algorithm 5.

5.2 Theoretical Analysis

In this section, we first show that our algorithm is correct and converges under the updating rules given in Equations (5.2)-(5.3). In addition, we show the efficiency of EV by analyzing its space and time requirements. Finally, we point out the advantages of EV when compared with other visualization methods.

5.2.1 Correctness and Convergence of EV

The correctness and convergence of the EV algorithm can be stated as the following two propositions.

Proposition 8 (Correctness of EV)**.** *Given the object function of Equation (5.1), the constrained solution satisfies KKT complementary conditions under the updating rules in Equations (5.2)- (5.3).*

Proposition 9 (Convergence of EV)**.** *The object function of Equation (5.1) is monotonically decreasing under the updating rules in Equations (5.2)- (5.3).*

We give an outline of the proof of the propositions and omit the details since the proof is as similar as in 4.3. First, following the standard theory of constrained optimization, we fix

Algorithm 5 Exemplar-based Visualization

INPUT: word-document matrix $\mathbf{X} \in R^{d_1 \times n}$, selected number of documents and words $r, c \in \mathbb{Z}^+$ s.t.$1 \leq r \leq d_1$, $1 \leq c \leq n$, number of topics $z \in \mathbb{Z}^+$ s.t. $1 \leq h \leq z$, and the label set of topics $L_{h=1}^{z}$

OUTPUT: Visualization of documents $\mathbf{Y} = \{\mathbf{y}_i\} \in R^{d_2 \times n}$ $(1 \leq i \leq n,\ d_2 < d_1)$ in the embedding space

1. Use a near-optimal low-rank approximation method to get $\mathbf{C} \in R^{d_1 \times c}$, $\mathbf{U} \in R^{c \times r}$, $\mathbf{R} \in R^{r \times n}$ and $\tilde{\mathbf{X}} \in R^{d_1 \times n}$,

2. Initialize $\mathbf{W}$ and $\mathbf{G}$ with non-negative values,

3. Iterate by the following updating rules for each i and h until *convergence*,

   (a) Let $\mathbf{A}_1 = \mathbf{C}^T\tilde{\mathbf{X}}$, $\mathbf{A}_2 = \tilde{\mathbf{X}}^T\mathbf{C}$ and $\mathbf{A}_3 = \mathbf{C}^T\mathbf{C}$, then split each matrix into the positive and negative parts:

   $$\mathbf{A}_q^+ = (|\mathbf{A}_q| + \mathbf{A}_q)/2, \qquad \mathbf{A}_q^- = (|\mathbf{A}_q| - \mathbf{A}_q)/2,$$

   where $q \in \{1, 2, 3\}$,

   (b) Weight matrix and cluster indicator matrix:

   $$\mathbf{W}_{ih} \leftarrow \mathbf{W}_{ih}\sqrt{\frac{(\mathbf{A}_1{}^+\mathbf{G})_{ih} + (\mathbf{A}_3{}^-\mathbf{W}\mathbf{G}^T\mathbf{G})_{ih}}{(\mathbf{A}_1{}^-\mathbf{G})_{ih} + (\mathbf{A}_3{}^+\mathbf{W}\mathbf{G}^T\mathbf{G})_{ih}}},$$

   $$\mathbf{G}_{ih} \leftarrow \mathbf{G}_{ih}\sqrt{\frac{(\mathbf{A}_2{}^+\mathbf{W})_{ih} + (\mathbf{G}\mathbf{W}^T\mathbf{A}_3{}^-\mathbf{W})_{ih}}{(\mathbf{A}_2{}^-\mathbf{W})_{ih} + (\mathbf{G}\mathbf{W}^T\mathbf{A}_3{}^+\mathbf{W})_{ih}}},$$

4. Normalize cluster indicator $\mathbf{G} = p(L_h|\mathbf{x}_i)$ such that $\sum_{h=1}^{z} p(L_h|\mathbf{x}_i) = 1$,

5. Use parameter embedding to obtain the embedding-space coordinates $\mathbf{y}_i$ for each document.

one variable $\mathbf{G}$ and introduce the Lagrangian multipliers λ_1 and λ_2 to minimize the Lagrangian function $L(\mathbf{W}, \mathbf{G}, \lambda_1, \lambda_2) = \|\tilde{\mathbf{X}} - \mathbf{CWG}^T\|_F^2 - \text{Tr}(\lambda_1 \mathbf{W}) - \text{Tr}(\lambda_2 \mathbf{G}^T)$. Second, based on the KKT complementarity condition, we set the gradient descent of $\frac{\partial L}{\partial \mathbf{W}}$ to be zero while fixing $\mathbf{G}$. Then, we successively update $\mathbf{W}$ using Equation (5.2) until J converges to a local minima. Similarly, given $\mathbf{W}$, we can set $\frac{\partial L}{\partial \mathbf{G}}$ to be zero and update $\mathbf{G}$ using Equation (5.3) until J converges to a local minima. $\mathbf{W}$ and $\mathbf{G}$ should update alternatively. Third, we construct auxiliary functions to prove that Equation (5.1) decreases monotonically under the updating rules. An auxiliary function $Z(\mathbf{W}^{t+1}, \mathbf{W}^t)$ should satisfy the two conditions: $Z(\mathbf{W}^{t+1}, \mathbf{W}^t) \geq J(\mathbf{W}^t)$, and $Z(\mathbf{W}^t, \mathbf{W}^t) = J(\mathbf{W}^t)$ for any $\mathbf{W}^{t+1}$ and $\mathbf{W}^t$. We define $\mathbf{W}^{t+1} = \min_{\mathbf{W}} Z(\mathbf{W}, \mathbf{W}^t)$, then we obtain the following equation $J(\mathbf{W}^t) = Z(\mathbf{W}^t, \mathbf{W}^t) \geq Z(\mathbf{W}^{t+1}, \mathbf{W}^t) \geq J(\mathbf{W}^{t+1})$. Thus, with a proper auxiliary function, $J(\mathbf{W}^t)$ is decreasing monotonically. Similarly, we can also prove $J(\mathbf{G}^t)$ is decreasing monotonically under an appropriate auxiliary function.

5.2.2 Time and Space Complexity

To visualize a large data set, efficiency in both space and speed is essential. In the following, we provide detailed analysis on the time and space complexity of EV. To simplify the analysis, we assume $n = d_1$ and $r = c$ though they are not necessarily equal in the algorithm.

In Algorithm 5, the near-optimal matrix approximation is very efficient, having time complexity of $\mathcal{O}(nc^2)$ given in [9]. In the decomposition step, even though $\tilde{\mathbf{X}}$ is used in the description of the algorithm, the computation is actually done using the three small matrices, $\mathbf{C}$, $\mathbf{U}$ and $\mathbf{R}$. Specifically, we first need to compute $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ with the following time,

$$\mathbf{A}_1 : c(n \times c + c^2 + c \times n),$$
$$\mathbf{A}_2 : c(n \times c + c^2 + c \times n),$$
$$\mathbf{A}_3 : c^2 n.$$

Then, we need to compute $\mathbf{W}$ and $\mathbf{G}$ in Equations (5.2) and (5.3). Assuming that the number

of iteration $t = 1$, the time for computing **W** and **G** are

$$\mathbf{W} : 2(c^2z + cz^2 + z^2n + cnz),$$
$$\mathbf{G} : 2(c^2z + cz^2 + z^2n + cnz).$$

Thus, the total time for matrix decomposition is $\mathcal{O}(c^2n + (c^2z + z^2n + cnz))$. In addition, the time complexity of PE is $\mathcal{O}(nz)$. Since $z \ll min(c, n)$ and $c \ll n$, the overall computational complexity is $\mathcal{O}(n)$.

