Matrix factorization framework for simultaneous data (co-)clustering and embedding

Kais Allab

▶ **To cite this version:**

Kais Allab. Matrix factorization framework for simultaneous data (co-)clustering and embedding. Data Structures and Algorithms [cs.DS]. Université Sorbonne Paris Cité, 2016. English. NNT : 2016USPCB083 . tel-02179223

HAL Id: tel-02179223
https://theses.hal.science/tel-02179223

Submitted on 10 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ÉCOLE DOCTORALE INFORMATIQUE, TÉLÉCOMMUNICATIONS ET ÉLECTRONIQUE (EDITE DE PARIS)

Thesis

Matrix Factorization Framework for Simultaneous Data (Co-)Clustering and Embedding

---

by

Kais Allab

This dissertation is submitted for the degree of

DOCTOR OF COMPUTER SCIENCE

PARIS DESCARTES UNIVERSITY

Thesis supervisors:

- Mohamed NADIF	Professor, University of Paris Descartes
- Lazhar LABIOD	Dr, University of Paris Descartes

Examining committee :

- Stephane CANU	Professor, INSA of Rouen
- Eric GAUSSIER	Professor, University of Grenoble Alps
- Gilbert SAPORTA	Professor, CNAM of Paris
- Yves GRANDVALET	Research Director, University of Compiègne
- Pascale KUNTZ-COSPEREC	Professor, University of Nantes

ÉCOLE DOCTORALE INFORMATIQUE, TÉLÉCOMMUNICATIONS ET ÉLECTRONIQUE (EDITE DE PARIS)

Thèse

Matrix Factorization Framework for Simultaneous Data (Co-)Clustering and Embedding

---

par

Kais Allab

Thèse présentée pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ EN INFORMATIQUE

UNIVERSITÉ DE PARIS DESCARTES

Directeur de thèse :	
- Mohamed NADIF	Professeur, Université Paris Descartes
Co-encadrant:	
- Lazhar LABIOD	Maitre de Conférence, Université Paris Descartes
Rapporteurs :	
- Stephane CANU	Professeur, INSA de Rouen
- Eric GAUSSIER	Professeur, Université Grenoble Alpes
Examinateurs :	
- Gilbert SAPORTA	Professeur, CNAM de Paris
- Yves GRANDVALET	Directeur de recherches, Université de Compiègne
- Pascale KUNTZ-COSPEREC	Professeur, Université de Nantes

Remerciements

A l'issue de la rédaction de cette recherche, je suis convaincu que la thèse est loin d'être un travail solitaire. En effet, je n'aurais jamais pu réaliser ce travail doctoral sans le soutien d'un grand nombre de personnes dont la générosité, la compétence, la rigueur scientifique et la clairvoyance m'ont beaucoup appris et m'ont permis de progresser. Ils ont ètè et resteront des moteurs de mon travail de chercheur.

J'aimerais tout d'abord remercier mon directeur de thèse, Pr. Mohamed Nadif, pour la confiance qu'il m'a accordée en acceptant d'encadrer ce travail doctoral, pour ses multiples conseils et pour toutes les heures qu'il a consacrées à diriger cette recherche. J'aimerais également lui dire à quel point j'ai apprécié sa grande disponibilité et sa relecture méticuleuse des documents que je lui ai adressés. Enfin, j'ai été extrêmement sensible à ses qualités humaines d'écoute et de compréhension tout au long de ce travail doctoral. Je tiens également à remercier Dr. Lazhar Labiod pour sa participation active à cette thèse en tant que co-encadrant.

Je tiens exprimer ma gratitude et mes plus vifs remerciements tous les membres de mon jury: Pr. Stephane CANU, Pr. Eric GAUSSIER, Pr. Gilbert SAPORTA, DR. Yves GRANDVALET et Pr. Pascale KUNTZ-COSPEREC, pour le temps qu'ils ont accordé à la lecture de cette thèse, à l'élaboration de leur rapport et pour les suggestions et les remarques judicieuses dont ils m'ont fait part.

Enfin, j'adresse ma gratitude et mon respect à tous les membres du LIPADE, et en particulier à mes collégues de l'équipe MLDS, Melissa Ailem et Aghiles Salah pour leur support et leur encouragements et pour tous les bons moments qu'on partagé durant ce travail doctoral.

Résumé

Les progrès des technologies informatiques et l'augmentation continue des capacités de stockage ont permis de disposer de masses de données de très grandes tailles et de grandes dimensions. Le volume et la nature même des données font qu'il est de plus en plus nécessaire de développer de nouvelles méthodes capables de traiter, résumer et d'extraire l'information contenue dans de tels types de données.

D'un point de vue extraction des connaissances, la compréhension de la structure des grandes masses de données est d'une importance capitale dans l'apprentissage artificiel et la fouille de données. En outre, contrairement á l'apprentissage supervisé, l'apprentissage non supervisé peut fournir des outils pour l'analyse de ces ensembles de données en absence de groupes (classes). Dans cette thèse, nous nous concentrons sur des méthodes fondamentales en apprentissage non supervisé notamment les méthodes de réduction de la dimension, de classification simple (*clustering*) et de classification croisée (*co-clustering*).

Notre contribution majeure est la proposition d'une nouvelle manière de traiter simultanément la classification et la réduction de dimension. L'idée principale s'appuie sur une fonction objective qui peut être décomposée en deux termes, le premier correspond á la réduction de la dimension des données, tandis que le second correspond á l'objectif du clustering et celui du co-clustering. En s'appuyant sur la factorisation matricielle, nous proposons une solution prenant en compte simultanément les deux objectifs : réduction de la dimension et classification.

Nous avons en outre proposé des versions régularisées de nos approches basées sur la régularisation du Laplacien afin de mieux préserver la structure géométrique des données. Les résultats expérimentaux obtenus sur des données synthétiques ainsi que sur des données réelles montrent que les algorithmes proposés fournissent d'une part de bonnes représentations dans des espaces de dimension réduite et d'autre part permettent d'améliorer la qualité des clusters et des co-clusters.