Regarding the space complexity, EV needs $2cn + c^2$ units to store **C**, **U** and **R**, and needs cz and nz units for **W** and **G**, respectively. In addition, the temporal storage for computing $\mathbf{A}_q$ and updating **W** and **G** require $\mathcal{O}(cn)$ units. Since $c \ll n$, the total space used is $\mathcal{O}(n)$.

In summary, both the time and space complexity of EV are linear, and thus it is highly scalable and suitable for visualizing a very large document collection.

5.2.3 Advantages of EV

From a theoretical point of view, EV has the following unique properties for visualizing large-scale text corpus when compared with other visualization methods:

- Accuracy: EV is a probabilistic multidimensional projection model with a well-defined objective function. Through iterative optimization, it can preserve the proximity in the high-dimensional input space and thus provide accurate visualization results.

- Efficiency: EV has a high computational and spacial efficiency, and thus it is especially useful to visualize large document data. Compared with the time complexity of other visualization approaches, EV has a linear running time. Moreover, EV only needs to compute the non-zero entries of the approximation matrix, which further reduces the computational time for a sparse matrix (e.g., word-document matrix). EV also has the space complexity of $\mathcal{O}(n)$ while other algorithms typically require $\mathcal{O}(n^2)$ storage units.

- Flexibility: EV decomposes a word-document matrix into three matrices with the representative data subspace **C**, which contains the exemplar documents from the collection. By choosing the subspace dimensions, EV can visualize text corpus with different granularity, effectively reducing the clutter/overlap in the layout and cognitive overload.

5.3 Experiments and Results

In this section, we compare EV with PLSA+PE, LSP, ISOMAP, MDS and PCA for visualizing text data sets. Specifically, we implement two EV models: EV-844 and EV-CUR, in our experiments. In EV-844, Algorithm 844 [9] is used to successively select a column or row at a time with the largest norm from text data, resulting in an unique subspace; while EV-CUR uses CUR [97] to pick the representative samples based on their probability distributions computed by the norms. Note that duplicates may exist in the CUR subspace because the samples with large norms are likely to be selected more than once. In the following, Section 5.3.1 gives the details of the data sets we used. In Section 5.3.2, we discuss the quantitative evaluation methods used to report the experimental results. On several public text data sets (including two large ones with $18,864$ and $15,565$ documents, respectively), we demonstrate the superior visualization results by EV in Section 5.3.3, in which we also compare the computational speed of all the algorithms.

5.3.1 Data Sets

For the experiments on document visualization, we use the *20Newsgroups* data [77] and *10PubMed* data.

20Newsgroups data consists of documents in the 20 Newsgroups corpus. The corpus contains $18,864$ articles categorized into 20 discussion groups [1] with a vocabulary size $26,214$. Note that at its full size the data here is too large to be processed by all the algorithms except EV. In order to make the comparison with existing methods, we construct two subsets

[1] http://www.cs.uiuc.edu/homes/dengcai2/Data/TextData.html

Table 5.1: Summary of data subsets from *20Newsgroups* used in the experiments.

Data Name	Groups Name	No. of Documents per Group	Total Documents
20Newsgroups-I	{comp.sys.ibm.pc.hardware}, {rec.sport.baseball},{sci.med}	100	300
20Newsgroups-II	all 20 groups	50	1000

Table 5.2: Summary of *10Pubmed* data used in the experiments.

	Document Name	No. of Documents
1	Gout	543
2	Chickenpox	732
3	Raynaud Disease	343
4	Jaundice	503
5	Hepatitis A	796
6	Hay Fever	1517
7	Kidney Calculi	1549
8	Age-related Macular Degeneration	3283
9	Migraine	3703
10	Otitis	2596

of *20Newsgroups* through uniform random sampling: *20Newsgroups-I* and *20Newsgroups-II*, shown in Table 5.1.

10PubMed data consists of published abstracts in the MEDLINE database [2] from 2000 to 2008, relating to 10 different diseases. We use "MajorTopic" tag along with the disease-related MeSH terms as queries to MEDLINE. Table 5.2 shows the 10 document sets ($15,565$ documents) retrieved. From all the retrieved abstracts, the common and stop words are removed, and the words are stemmed using Porter's suffix-stripping algorithm [98]. Finally, we build a word-document matrix of the size 22437×15565.

5.3.2 Evaluation Measurement

We evaluate the visualization results quantitatively based on the label predication accuracy with the k-nearest neighbor (k-NN) method [36] in the visualization space. Documents are labeled with discussion groups in the *20Newsgroups* data, and with disease names in the *10PubMed* data. Majority voting among the training documents in the k neighbors of a test document is used to decide its predicted label. The accuracy generally becomes high when

[2] http://www.ncbi.nlm.nih.gov/pubmed/

documents with the same label are located together while documents with different labels are located far away from each other in the visualization space.

Quantitatively, the accuracy $AC(k)$ is computed as,

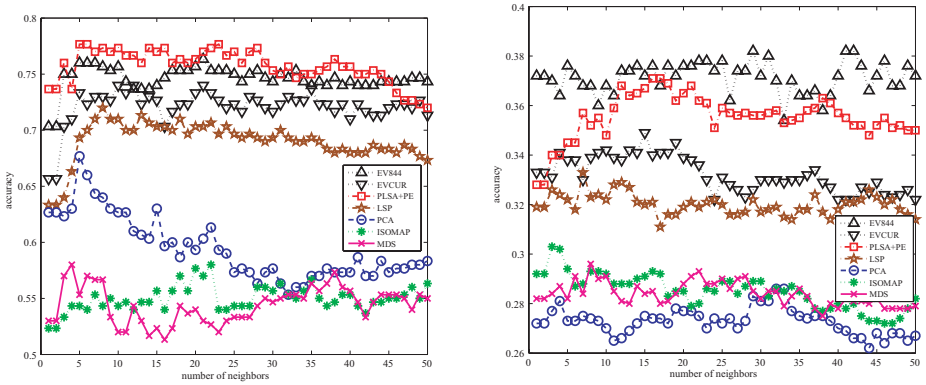
$$AC(k) = \frac{1}{n}\sum_{i=1}^{n} I(l_i, \hat{l_k}(\mathbf{y}_i)), \tag{5.7}$$

where n denotes the total number of documents in the experiment, l_i is the ground truth label of the ith document, $\hat{l_k}(\mathbf{y}_i)$ is the predicted label by k-NN in the embedding space, and I is the delta function that equals one if $\hat{l_k}(\mathbf{y}_i) = l_i$, and zero otherwise.

5.3.3 Results

First, we compare the neighbor-preserving accuracy in two-dimensional visualization generated by EV-844, EV-CUR, PLSA+PE, LSP, ISOMAP, MDS, and PCA on the data sets *20Newsgroups-I* and *20Newsgroups-II*. Through uniform random sampling, we create 10 independent evaluation sets for each data set, with given number of topics (3 for *20Newsgroups-I* and 20 for *20Newsgroups-II*) and documents (100 for *20Newsgroups-I* and 50 for *20Newsgroups-II*). The average accuracy values are obtained using k-NN over the 10 sets with $k = \{1, 2, ..., 50\}$, shown in Figure 5.1.

Generally, the AC values obtained by the seven methods are higher for a small number of topics (e.g., z=3 in Figure 5.1(a)) than those with a large number of topics (e.g., z=20 in Figure 5.1(b)). Moreover, the accuracy achieved by the topic models (i.e., EV-844, EV-CUR, PLSA+PE and LSP) is significantly higher than the traditional projection methods (i.e., PCA, MDS and ISOMAP). These results indicate that topic information is very helpful for the data visualization. When visualizing real-world text corpus, particularly the ones collected from the World Wide Web, the number of topics is typically unknown and thus has to be estimated through topic model detection. Some well-known approaches include Bayesian Inference Criteria (BIC) and Minimum Message Length (MML). A detailed discussion of model detection



(a) (b)

Figure 5.1: Accuracy with k-NN in the two-dimensional visualization space with different k: (a) *20Newsgroups-I* (3 topics), and (b) *20Newsgroups-II* (20 topics).

can be found in [89]. In our experiments, the number of topics for all the topic models is simply set based on the ground truth. Another important observation from Figure 5.1 is that EV-844 constantly provides a higher accuracy value than EV-CUR. This is mainly because Algorithm 844 selects unique columns (exemplars) while CUR may choose replicated ones to build the subspace. Thus, we use EV-844 in the rest of our experiments and refer it to EV without special mention. Finally, as shown in Figure 5.1(a), the two probabilistic topic models (i.e., EV and PLSA+PE) have comparable performance on *20Newsgroups-I*. However, as the number of topics increases, EV clearly outperforms PLSA+PE on *20Newsgroups-II* in Figure 5.1(b). These results imply that EV can appropriately embed documents in a two-dimensional Euclidean space while keeping the essential relationship of the documents, especially for a data set with a large number of topics.

Figures 5.2 and 5.3 show the visualization results obtained by EV, PLSA+PE, LSP, ISOMAP, MDS, and PCA on *20Newsgroups-I* and *20Newsgroups-II*, respectively. Here, each point represents a document, and the different color shapes represent the topic labels. For example, there are three different color shapes in Figure 5.2, representing three groups of news: black diamond for "comp.sys.ibm.pc", green triangle for "rec.sport.baseball" and red circle for "sci.med". In

the EV visualization (Figure 5.2(f)), documents with the same label are nicely clustered together while documents with different labels tend to be placed far away. In PLSA+PE and LSP (Figures 5.2(e) and (d)), documents are located slightly more mixed than those in EV. On the other hand, with PCA, MDS and ISOMAP (Figures 5.2(a)-(c)), documents with different labels are mixed, and thus the AC values of the corresponding layout are very low. These results also imply that the topic models generally provide better visualization layout. Figures 5.3(a)-(f) show 20-topic news groups visualized by the six methods. Similarly, EV provides the best view since news in similar topics are closer while news of distinct topics are placed further away.

As discussed earlier, by choosing the dimension of the subspace, EV can visualize documents with different granularity and enhance the interpretability of the visualization. In Figures 5.2(g)-(i) and 5.3(g)-(i), the representative documents selected in the low-rank subspace are embedded in a two-dimensional layout, for *20Newsgroups-I* and *20Newsgroups-II*, respectively. In Figures 5.2(g)-(i), we provide a series of visualization for *20Newsgroups-I*, from the most abstract view to the visual layout with considerate amount of details as the number of selected exemplars increases from 10 to 40. This result demonstrates that EV can use exemplars to summarize the distribution of the entire document collection. Similarly, Figures 5.3(g)-(i) illustrate the visualization from abstract to details when the number of exemplars increases from 100 to 400 in *20Newsgroups-II*. In these figures, the overlapping in the original layout (Figure 5.3(f)) is greatly reduced, making users easier to understand the relations between news documents.

Second, we compare the computational speed of six visualization methods: EV, PLSA+PE, PCA, LSP, MDS and ISOMAP. From a theoretical perspective, the time complexity of EV is $\mathcal{O}(n)$, PLSA+PE and PCA are $\mathcal{O}(n^2)$, LSP is $\mathcal{O}(f(n,s)) = \mathcal{O}(\max\{n^{\frac{3}{2}}, n\sqrt{s}\})$, and MDS and ISOMAP are $\mathcal{O}(n^3)$, where n is the number of documents and s is the condition number in LSP. Our experiments are performed on a machine with Quad 3GHz Intel Core2 processors

Table 5.3: Comparison of computation time (in seconds) for: EV, PLSA+PE, PCA, LSP, MDS and ISOMAP. A cross x indicates that an algorithm does not provide a result in a reasonable time.

Data size	EV	PLSA+PE	PCA	LSP	MDS	ISOMAP
n	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(f(n,s))$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$
1×10^3	0.49	0.42	0.40	15.25	20.48	200.05
2×10^3	0.95	1.50	1.36	30.40	216.62	1611.72
3×10^3	1.43	3.20	2.24	80.62	801.30	x
4×10^3	1.93	5.49	3.78	160.10	1881.00	x
5×10^3	2.55	8.38	x	x	x	x
1×10^4	5.79	x	x	x	x	x

and 4GB RAM. In order to compare under the same condition, the running time are reported based on a single iteration if an algorithm uses the iterative approach. Table 5.3 summarizes the computation time in seconds for all six methods with increasing number of documents. From Table 5.3, EV clearly is the quickest among the six, followed by PLSA+PE and PCA, while the computing time of LSP, MDS and ISOMAP increases quickly with the number of documents. More important, we observe that some algorithms fail to provide a result within a reasonable time for relatively large document sets. Specifically, ISOMAP is the slowest and cannot give a result when the matrix contains more than $3,000$ documents due to insufficient memory. When we have more than $10,000$ samples, only EV can provide a result within a reasonable computation time, while all other methods fail (indicated by a cross x in the table). Clearly, EV is suitable to visualize large text corpus we are increasingly facing these days thanks to its high computational efficiency.

We also develop an Exemplar-based Visualization software tool to offer a range of functions of creating visualization with user-specified configuration and thus supporting visual exploration of document data. First, when choose "**View All**" menu, the system can show all the documents at once for the *20Newsgroups* and *10PubMed* data sets. In this case, EV is the only one among the six algorithms that can produce a projection in a reasonable time. For example, Figure 5.4(a) shows visualization by EV for the $18,864$ documents in *20Newsgrups*. Again, each point represents a document, and the different color shapes represent the topic labels. Note that it is difficult to see the details because the number of documents is

very large, leading to extremely heavy overlapping. If one clicks "**View Exemplars**" and sets the number of exemplars at $1,000$, Figure 5.4(b) shows the representative documents selected by EV to summarize the whole document collection. Clearly, the cognitive overload and serious overlapping are greatly reduced. Here, a big color shape indicates the mean coordinate of documents for one group, calculated by $\mu_l = \frac{1}{n_l}\sum_{i=1}^{n} I(l_i = l))\mathbf{y}_i$, where n_l is the number of documents labeled with l. Obviously, documents with the same label are clustered together, and similar documents with closely related labels are placed nearby, such as "comp.graphics', "comp.os.ms.windows.misc' and "comp.windows.x" in the "computer" category, or "rec.autos", "rec.motorcycles", "rec.sport.baseball" and "rec.sport.hockey" in the "recreation" news group. Based on the visualized exemplars, EV provides several additional options for a user to further explore the data set. For example, on the click of "**View Clusters**", a magnified layout of all corresponding documents in the groups of "comp.graphics", "comp.os.ms.windows.misc" and "comp.windows.x" is given in Figure 5.4(c), which provides further details. Similarly, a user can specify a neighborhood (the rectangle in Figure 5.4(b)), clicking "**Zoom In**" will generate a magnified view of all or representative documents in the selected area. Also, if desired, further clustering and visualization can be performed in an area of interest, leading to a hierarchical structure for data exploration.