Motivés par les bons résultats obtenus par les méthodes du clustering et du co-clustering basés sur la régularisation du Laplacien, nous avons développé un nouvel algorithme basé sur l'apprentissage multi-variétés (multi-manifold) dans lequel une *variété consensus* est approximée par la combinaison d'un ensemble de variétés candidates reflétant au mieux la structure géométrique locale des données.

Enfin, nous avons aussi étudié comment intégrer des contraintes dans les Laplaciens utilisés pour la régularisation á la fois dans l'espace des objets et l'espace des variables. De cette faon, nous montrons comment des connaissances a priori peuvent contribuer à l'amélioration de la qualité du co-clustering.

Abstract

Advances in computer technology and recent advances in sensing and storage technology have created many high-volume, high-dimensional data sets. This increase in both the volume and the variety of data calls for advances in methodology to understand, process, summarize and extract information from such kind of data. From a more technical point of view, understanding the structure of large data sets arising from the data explosion is of fundamental importance in data mining and machine learning. Unlike supervised learning, unsupervised learning can provide generic tools for analyzing and summarizing these data sets when there is no well-defined notion of classes. In this thesis, we focus on three important techniques of unsupervised learning for data analysis, namely data dimensionality reduction, data clustering and data co-clustering.

Our major contribution proposes a novel way to consider the clustering (resp. co-clustering) and the reduction of the dimension simultaneously. The main idea presented is to consider an objective function that can be decomposed into two terms where one of them performs the dimensionality reduction while the other one returns the clustering (resp. co-clustering) of data in the projected space simultaneously. We have further introduced the regularized versions of our approaches with graph Laplacian embedding in order to better preserve the local geometry of the data. Experimental results on synthetic data as well as real data demonstrate that the proposed algorithms can provide good low-dimensional representations of the data while improving the clustering (resp. co-clustering) results.

Motivated by the good results obtained by graph-regularized-based clustering (resp. co-clustering) methods, we developed a new algorithm based on the multi-manifold learning. We approximate the intrinsic manifold using a subset of candidate manifolds that can better reflect the local geometrical structure by making use of the graph Laplacian matrices. Finally, we have investigated the integration of some selected instance-level constraints in the graph Laplacians of both data samples and data features. By doing that, we show how the addition of priory knowledge can assist in data co-clustering and improves the quality of the obtained co-clusters.

Contents

List of Figures

List of Tables

Chapter 1

Thesis Outline

1.1 Introduction

Data analysis has a capital role to play in several real-world domains, such as medicine and market analysis. Specifically, with recent advances in sensing and storage technology, many high-volume data collections were created and there is a need for accurate automatic data analysis tools in order to be able to process these great amounts of data in a timely manner. Lately, a new fashion of unsupervised data analysis algorithms, which consider the clustering (resp. co-clustering) and the reduction of the dimension simultaneously, has emerged. These approaches takes advantage of the mutual reinforcement between a manifold learning technique which provide a low-dimensional representation of data and a matrix factorization based clustering (resp. co-clustering) method that learns this low-dimensional representation and lends itself to a clustering (resp. co-clustering) interpretation.

Clustering is a fundamental topic in unsupervised machine learning. It consists of detecting the best structure inferred by the distribution of a set of non labeled data. In this context, it aims at organizing the data in homogeneous groups (clusters) by respecting both the cohesion and the separation. This organization is only made on data samples according to all the data features. Recently, Non-negative Matrix Factorization (NMF) [Lee and Seung, 1999] has become one of the most frequently used in clustering. NMF was proposed to learn a parts-based representation, but it focuses on unilateral clustering i.e. on only one of the two sets of samples or features of a data matrix.

However, in many real world applications, the data set to be analyzed involves two types. For example, words and documents in document analysis, bloggers and content in social networks, users and product in recommendation systems, experimental conditions and genes in microarray data analysis. In addition, usually there exist close relationships between the two types of data points, and it is difficult for the traditional clustering algorithms to use this relationship information efficiently. Motivated by the duality between samples and feature clusters, a number of different formulations of the co-clustering problem have been proposed to cluster simultaneously samples and features sets, using different mathematical concepts. These include the bi-clustering model [Cheng and Church, 2000], graph-based methods [Dhillon, 2001a], information-theoretic [Dhillon et al., 2003a,c], model-based co-clustering methods [Govaert and Nadif, 2003, 2005, 2014] and co-clustering methods based on matrix factorization [Anagnostopoulos et al., 2008]. These last, have recently been emerging as a promising tool for co-clustering, mainly because of the simplicity of the formalization and the close relationships to other well-studied problems, such as spectral clustering and matrix decomposition [Ding et al., 2006b; Long et al., 2005]. There are many different Co-clustering approaches fulfilling this task; see for instance [Anagnostopoulos et al., 2008].

Despite the popularity of factorization-based clustering and co-clustering methods, one drawback is that they rest on only a global Euclidean geometry, hence a local manifold geometry is not fully considered. To address this major limitation, some researchers have sought to take into account a local geometrical structure. In order to take into account the manifold structures in both sample and feature spaces, many different graph regularization based clustering (resp. co-clustering) approaches were proposed [Gu and Zhou, 2009; Shang et al., 2012; Wang et al., 2011b; Wang and Zhang, 2013].

To this end, manifold learning technique can be used to map a set of high-dimensional data into a low-dimensional space, while preserving the intrinsic structure of the data. These dimensionality reduction methods include different techniques for capturing the non-linearity of the underlying manifold, and they incorporate local distance information in different ways. Furthermore, the effectiveness of different dimensionality reduction methods varies, and it has been shown that no single method constantly outperforms the others. Rather than choosing a single method, therefore, we seek to apply a set of dimensionality reduction methods and to merge the output of the different methods. Indeed, multi-manifold learning was proposed to approximate the intrinsic manifold using a subset of candidate manifolds, which can better reflect the local geometrical structure by making use of the graph Laplacian. For example, some linear approaches for multi-manifold learning were proposed in [Fan et al., 2012; Goldberg et al., 2009; Lu et al., 2013; Yang et al., 2011]. These multi-manifold learning algorithms aim to overcome the drawbacks of single manifold learning methods and to combine the different data structures to which they give rise.