Figure 5.5 shows the EV model to visualize the $15,565$ documents in the *10PubMed* data set. Exemplars and means of *10PubMed* data illustrated in Figure 5.5(a) help us gain a better understanding on the distribution and relations of these documents. It is clear that documents with same disease are likely to be located closely while documents with different diseases are moved further away. We notice that there is less overlapping in the *10PubMed* data set than in *20Newsgroups*. One reason is that the number of topics in *10PubMed* is less than in *20Newsgroups* while another one is that the abstracts in the literature for various diseases is actually easier to be separated than the documents in different news groups. The average value of AC is about 60% in the *10PubMed* data set; it is only approximately 30% in *20Newsgroups*. If de-

sired, users can further explore the data set by clusters. In Figure 5.5(b), documents related to two diseases ("Gout" and "Chickenpox") are shown, where the selected exemplars (100 in total) are emphasized by the bigger black shapes. First, our method provides a clear visualization with little clutter. Second, users can quickly browse the large document collection by reading only the representative documents (exemplars) in each cluster. The actual time required by EV to produce visualization for *20Newsgroups* and *10PubMed* (with $1,000$ exemplars and $1,000$ iterations) are 30 and 25 minutes, respectively. These results clearly show that EV provides a very powerful tool for visualizing large text data sets.

5.4 Summary

In this chapter, we propose an Exemplar-based approach to Visualize (EV) extremely large text corpus. In EV, a representative text data subspace is first computed from the low-rank approximation of the original word-document matrix. Then, documents are soft clustered using the matrix decomposition and visualized in the Euclidean embedding space through parameter embedding. By selecting the representative documents, EV can visualize tens of thousands of documents with high accuracy (in retaining neighbor relations), high efficiency (in computation), and high flexility (through the use of exemplars).

The algorithms discussed here have been fully integrated into a visualization software package, which has been released publicly on the website [3]. In the future, we plan to conduct practical user studies to solicit feedbacks so that the software can be improved with more convenient and user-friendly features. We also intend to pursue incorporating topic detection model into our system, making it more appropriate for real-world data visualization. Another direction we are considering for the future work is to develop an interaction tool based on the EV model for the visualization of other types of data.

[3] http://vii.wayne.edu

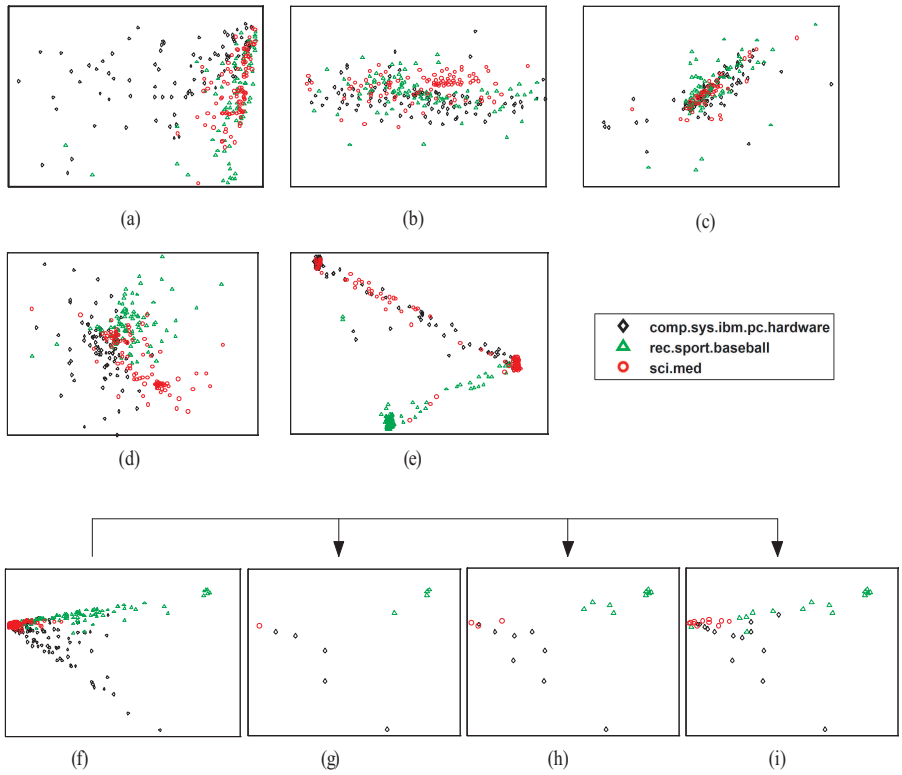


Figure 5.2: Visualization of documents in *20Newsgroups-I* (300 documents, 3 topics) by (a)PCA, (b)MDS, (c)ISOMAP, (d)LSP, (e)PLSA+PE, (f)EV, and visualization of (g)10 exemplars, (h)20 exemplars, (i)40 exemplars by EV.

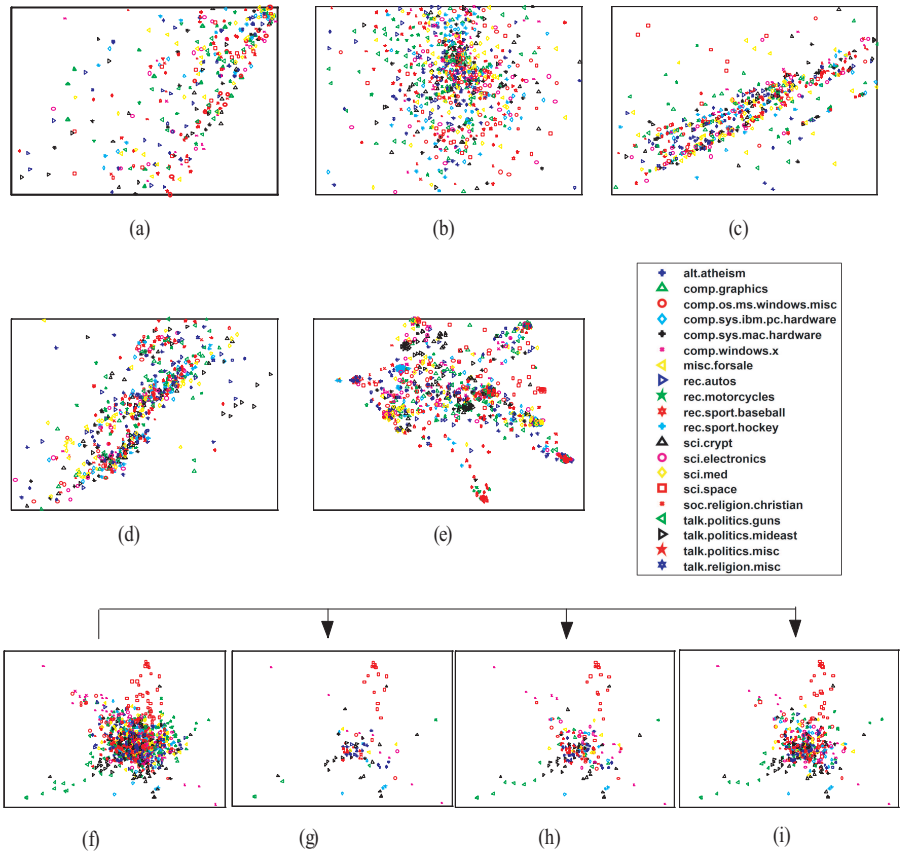


Figure 5.3: Visualization of documents in *20Newsgroups-II* (1000 documents, 20 topics) by (a)PCA, (b)MDS, (c)ISOMAP, (d)LSP, (e)PLSA+PE, (f)EV, and visualization of (g)100 exemplars, (h)200 exemplars, (i)400 exemplars by EV.