Recently, semi-supervised co-clustering algorithms, referred as *constrained co-clustering*, has emerged. These new algorithms can incorporate some background knowledge, allowing the user to guide the co-clustering process and improve the quality of its results. This a priori information is given to the algorithm as a set of pairwise constraints involving pairs of data points and expressing some restrictions or preferences about whether or not these pairs of data points should be in the same co-cluster. These pairwise constraints do not have to be numerous or be distributed among the whole data set in order to have a noticeable effect on the co-clustering process, which enables us to attain large improvements in the final quality of the co-clusters. Furthermore, using the measure of informativeness, the constrained co-clustering method selects the constraints that can correct the failures of most of the basic clustering and co-clustering methods. This is specifically relevant with the presence of some critical data located on the boundaries among the classes.

1.2 Motivation

Although clustering and co-clustering techniques represent active research topics in machine learning that have already proven their efficiency in many real-world applications, several data mining scenarios may degrade their performance due to different reasons. We can divide these reasons into those arising from the data structure and those caused by applications constraints. The former include problems related to high dimensionality, sparsity and heterogeneity of the data. Indeed, as we are now able to collect and extract a large number of features from raw data, the dimension of feature vectors increases and may even exceed the size of the vector of instances. This phenomenon, also known as the curse of dimensionality [Friedman, 1994] is directly associated with sparsity problems and may yield bad results in terms of clustering. Indeed, clustering and, by extension, co-clustering aim to group objects that are close and therefore rely on the notion of distance or similarity. However, in high dimensional data, all objects tend to be equidistant from one another.

Moreover, the aim of cluster analysis is the discovery of a finite number of homogeneous classes from data. These classes can be assumed to lie in a low-dimensional subspace of data. Generally when a user aims to clustering (resp. co-clustering), he then seeks to visualize the clusters in a reduced dimension space. This procedure can be carried out into two simple steps:

- **Step 1. Data Embedding:** Principal Component Analysis (PCA) is performed, and the first few components are retained.
- **Step 2. Data Clustering:** A clustering method (k-means) is performed on these first principal components.

This two-step procedure is called **tandem clustering** by Arabie and Hubert [Arabie and Hubert, 1994] and has been discouraged by several authors [Arabie and Hubert, 1994; Vichi and Kiers, 2001]. Because the first few principal components of PCA do not necessarily reflect the cluster structure in data, therefore the appropriate clustering result may not be obtained by using the tandem clustering approach. In our thesis, unlike to tandem clustering methods that combine a dimension reduction method (e.g PCA) and a clustering method (such as k-means) separately, we provide two convenient ways to integrate the data embedding and the data clustering steps into a single framework which performs the two tasks simultaneously [Timmerman et al., 2010; Vichi and Saporta, 2009]. This convenience is mainly due to two reasons. Firstly, clustering and dimension reduction are performed via an iterative optimization procedure to mutually reinforce the relationships between the coefficients. Secondly, the mutually reinforcing optimization exploits the relationships of the data clustering and dimension reduction and enables a simultaneous data clustering and embedding. This allows a better approximation of data reduction by a clustering solution.

Recently, the use of NMF for partitional clustering has attracted much interest because of the simplicity of the formalization and its close relationships to other well-studied problems, such as spectral clustering and matrix decomposition [Ding et al., 2006b; Li and Ding, 2006; Long et al., 2005]. Indeed, several theoretical papers also appeared proving the equivalence between NMF, spectral clustering and the k-means algorithm [Ding et al., 2005]. For instance, Zass and Shashua [2005] show that spectral clustering, normalized cuts, and Kernel k-means are particular cases of the clustering with NMF under a doubly stochastic constraint. However, NMF was proposed to learn a parts-based representation, but it focuses on unilateral clustering. Largely because of this, Nonnegative Matrix Tri-Factorization (NMTF) [Wang et al., 2011a; Yoo and Choi, 2010] has been developed for co-clustering dyadic data. Motivated by this, all our proposed clustering and co-clustering methods are based on matrix factorization. In spite of the very practical nature of these advantages, one drawback of the factorization-based clustering (resp. co-clustering) methods is that they are based only on the global Euclidean geometry, and the local manifold geometry is not fully considered. In addition, we known that most of the dimensionality reduction methods provides an embedding for the data lying on a linear manifold. However, in many applications, data lie in a non-linear manifold. In order to tackle this major limitations, motivated by recent progress in matrix factorization and manifold learning, one popular method is to use the graph Laplacian based embedding to incorporate the manifold information. Unfortunately, the first graph-regularized-based clustering (resp. co-clustering) methods fail to maximally approximate the intrinsic manifolds of both sample and feature spaces. To this end, multi-manifold learning was proposed to approximate the intrinsic manifold using a subset of candidate manifolds that can better reflect the local geometrical structure by making use of the graph Laplacian matrices.

Finally, in many real world applications, the data set to be analyzed presents obstacles such as large dimension, sparsity, heterogeneity and negativity. For this reason, efforts have been made in recent years to extend existing co-clustering methods to constrained co-clustering [Chen et al., 2010; Pensa and Boulicaut, 2008; Song et al., 2010; Wang et al., 2008b]. These last cluster simultaneously samples and feature sets, guided by certain supervisory information. Usually, this background knowledge can be represented as a set of pairwise constraints that can be generated from a subset of labeled data. Most of these methods encodes *Must-link* (*ML*) and *Cannot-link* (*CL*) constraints by modifying the graph Laplacian, constraining the underlying eigenspace, or by encoding them as part of a constrained optimization problem. In our contribution, we propose new applications of constrained co-clustering which, besides the similarity information encoded in the Laplacian graph in both sample and feature sides, allows to use label information to modify both Laplacian graphs according to the specified pairwise constraints.

1.3 Outline

The main novel contributions of this thesis are presented in chapters 3-7. The thesis manuscript is organized as follows.

Chapter 2 is a brief survey on clustering and co-clustering topics. First, we introduce basic principles related to clustering approaches and we describe the criteria that we use in this thesis in order to evaluate the performance of clustering algorithms. Then, we give a definition of co-clustering and describe the state-of-the-art methods and theoretical results that exist on this matter. Finally, we describe several Matrix Factorization based Clustering (resp. Co-clustering) algorithms which will be used or referenced along this work and we enumerate some research opportunities still open in this domain, some of which will be addressed in this thesis.

Chapter 3 [Allab et al., 2015b, 2016a] proposes a novel approach to finding an optimal subspace of multi-dimensional variables for identifying a partition of the set of objects. The proposed solution, relying on PCA and Semi-NMF, combines *simultaneously* the dimensionality reduction and the clustering (figure 1.1). The use of a low-dimensional representation is of help in providing better separated clusters and then easily interpretable.

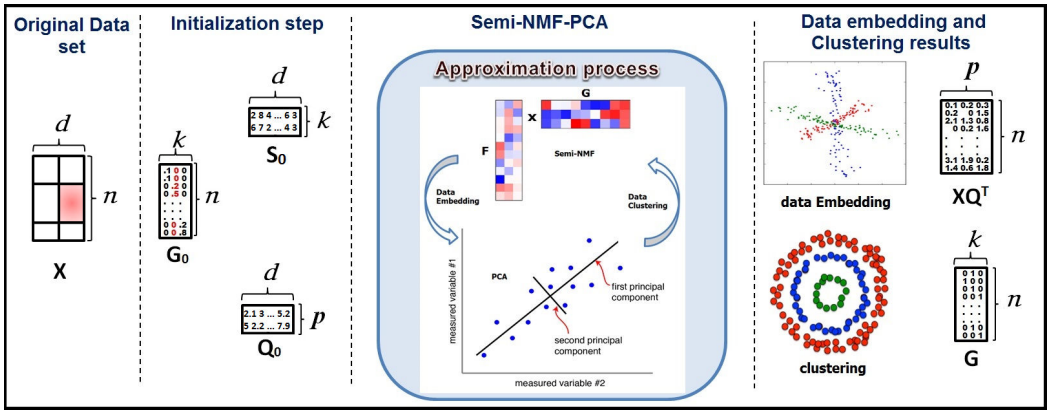


Figure 1.1 – Schema of SemiNMF-PCA method.

Chapter 4 [Allab et al., 2016b] aims to develop a new method of clustering in a spectral clustering framework. Spectral clustering is often based on a tandem approach where the two steps: affinity matrix eigendecomposition and k-means clustering, are performed separately. In this chapter we propose to perform simultaneously the eigendecomposition of the affinity matrix and clustering tasks, and to use the *Power method* to speed up the unified process

convergence (figure 1.2). We show that by doing so, our method can learn low-dimensional representations that are better suited to clustering.

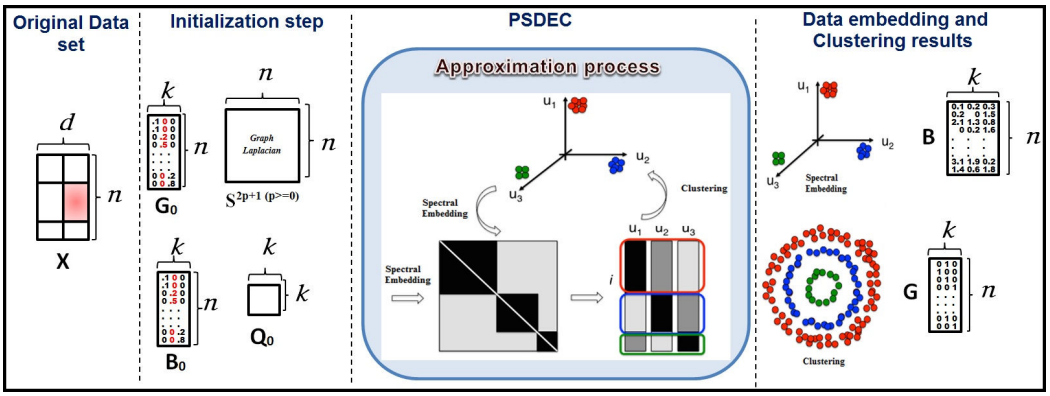


Figure 1.2 – Schema of PSDEC method.

Chapter 5 [Allab et al., 2016c] presents a novel way to consider the co-clustering and the reduction of the dimension simultaneously (figure 1.3). We show how we can extend the approach proposed in chapter 4 to tackle the co-clustering problem.

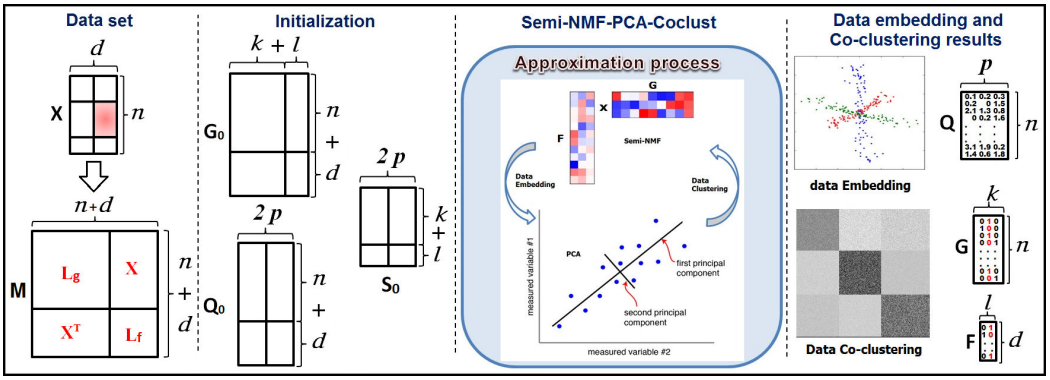


Figure 1.3 – Schema of SemiNMF-PCA-Coclust method.