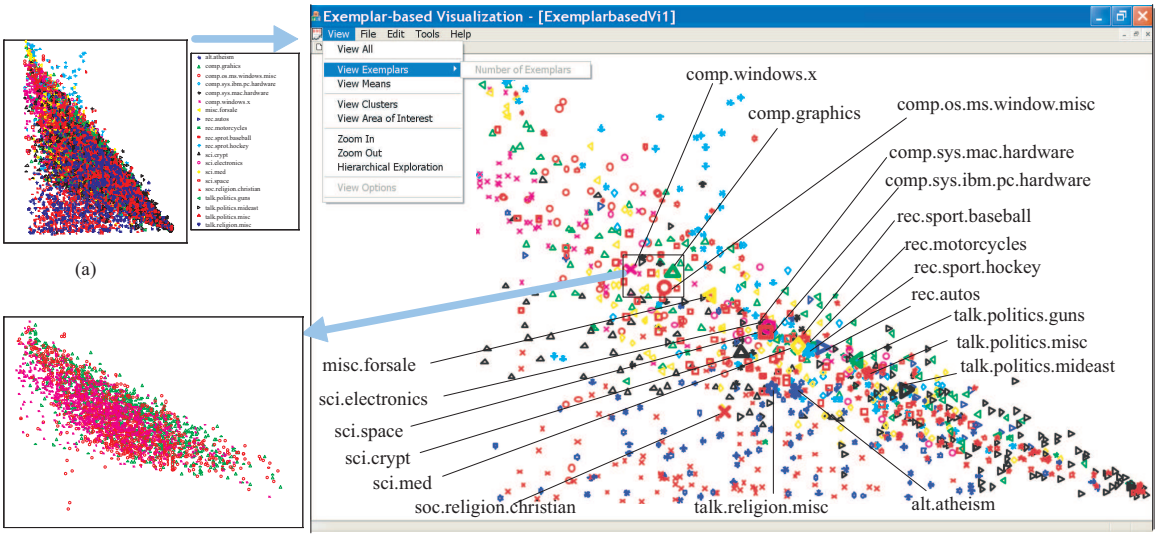


Figure 5.4: Visualization of documents in *20Newsgroups* (18,864 documents, 20 topics) by EV. Each point represents a document; each color shape represents a news topic; and the corresponding big color shape indicates the mean of a news group. Visualization of (a) all documents, (b) 1000 exemplars with their means, and (c) three similar groups of news: "comp.os.ms.windows.misc", "comp.graphics" and "comp.windows.x".

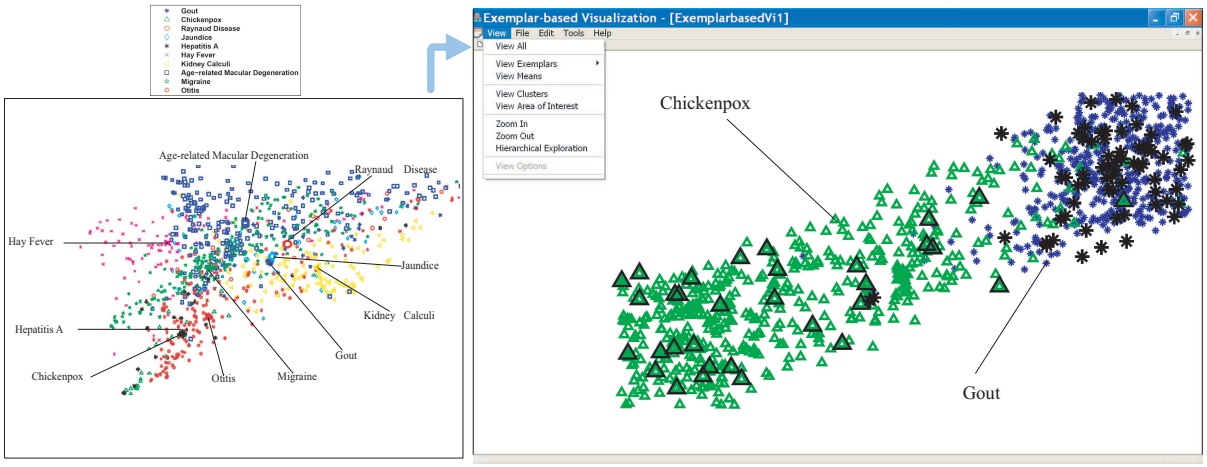


Figure 5.5: Visualization of abstracts in *10PubMed* (15,565 documents, 10 topics) by EV. Each point represents an abstract; each color shape represents a disease; and the corresponding big color shape indicates the means of an abstract group. Visualization of (a) 1000 exemplars with their means, and (b) two distinct groups of diseases: "Gout" and "Chickenpox" with the selected exemplars (100 in total), emphasized by the bigger black shapes.

CHAPTER 6

CONCLUSION

The purpose of this chapter is twofold. We first summarize the contributions made by this dissertation, then we point towards the future work.

6.1 Contributions

In this dissertation, we are dedicated to present matrix-based models for data clustering and visualization. We have made contributions in different research topics through matrix factorization, such as semi-supervised clustering, semi-supervised co-clustering, and exemplar-based clustering and visualization. Specifically,

1. Proposed and implemented SS-NMF: a semi-supervised approach for clustering based on non-negative matrix factorization. In the proposed framework, users are able to provide supervision in terms of *must-link* and *cannot-link* pairwise constraints on the data objects. We derived an iterative algorithm to perform symmetric tri-factorization of the data similarity matrix. We have mathematically shown the correctness and convergence of SS-NMF. Moveover, we proved that SS-NMF provides a general and unified framework for semi-supervised data clustering. Existing approaches can be considered as special cases of it. Empirically, we showed that SS-NMF outperforms well-established unsupervised and semi-supervised clustering methods in many real-world applications, such us text mining, gene expression analysis, images grouping and other publicly available UCI data sets clustering [14, 15, 16].

2. Developed and implemented a novel semi-supervised approach for data co-clustering: SS-NMF. In the proposed SS-NMF co-clustering 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. Tri-factorizations of the new matrices are then performed to obtain the simultaneous grouping of central data type and multiple feature modalities. Theoretically, we proved the convergence and correctness of the proposed co-clustering algorithm. In addition, we discussed that the relationship between our model and other representative co-clustering approaches. Our experimental results on publicly available data sets in text mining, bioinformatics and image grouping showed the superior performance of SS-NMF over existing methods for heterogeneous data co-clustering [17, 18, 19].

3. Proposed a novel method, EV, to visualize large document data sets in the low-rank subspace. From a theoretical perspective, EV presents a probabilistic multidimensional projection model with a sound objective function. Based on the rigorous derivation, the final visualization is obtained through iterative optimization. By selecting the representative rows and columns, EV obtains a compact approximation of the text data. This makes the visualization efficient and flexible. In addition, the selected exemplars neatly summarize the document collection and greatly reduce the cognitive overload in the visualization, leading to an easier interpretation of the text mining results. Through extensive experiments performed on the publicly available text data sets, we demonstrated the superior performance of EV when compared with existing visualization techniques [20].

6.2 Future Work

This dissertation also opens several venues for future work, with the focus on open problems in data mining.

1. **Exemplar-based Semi-supervised Learning for Complex Data**: Today digital data are accumulated at the faster than ever speed in science, engineering, biomedicine, and business. Usually, these extremely large data are highly complex, sharing one or several following prominent characteristics: they come unstructured with heterogeneous modalities or relations; they are tremendous in size with millions of objects and millions of

features; and user-guided knowledge is often embedded in large amounts of data. We have integrated user-guided information into matrix-based model to mine the relations among different objects or provided clear view of similar and disparate objects with representative samples in the entire data collection.