Chapter 6 [Allab et al., 2015a] summarises our Multi-Manifold Matrix Decomposition for Co-clustering (M3DC) algorithm. Specifically, multiple candidate manifolds are constructed separately to take local invariance into account. Then, multi-manifold learning is employed to approximate the optimal intrinsic manifold, which better reflects the local geometrical structure, by linearly combining these candidate manifolds. The candidate manifolds are obtained using various manifold-based dimensionality reduction methods (figure 1.4).

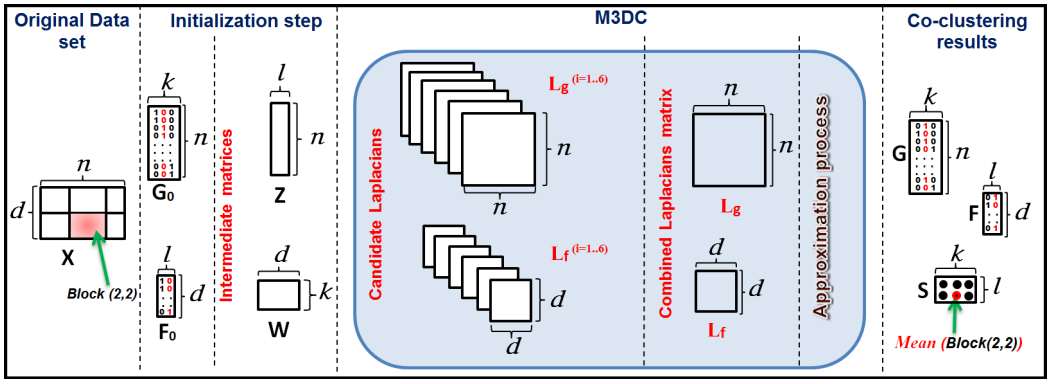


Figure 1.4 – Schema of M3DC method.

Chapter 7 [submitted in KBS journal] is devoted to constrained co-clustering methods under the guidance of some supervisory information. This information can take the form of pairwise constraints that indicate similarities or dissimilarities in the set of samples and the set of features. Based on matrix 3−factor decomposition, the aim of the proposed approach, referred to as Constrained Matrix Decomposition based Co-Clustering (CMDC) (figure 1.5), is to co-cluster efficiently data sets by introducing the most beneficial background knowledge on both the sample and feature spaces. Using Laplacian locality preserving, we project the samples and the features into lower-dimensional subspaces while preserving their local geometry.

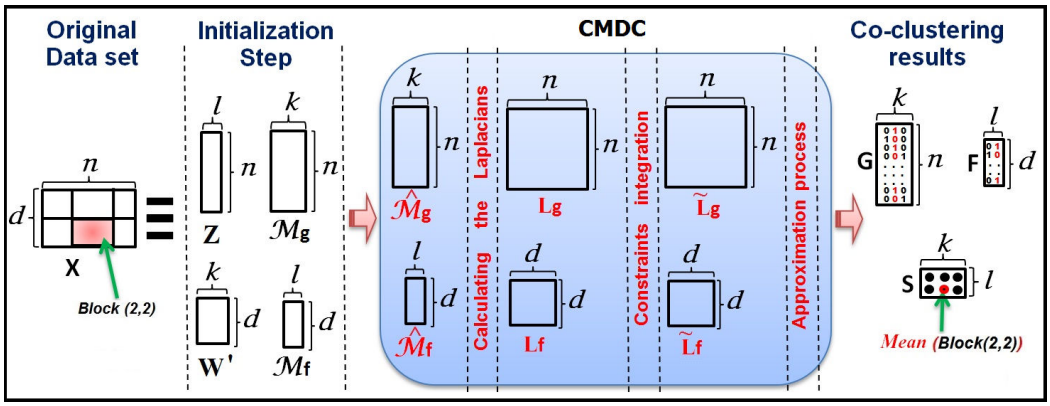
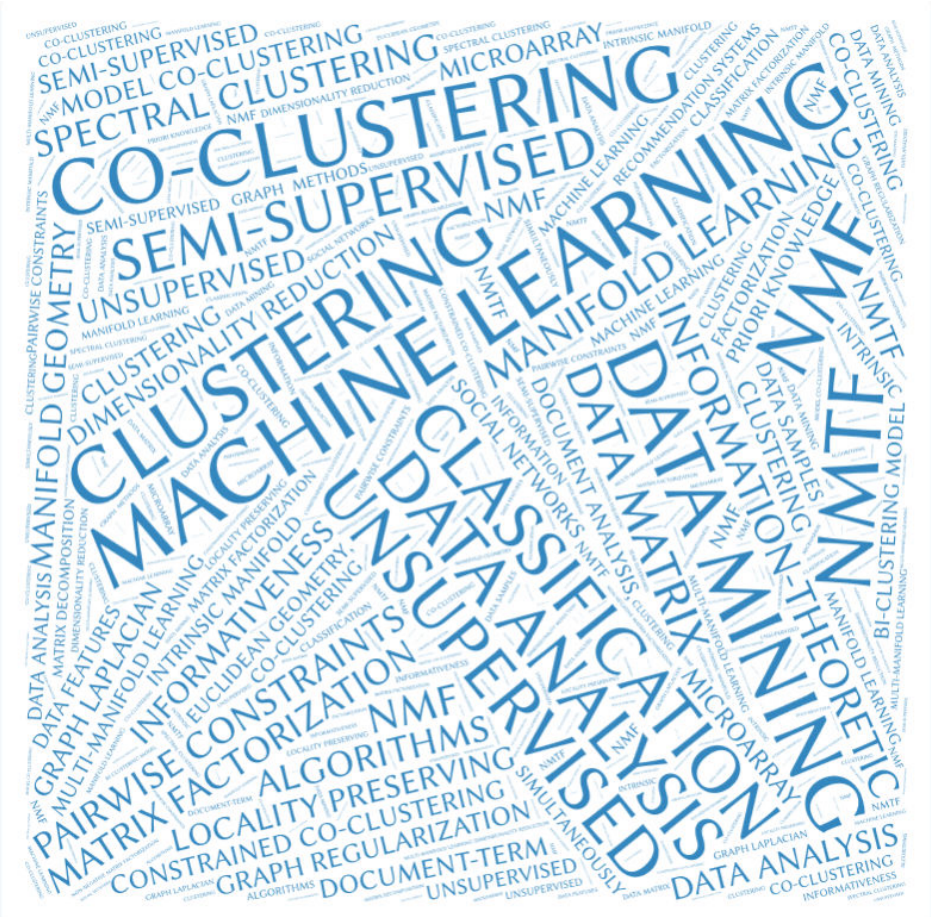


Figure 1.5 – Schema of CMDC method.