It would be interesting to to develop a novel model to integrate semi-supervised learning and exemplar-based model into matrix factorization techniques: (1) For heterogenous data, this model would soundly mine and visualize the rich structure of relations among different objects; (2) For large-scale data with high-dimensionality features, it would provide the most representative samples in the entire data collection by exemplar-based learning and identify the most relevant or discriminant features by multiple kernel manifold learning. Therefore, this model could facilitate effective and efficient cluster/summary analysis and help convenient visualization through embedding the data into a low-dimensional space; (3) For world knowledge within the learning system, it would inject domain information into the knowledge discovery process, thus providing a higher quality (e.g., accuracy) of mining results from complex data.

2. **Mining Interesting Domain Knowledge for Learning**: The major objective of data mining is to obtain useful information for humans who are interested in. Therefore, there is now a strong need for integrating data mining and knowledge inference. The biggest gap between what data mining systems can do and what we would like them to do is to relate the results of mining to the real-world decisions. Can we hand the results back to the user automatically and effectively? Compared to mining interesting information from complex data, the topic of mining interesting domain knowledge remains important.

   In this dissertation, we have solved this problem by injecting simple domain information into the knowledge discovery process. However, challenges remain as follows: how to make the discovered patterns "interesting" from the end-user perspective, how to transfer the "weak" or "soft" feedback or knowledge into the learning model, such as a user's

rating given as a percentage not a integer number.

In the future, our knowledge and experience gained in this dissertation will be applied to keep pursing the new directions.

REFERENCES

[1] AIROLDI, E. M., BLEI, D. M., FIENBERG, S. E., AND XING, E. P. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research 9* (2008), 1981–2014.

[2] BANERJEE, A., DHILLON, I. S., GHOSH, J., MERUGU, S., AND MODHA, D. S. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2004), pp. 509–514.

[3] BAR-HILLEL, A., HERTZ, T., SHENTAL, N., AND WEINSHALL, D. Learning distance functions using equivalence relations. In *Proceedings of the 20th International Conference on Machine Learning* (2003), pp. 11–18.

[4] BASU, S., BANERJEE, A., AND MOONEY, R. J. Semi-supervised clustering by seeding. In *Proceedings of the 19th International Conference on Machine Learning* (2002), pp. 27–34.

[5] BASU, S., BILENKO, M., AND MOONEY, R. J. A probabilistic framework for semi-supervised clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2004), pp. 59–68.

[6] BEKKERMAN, R., AND JEON, J. Multi-modal clustering for multimedia collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2007), pp. 1–8.

[7] BEKKERMAN, R., AND SAHAMI, M. Semi-supervised clustering using combinatorial MRFs. In *Proceedings of ICML-23 Workshop on Learning in Structured Output Spaces* (2006).

[8] BERKHIN, P. Survey of clustering data mining techniques. Tech. rep., Accrue Software, San Jose, CA, 2002.

[9] BERRY, M. W., PULATOVA, S. A., AND STEWART, G. W. Algorithm 844: Computing sparse reduced-rank approximations to sparse matrices. *ACM Transactions on Mathematical Software 31*, 2 (2005), 252–269.

[10] BLUM, A., AND MITCHELL, T. M. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory* (1998), pp. 92–100.

[11] BOLEY, D. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery 2(4)* (1998), 325–344.

[12] BORNER, K., CHEN, C., AND BOYACK, K. Visualizing knowledge domains. *Annual Review of Information Science and Technology 37* (2003), 1–51.

[13] CAI, D., SHAO, Z., HE, X., YAN, X., AND HAN, J. Mining hidden community in heterogeneous social networks. In *Proceedings of Workshop on Link Discovery: Issues, Approaches and Applications* (2005), pp. 58–65.

[14] CHEN, Y., REGE, M., DONG, M., AND FOTOUHI, F. Deriving semantics for image clustering from accumulated user feedbacks. In *Proceedings of 15th annual ACM International Conference on Multimedia* (2007), pp. 313–316.

[15] CHEN, Y., REGE, M., DONG, M., AND HUA, J. Incorporating user provided constraints into document clustering. In *Proceedings of the 7th IEEE International Conference on Data Mining* (2007), pp. 103–112.

[16] CHEN, Y., REGE, M., DONG, M., AND HUA, J. Non-negative matrix factorization for semi-supervised data clustering. *Journal of Knowledge and Information Systems 17*, 3 (2008), 355–379.

[17] CHEN, Y., WANG, L., AND DONG, M. A matrix-based approach for semi-supervised document co-clustering. In *Proceedings of the ACM 17th Conference on Information and Knowledge Management* (2008), pp. 1523–1524.

[18] CHEN, Y., WANG, L., AND DONG, M. Semi-supervised document clustering with simultaneous text representation and categorization. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Lecture Notes in Artificial Intelligence* (2009), vol. Part I, pp. 211–226.

[19] CHEN, Y., WANG, L., AND DONG, M. Non-negative matrix factorization for semi-supervised heterogeneous data co-clustering. *IEEE Transactions on Knowledge and Data Engineering* (appear to 2010).

[20] CHEN, Y., WANG, L., DONG, M., AND HUA, J. Exemplar-based visualization of large document corpus. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (November/December 2009), 1161–1168.

[21] CHERNOFF, H. Using faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association 68* (1973), 361–368.

[22] CHUNG, F. R. K. *Spectral Graph Theory*. American Mathematical Society, 1997.

[23] COX, T., AND COX, M. *Multidimensional Scaling*. Chapman and Hall/CRC, 2nd. ed., 2001.

[24] DE OLIVEIRA, M. C. F., AND LEVKOWITZ, H. From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualiztion and Computer Graphics 9*, 3 (2003), 378–394.

[25] DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society 39*, 1 (1977), 1–38.

[26] DHILLON, I. S. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2001), pp. 269–274.

[27] DHILLON, I. S., MALLELA, S., AND MODHA, D. S. Information-theoretic co-clustering. In *Proceedings the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2003), pp. 89–98.

[28] DING, C. Unsupervised feature selection via two-way ordering in gene experession analysis. *Bioinformatics 19*, 10 (2003), 1259–1266.

[29] DING, C., HE, X., AND SIMON, H. D. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of SIAM International Conference of Data Mining* (2005), pp. 606–610.

[30] DING, C., HE, X., ZHA, H., AND SIMON, H. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of 1st IEEE International Conference of Data Mining* (2001), pp. 107–114.

[31] DING, C., LI, T., AND JORDAN, M. I. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 1 (2010), 45–55.

[32] DING, C., LI, T., AND PENG, W. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics and Data Analysis 52*, 8 (2008), 3913–3927.

[33] DING, C., LI, T., AND PENG, W. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics and Data Analysis 52*, 8 (2008), 3913–3927.

[34] DING, C., LI, T., PENG, W., AND PARK, H. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2006), pp. 126–135.

[35] DUBNOV, S., EI-YANIV, R., GDALYAHU, Y., SCHNEIDMAN, E., TISHBY, N., AND YONA, G. A new nonparametric pairwise clustering algorithm based on iterative estimation of distance profiles. *Machine Learning 47(1)* (2002), 35–61.

[36] DUDA, R. O., HART, P. E., AND STORK, D. G. *Pattern Classification*, second ed. Wiley, New York, 2001.

[37] FAYYAD, U., GRINSTEIN, G., AND WIERSE, A. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann Publisher, 2001.

[38] FIEDLER, M. Algebraic connectivity of graphs. *Czechoslovak Mathematics Journal 23* (1973), 298–305.

[39] FIEDLER, M. Eigenvectors of acyclic matrices. *Czechoslovak Mathematics Journal 25* (1975), 607–618.

[40] FIEDLER, M. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematics Journal 25* (1975), 619–633.

[41] FISHER, D. Knowledge acquisition via incremental conceptual clustering. *Machine Learning 2* (1987), 139–172.

[42] FORTUNA, B., GROBELNIK, M., AND MLADENIC, D. Visualization of text document corpus. *Informatica 29*, 4 (2005), 497–502.

[43] FUA, Y.-H., WARD, M., AND RUNDENSTEINER, E. A. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of IEEE Conference on Visualization* (1999), pp. 43–50.

[44] GAO, B., LIU, T.-Y., FENG, G., QIN, T., CHENG, Q.-S., AND MA, W.-Y. Hierarchical taxonomy preparation for text categorization using consistent bipartite spectral graph copartitioning. *IEEE Transactions on Knowledge and Data Engineering 17*, 9 (2005), 1263–1273.

[45] GAO, B., LIU, T.-Y., AND MAO, W.-Y. Star-structured high-order heterogenous data co-clustering based on consistent information theory. In *Proceedings of the 6th IEEE International Conference on Data Mining* (2006), pp. 880–884.

[46] GAREY, M., AND JOHNSON, D. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, NY, 1982.

[47] GHAHRAMANI, Z., AND JORDAN, M. I. Supervised learning from incomplete data via the em approach. In *Advances in Neural Information Processing Systems* (1994), pp. 120–127.

[48] GRADY, L., AND SCHWARTZ, E. L. Isoperimetric graph partitioning for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28* (2006), 469–475.

[49] GUHA, S., RASTOGI, R., AND SHIM, K. Rock: A robust clustering algorithm for categorical attributes. In *Proceedings of the 15th International Conference on Data Engineering* (1999), pp. 512–521.

[50] HAGEN, L., AND KAHNG, A. B. New spectral methods for ratio cut partitioning and clustering. *IEEE Transaction on CAD of Integrated Circuits and Systems 11*, 9 (1992), 1074 – 1085.

[51] HAN, E.-H., AND KARYPIS, G. Centroid-based document classification: Analysis and experimental results. In *Proceedings of the 4th European Conference on Principles of Knowledge Discovery* (2000), pp. 424–431.

[52] HERSH, W., BUCKLEY, C., LEONE, T., AND HICKAM, D. Ohsumed: An interactive retrival evaluation and new large test collection for research. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval* (1994), pp. 192–201.

[53] HINNEBURG, A., AND KEIM, D. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining(1998)* (1998), pp. 58–65.

[54] HOFFMAN, T. Probablistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence* (1999), pp. 289–296.

[55] HOFFMAN, T., AND PUZICHA, J. Latent class models for collaborative filtering. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence* (1999), pp. 688–693.

[56] HOTHO, A., STAAB, S., AND STUMME, G. Text clustering based on background knowledge. Tech. Rep. 425, University of Karlsruhe, Institute AIFB, 2003.

[57] HOYER, P. O. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research 5* (2004), 1457–1469.

[58] HUANG, Y., AND MITCHELL, T. M. Text clustering with extended user feedback. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2006), pp. 413–420.

[59] INSELBERG, A., AND DIMSDALE, B. Parallel coordinates: a tool for visualizing multidimensional geometry. In *Proceedings of IEEE Conference on Visualization* (1990), pp. 361–378.

[60] IWATA, T., SAITO, K., UEDA, N., STROMSTEN, S., GRIFFITHS, T. L., AND TENENBAUM, J. B. Parametric embedding for class visualization. *Neural Computation 19* (2007), 2536–2556.

[61] IWATA, T., YAMADA, T., AND UEDA, N. Probabilistic latent semantic visualization: Topic model for visualizing documents. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2008), pp. 363–371.

[62] J. LEBLANC, M. O. W., AND WITTELS, N. Exploring n-dimensional database. In *Proceedings of IEEE Conference on Visualization* (1990), pp. 230–237.

[63] JAIN, A., AND VAZIRANI, V. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relazation. *Journal of the ACM* (2001), 274–296.

[64] JAIN, A. K., MURTY, M. N., AND FLYNN, P. J. Data clustering: a review. *ACM Computing Surveys 31*, 3 (1999), 264–323.

[65] JI, X., AND XU, W. Document clustering with prior knowledge. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2006), pp. 405–412.

[66] JOACHIMS, T. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning* (1999), pp. 200–209.

[67] JOLLIFFE, I. T. *Principal Component Analysis*, 2nd ed. Springer-Verlag, 2002.

[68] KAMVAR, S., KLEIN, D., AND MANNING, C. D. Spectral learning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (2003), pp. 561–566.

[69] KARYPIS, G., HAN, E. H., AND KUMAR, V. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer* (1999), 68–75.

[70] KARYPIS, G., AND KUMAR, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing* (1998), 359–392.

[71] KAUFMAN, L., AND ROUSSEEUW, P. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York, 1990.

[72] KEIM, D. A. Information visualization and visual data mining. *IEEE Transactions on Visualiztion and Computer Graphics 8*, 1 (2002), 1–8.

[73] KEIM, D. A., AND KRIEGEL, H.-P. Visdb: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications 14*, 5 (1994), 40–49.

[74] KEMP, C., TENENBAUM, J. B., GRIFFITHS, T. L., YAMADA, T., AND UEDA, N. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence* (2006), pp. 381–388.

[75] KLEIN, D., KAMVAR, S., AND MANNING, C. From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering. In *Proceedings of the 19th International Conference on Machine Learning* (2002), pp. 307–314.

[76] KULIS, B., BASU, S., DHILLON, I., AND MOONEY, R. Semi-supervised graph clustering: a kernel approach. In *Proceedings of the 22nd International Conference on Machine Learning* (2005), pp. 457–464.

[77] LANG, K. Newsweeder: Learning to filter netnews. In *Proceedings of the 12nd International Conference on Machine Learning* (1995), pp. 331–339.

[78] LEE, D., AND SEUNG, H. Algorithms for non-negative matrix factorization. In *Proceedings of the 13th Neural Information Processing Systems* (2001), pp. 556–562.

[79] LEE, D. D., AND SEUNG, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature 401* (1999), 788–791.

[80] LEWIS, D. D. Reuters-21578 text categorization test collection distribution 1.0. http://www.research.att/ lewis, 1999.

[81] LI, T. A general model for clustering binary data. In *Proceedings of 11st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2005), pp. 188–197.

[82] LI, T., AND DING, C. The relationships among various nonnegative matrix factorization methods for clustering. In *Proceedings of the 6th IEEE International Conference on Data Mining* (2006), pp. 362–371.

[83] LONG, B., WU, X., ZHANG, Z., AND YU, P. S. Spectral clustering for multi-type relational data. In *Proceedings of the 23rd International Conference of Machine Learning* (2006), pp. 585–592.

[84] LONG, B., ZHANG, Z., WU, X., AND YU, P. S. Unsupervised learning on k-partite graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2006), pp. 317–326.

[85] LONG, B., ZHANG, Z., AND YU, P. S. Co-clustering by block value decomposition. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2005), pp. 635–640.

[86] LONG, B., ZHANG, Z., AND YU, P. S. A probabilistic framework for relational clustering. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2007), pp. 470–479.

[87] MA, W.-Y., AND ZHANG, H. Benchmarking of image features for content-based retrieval. In *Proceedings of the 32th Asilomar Conference on Signals, Systems and Computers* (1998), pp. 253–257.