Finally, we present the conclusions of the thesis and a summary of the future research directions. Note that we try to keep the chapters introducing the main novel contributions of the thesis as self-contained as possible each of them contains its an introduction to its specific topic and its own state-of-the-art.

Chapter 2

Clustering and Co-clustering: A brief review



2.1 Clustering, definitions and algorithms

2.1.1 Clustering and Challenges

Data clustering, is also called cluster analysis, segmentation analysis, taxonomy analysis, or unsupervised classification. By definition, for a given set of data points and a similarity/dissimilarity measure, clustering is a method of creating groups of objects (clusters), in such a way that data objects in the same cluster are as similar as possible and data objects in different clusters are as dissimilar as possible. Unlike the classification in which the clusters are known, clustering can be viewed as an exploratory data analysis. Therefore, the explorer might have no or little information about the parameters of the resulting cluster analysis. Many fundamental questions arise when dealing with clustering.

- What is the interest in clustering the set of data objects?
- How many clusters?
- What are the relevant objects for the cluster analysis?
- What are the relevant features that describe the objects?
- Can we combine simultaneously clustering and visualization?
- What is suitable algorithm for data clustering?
- What is the quality of the obtained clustering?

In the sequel, we introduce some concepts that will be encountered frequently in cluster analysis.

2.1.2 Data Samples and Features

In machine learning and statistics, different terms can be used to express the same thing. For instance, given a data set, the data point, record, instance, observation, individual, record and sample are all used to denote a single data object. In our work, we will use data point or data sample to denote a single object. Also, we shall use data feature to denote a record scalar component. We almost use the case-by-variables data structure [Hartigan, 1975].

Through the thesis, the data set to classify is organized in a matrix. Given a data matrix X, it contains n objects $X := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and each object is in a d-dimensional space, i.e. each object $\mathbf{x}_i = (x_{1i}, x_{2i}, \ldots, x_{ji}, \ldots, x_{di})^T$ is a vector denoting the i^{th} data sample and x_{ji} is a scalar denoting the j^{th} component of $\mathbf{x}_i$. The number of features d is also called

dimensionality of the data set. This can be expressed in a matrix format as:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ x_{d1} & x_{d2} & \cdots & x_{dn} \end{bmatrix} \tag{2.1}$$

2.1.3 Similarity/Dissimilarity and Distance

A similarity/dissimilarity measures, or a distance are used to describe quantitatively the proximity between two data samples or two clusters. All clustering algorithms are based explicitly or implicitly on similarity/dissimilarity measures between data samples [Jain and Dubes, 1988]. Hence, the high quality of clustering is to obtain high within-cluster similarity and low between-cluster similarity. In addition, when we use the dissimilarity/distance concept, the latter sentence becomes: the high quality of clustering is to obtain low within-cluster dissimilarity and high between-cluster dissimilarity.

Various similarity and dissimilarity measures have been discussed [Anderberg, 1973; Everitt and Dunn, 2001; Gordon, 1999; Sokal and Sneath, 1963]. In the sequel, we present a suite of measures which are commonly used for calculating the similarity of among objects.

Minkowski Metric L_q. It calculates the distance between the two objects $\mathbf{x}$ ans $\mathbf{y}$ by comparing the value of their d features, cf. Equation 2.2.

$$L_q(\mathbf{x}, \mathbf{y}) = \sqrt[q]{\sum_{i=1}^{d} (x_i - y_i)^q}. \tag{2.2}$$

Two important special cases of the Minkowski metric are $q = 1$ and $q = 2$, cf. Equations 2.3 and 2.4:

1. *Manhattan distance or City block distance or L_1 norm*:

$$L_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} |x_i - y_i| . \tag{2.3}$$

2. *Euclidean distance or L_2 norm* :

$$L_2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=2}^{d} (x_i - y_i)^2}. \tag{2.4}$$

Kullback-Leibler Divergence. This divergence (KL) is defined in Equation 2.5; KL is a measure from information theory which determines the inefficiency of assuming a model distribution given the true distribution [Cover and Thomas, 1991]. It is generally used for $\mathbf{x}$ and $\mathbf{y}$ representing probability mass functions

$$KL(\mathbf{x} \parallel \mathbf{y}) = \sum_{i=1}^{d} x_i \times \log \frac{x_i}{y_i}. \tag{2.5}$$

The Kullback-Leibler divergence is not defined in case $y_i = 0$ so the probability distributions need to be smoothed by performing one of the two variants of KL, information radius or skew divergence. Both variants can tolerate zero values in the distribution, because they work with a weighted average of the two distributions compared. Lee [2001] has shown that the skew divergence is an effective measure for distributional similarity in NLP.

Kendalls τ coefficient [Kendall and Gibbons, 1990]. This coefficient compares all feature pairs of the two objects $\mathbf{x}$ and $\mathbf{y}$ in order to calculate their distance. if $\langle x_i, y_i \rangle$ and $\langle x_j, y_j \rangle$ are two pairs of the features i and j for the objects $\mathbf{x}$ and $\mathbf{y}$, the pairs are concordant if $x_i > x_j$ and $y_i > y_j$. If the distributions of the $\mathbf{x}$ and $\mathbf{y}$ are similar, a large number of concordances f_c is expected, otherwise a large number of discordances f_d is expected. τ is defined in Equation 2.6, with p_c the probability of concordances and p_d the probability of discordances; τ ranges from -1 to 1. The τ coefficient can be applied to frequency and probability values. Hatzivassiloglou and McKeown [1993] used τ to measure the similarity between adjectives.

$$\tau(\mathbf{x}, \mathbf{y}) = \frac{f_c}{f_c + f_d} - \frac{f_d}{f_c + f_d} = p_c - p_d. \tag{2.6}$$

Cosine similarity. This similarity allows to measure the similarity between two objects $\mathbf{x}$ and $\mathbf{y}$ by calculating the *cosine of the angle* between their feature vectors. For positive feature values, the cosine lies between 0 and 1. The cosine measure can be applied to frequency, probability and binary values.

$$cos(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{d} x_i \times y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \times \sqrt{\sum_{i=1}^{d} y_i^2}}. \tag{2.7}$$

2.1.4 Popular Clustering algorithms

Many algorithms have been proposed to perform the data clustering task. No clustering technique is universally applicable, and different techniques are in favour for different clustering purposes. So an understanding of both the clustering problem and the clustering technique is required to apply a suitable method to a given problem. The choice of a clustering algorithm determines the setting of the parameters. The clustering problems can be categorized into two main types: fuzzy clustering and hard clustering. In fuzzy clustering, data points can belong to more than one cluster with probabilities between 0 and 1 [Bezdek and Pal, 1992; Karaboga and Ozturk, 2010] which indicate the strength of the relationships between the data points and a particular cluster. One of the most popular fuzzy clustering algorithms is fuzzy c-mean algorithm [Bezdek, 1981; Hoppner et al., 1999]. In hard clustering, data points are divided into distinct clusters, where each data point can belong to one and only one cluster. The hard clustering is divided into hierarchical and partitional algorithms.

2.1.4.1 Hierarchical clustering

Hierarchical clustering aims to obtain a dendrogram of clusters that shows how the clusters are related to each other. The clustering result of the data objects can be obtained by cutting the obtained dendrogram at the suitable level. These methods proceed either by iteratively merging small clusters into larger ones (agglomerative algorithms) or by splitting large clusters (divisive algorithms) (figure 2.1). Based on these, it can be classified into the following categories:

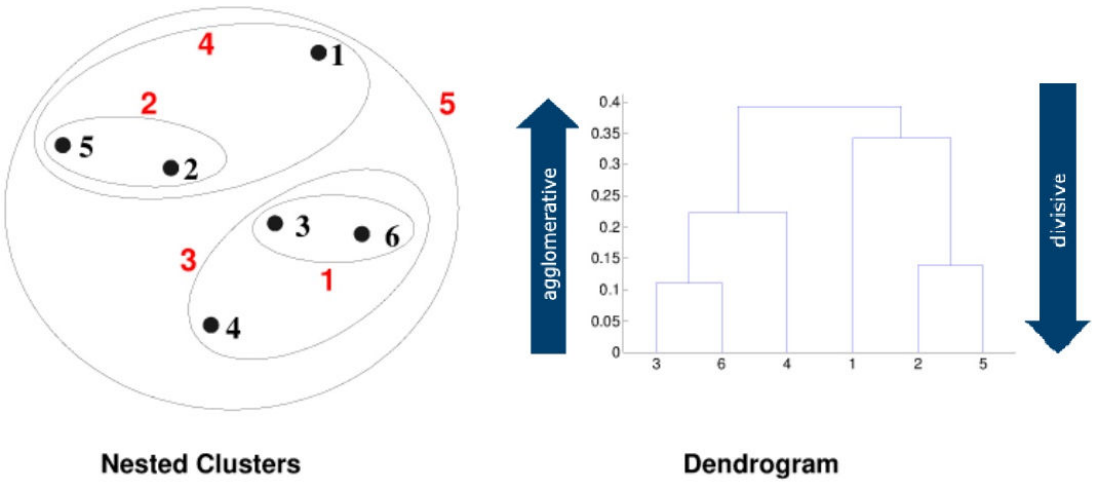


Figure 2.1 – Hierarchical clustering algorithms.

Agglomerative Algorithm: The family of agglomerative algorithms [Jain et al., 1999] is arguably the most popular example of hierarchical clustering algorithms. 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 until all the data are merged into a single cluster. According to the definition of the similarity measure between two clusters, several agglomerative variants exist, but the most commonly used are the single linkage criterion [Sibson, 1973], the complete linkage criterion [Sorensen, 1948] and the average linkage criterion [Sokal and Michener, 1958]. When the objects are described by continuous variables, the Ward's method is frequently used.

Divisive Algorithm: It aims to create 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 each data point is into its own singleton cluster. Note that stop conditions can be used, and the division into clusters is governed by whether or not a particular property is satisfied.

2.1.4.2 Partitional Clustering

Partitional clustering attempts to obtain a partition which minimizing the within-cluster sum of squares or maximizing the between-cluster sum of squares. 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. In the real clustering applications, partitional clustering techniques have been considered more appropriate for large data sets than hierarchical techniques in which the construction of the dendrogram is computationally expensive. However, the determination of the number of clusters is one of the most problematic issues in partitional clustering methods. The partitional algorithms often use a certain objective function and produce the desired clusters by optimizing this objective function [Hansen and Jaumard, 1997]. The most popular Partitional clustering algorithm is k-means [McQueen, 1967]. Due to its simplicity and good performance, k-means is one of the most widely-used clustering algorithms. It is an iterative algorithm, whose goal is distributing the data in clusters such that the within-cluster sum of squares $\mathcal{W}$ is minimised, which is defined as

$$\mathcal{W}(\mathcal{P}) = \sum_{h=1}^{k} \sum_{\mathbf{x} \in \mathcal{P}_h} ||\mathbf{x} - \overline{\mathcal{P}}_h||^2 \tag{2.8}$$

where $\mathcal{P}$ is a partition into k clusters $\{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$, and $\overline{\mathcal{P}}_h$ is the centroid of cluster h.

The pseudo-code for k-means is shown in Algorithm 1.

Algorithm 1: The k-means algorithm .