[88] MACQUEEN, J. Some methods for classsification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (1967), pp. 281–297.

[89] MCLACHLAN, G., AND PEEL, D. *Finite Mixture Models*, 1st ed. New York: John Wiley & Sons, Inc., 2002.

[90] MITCHELL, T. *Machine Learning*. McGraw-Hill, New York, NY, 1997.

[91] NG, A. Y., JORDAN, M., AND WEISS, Y. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14* (2002), pp. 849–856.

[92] NG, A. Y., AND JORDAN, M. I. On discriminative vs. genreative classifiers: A comparion of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14* (2002), pp. 605–610.

[93] NIGAM, K., MCCALLUM, A. K., AND MITCHELL, T. Text classification from labeled and unlabeled documents. *Machine Learning 39* (2000), 103–134.

[94] NIGAM, K., MCCALLUM, A. K., THRUN, S., AND MITCHELL, T. M. Learning to classify text from labeled and unlabeled documents. In *Proceedings of the 15th Conference of the American Association for Artificial Intelligence* (1998), pp. 792–799.

[95] PAULOVICH, F., AND MINGHIM, R. Hipp: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1229–1236.

[96] PAULOVICH, F., NONATO, L. G., MINGHIM, R., AND LEVKOWITZ, H. Least square projection: a fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualiztion and Computer Graphics 14*, 3 (2008), 564–575.

[97] PETROS, D., RAVI, K., AND MICHAEL, W. M. Fast monte carlo algorithms for matrices iii: computing a compressed approximate matrix decomposition. *SIAM Journal on Computing 36* (2006), 184–206.

[98] PORTER, M. An algorithm for suffix stripping. *Program 14*, 3 (1980), 130–137.

[99] ROWEIS, S. T., AND SAUL, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500 (2000), 2323 – 2326.

[100] SEEGER, M. Learning with labeled and unlabeled data. Tech. rep., Institute for ANC, Edinburgh, UK, 2000. http://www.dai.ed.as.uk/ seeger/papers.html.

[101] SHAN, H., AND BANERJEE, A. Bayesian co-clustering. In *Proceedings of the 13th IEEE International Conference on Data Mining* (2008), pp. 530–539.

[102] SHEIKHOLESAMI, G., CHATTERJEE, S., AND ZHANG, A. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the International Confrence on Very Large Databases* (1998), pp. 428–439.

[103] SHI, J., AND MALIK, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2000), 88–905.

[104] STREHL, A., AND GHOSH, J. A scalable approach to balanced, high-dimensional clustering of market-baskets. In *Proceedings of the 7th International Conference on High Performance Computing (2000)* (2000).

[105] SUN, J., XIE, Y., ZHANG, H., AND FALOUTSOS, C. Less is more: sparse graph mining with compact matrix decomposition. *Statistical Analysis and Data Mining 1*, 1 (2008), 6–22.

[106] TEJADA, R., MINGHIM, R., AND NONATO, L. On improved projection techniques to support visual exploaration of mutlidimensional data sets. *Information Visualization 2*, 4 (2003), 218–231.

[107] TENENBAUM, J. B., DE SILVA, V., AND LANGFORD, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319–2323.

[108] TREC. Text retrieval conference, http://trec.nist.gov.

[109] VAILAYA, A., JAIN, A., AND ZHANG, H. On image classification: City images vs. landscapes. *Patterm Recognition 31*, 12 (1998), 1921–1935.

[110] VAPNIK, V. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[111] WAGSTAFF, K., CARDIE, C., ROGERS, S., AND SCHROEDL, S. Constrained k-means clustering with background knowledge. In *Proceedings of the 18th International Conference on Machine Learning* (2001), pp. 577–584.

[112] WANG, X., SUN, J., ZHEN, Z., AND ZHAI, Z. Latent semantic analysis for multi-type interrelated data objects. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval* (2006), pp. 236–243.

[113] WISE, J., THOMAS, J. J., PENNOCK, K., LANTRIP, D., POTTIER, M., SCHUR, A., AND CROW, V. Visualizing the non-visual: spatial analysis and iteraction with information from text documents. In *Proceedings of Information Visualization* (1995), pp. 51–58.

[114] XING, E. P., NG, A. Y., JORDAN, M. I., AND RUSSELL, S. Distance metric learning, with application to clustering with side-information. In *Proceedings of 16th Neural Information Processing Systems* (2002), pp. 505–512.

[115] XU, W., LIU, X., AND GONG, Y. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2003), pp. 267–273.

[116] ZHANG, T., RAMAKRISHNAN, R., AND LIVNY, M. Birch: An efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (1996), pp. 103–114.

ABSTRACT

DATA CLUSTERING AND VISUALIZATION THROUGH MATRIX FACTORIZATION

by

YANHUA CHEN

MAY 2010

Advisor: Dr. Ming Dong

Major: Computer Science

Degree: Doctor of Philosophy

Clustering is traditionally an unsupervised task which is to find natural groupings or clusters in multidimensional data based on perceived similarities among the patterns. The purpose of clustering is to extract useful information from unlabeled data. In order to present the extracted useful knowledge obtained from clustering in a meaningful way, data visualization becomes a popular and growing area of research field. Visualization can provide us a better understanding of large and complex data sets by displaying them in a logical layout. The contribution of this dissertation is two-fold: Semi-Supervised Non-negative Matrix Factorization (SS-NMF) for data clustering/co-clustering and Exemplar-based data Visualization (EV) through matrix factorization. Compared to traditional data mining models, matrix-based methods are fast, easy to understand and implement, especially suitable to solve large-scale challenging problems in text mining, image grouping, medical diagnosis, and bioinformatics.

In this dissertation, we present two effective matrix-based solutions in the new directions of data clustering and visualization.

First, in many practical learning domains, there is a large supply of unlabeled data but limited labeled data, and in most cases it might be expensive to generate large amounts of labeled data. Traditional clustering algorithms completely ignore these valuable labeled data and thus are inapplicable to these problems. Consequently, semi-supervised clustering, which

can incorporate the domain knowledge to guide a clustering algorithm, has become a topic of significant recent interest. Thus, we develop a Non-negative Matrix Factorization (NMF) based framework to incorporate prior knowledge into data clustering. Moreover, 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 co-clustering of multiple data types, usually involving high heterogeneity. To this end, we extend SS-NMF to perform heterogeneous data co-clustering. From a theoretical perspective, SS-NMF for data clustering/co-clustering is mathematically rigorous. The convergence and correctness of our algorithms are proved. In addition, we discuss the relationship between SS-NMF with other well-known clustering and co-clustering models. Second, most of current clustering models only provide the centroids (e.g., mathematical means of the clusters) without inferring the representative exemplars from real data, thus they are unable to better summarize or visualize the raw data. A new method, Exemplar-based Visualization (EV), is proposed to cluster and visualize an extremely large-scale data. Capitalizing on recent advances in matrix approximation and factorization, EV provides a means to visualize large scale data with high accuracy (in retaining neighbor relations), high efficiency (in computation), and high flexibility (through the use of exemplars). Empirically, we demonstrate the superior performance of our matrix-based data clustering and visualization models through extensive experiments performed on the publicly available large scale data sets.

AUTOBIOGRAPHICAL STATEMENT

YANHUA CHEN

Yanhua Chen received the MS degree in Computer Science and Engineering from Michigan State University, East Lansing, MI, in 2004. She is currently a PhD candidate in the Machine Vision and Pattern Recognition Laboratory in 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, ACM and INNS.