Input: - X, the data matrix to cluster;
- k, the number of clusters.
Output: $\mathcal{P}^* = \{\mathcal{P}^*{}_1, \ldots, \mathcal{P}^*_k\}$, the partition ($k$ clusters) of data
Initialize: the centroids $\overline{\mathcal{P}}_h$
while *convergence is not attained* **do**
  for $x \in X$ **do**
    $h \leftarrow \texttt{argmin}_{m \in 1 \ldots k} ||\mathbf{x} - \overline{\mathcal{P}}_h||^2$
    $\texttt{Assign}(x, \mathcal{P}_h)$
  end
  for $\mathcal{P}_h \in \mathcal{P}$ **do**
    $\texttt{Recalculate-Centroid}\,(\overline{\mathcal{P}}_h)$
  end
end

One of the most important problems of k-means is its dependency on the seeds which have been chosen in the initialisation phase (the centroids initial values). A good initialization leads to a good solution while a bad one can lead to to an unsuitable partition. In order to offer better algorithms than k-means, many variants are proposed. Further, this algorithm has inspired many other algorithms that can be categorized in various families. Without being exhaustive, hereafter some different types of clustering algorithms.

Density-based Clustering. 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. Among the well-known methods, there are *Denclue* which tries to analytically model the overall density around a point [Hinneburg and Keim, 1998], and *WaveCluster* which uses wavelet-transform to find high density regions [Sheikholesami et al., 1998]. Note that, density-based methods typically have difficulty scaling up to very high dimensional data ($>$ 10000), which are common in text mining for instance.

Model-based Clustering. The mixture approach assumes that each cluster is generated according to a distribution with some specific parameters. The approach relies on the maximisation of the likelihood. The estimation of parameters can be performed by the Expectation Maximization (EM) algorithm [Dempster et al., 1977]. Many variants of EM were proposed to overcome some drawbacks of EM. The flexibility of the mixture model makes this approach very powerful; see for instance [McLachlan and Peel, 2004]. Several criteria used in clustering context, such as the within-cluster sum squares, are associated to a restricted gaussian mixture model. Nevertheless, the high dimensionality is a challenge for this type of approach.

Graph-theoretic Clustering. Another type of clustering algorithms is based on the construction of similarity graphs in which a given set of data points is transformed into vertices and edges. The constructed graph can be used to obtain a single highly connected graph that is then partitioned by edge cutting to obtain sub graphs [Santos et al., 2008; Shi and Malik, 2000]. Basically, the kinds of graphs are ε-neighborhood, k-nearest neighbor and fully connected graph [Barbakh and Fyfe, 2008; Luxburg, 2007].

Spectral clustering has many fundamental advantages compared to traditional model-based clustering algorithms such as k-means. Results obtained by spectral clustering often outperform the traditional approaches, and it is very simple to implement and can be solved by computing eigenvalue/eigenvector problem. However, spectral clustering suffers from heavily computations. The core of the spectral clustering algorithms is to use the properties of eigenvectors of Laplacian matrix for performing graph partitioning [Fielder, 1975; Luxburg, 2007; Ng et al., 2001; Santos et al., 2008; Verma and Meila, 2003].

In order to address the computational difficulties and to improve the results of spectral clustering, Chen et al. [2011] proposed sparsification and Nystrom approaches. This latter is a technique for finding an approximate eigendecomposition. Spectral clustering using Nystrom method requires less computation and does not need the prespecified number of nearest neighbors as in sparsification method. The spectral clustering using Nystrom method uses randomly sample data points from the data set to approximate the similarity matrix of all data points in the data set. Then it finds the first k eigenvectors of the normalized Laplacian matrix of the Nystrom method and performs k-means to cluster data set.

Matrix Factorization Based Clustering. Recently there has been significant development in the use of non-negative matrix factorization (NMF) methods for various clustering tasks. NMF factorizes an input nonnegative matrix into two nonnegative matrices of lower rank. Although NMF can be used for conventional data analysis, the recent overwhelming interest in NMF is due to the newly discovered ability of NMF to solve challenging data mining and machine learning problems. In particular, NMF with the sum of squared error cost function is equivalent to a relaxed k-means. In addition, NMF with the I-divergence cost function is equivalent to probabilistic latent semantic indexing, another unsupervised learning method popularly used in text analysis. Many other data mining and machine learning problems can be reformulated as an NMF problem. In section 2.3, we provide a brief review of non-negative matrix factorization methods for clustering and co-clustering. In particular, we outline the theoretical foundations on NMF for clustering, provide an overview of some variants of NMF formulations, and examine several practical issues.

2.1.5 Clustering evaluation

To measure the clustering performance, we use the accuracy, the Normalize Mutual Information [Strehl and Ghosh, 2002] and the Adjusted Rand Index [Hubert and Arabie, 1985].

2.1.5.1 Accuracy

The clustering accuracy noted (Acc) discovers the one-to-one relationship between two partitions and measures the extent to which each cluster contains data points from the corresponding class. It is defined as follows:

$$\text{Acc} = \frac{1}{n}\sum_{i=1}^{n}\delta(\mathcal{C}_i, map(\mathcal{P}_i))$$

where n is the total number of samples, $\mathcal{P}_i$ is the i^{th} obtained cluster and $\mathcal{C}_i$ is the true i^{th} class provided by the data set. $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(\mathcal{P}_i)$ is the permutation mapping function that maps the obtained label $\mathcal{P}_i$ to the equivalent label from the data set. The best mapping can be found by using the Kuhn-Munkres algorithm [Lovász and Plummer, 2009].

2.1.5.2 Normalized Mutual Information

The second measure employed is the Normalized Mutual Information (NMI); it is estimated by

$$\text{NMI} = \frac{\sum_{k,\ell}\frac{n_{k\ell}}{n}\log\frac{nn_{k\ell}}{n_k\hat{n}_\ell}}{\sqrt{(\sum_k n_k\log\frac{n_k}{n})(\sum_\ell \hat{n}_\ell\log\frac{\hat{n}_\ell}{n})}}$$

where n_k denotes the number of data contained in cluster $\mathcal{P}_k(1 \leq k \leq K)$, $\hat{n}_\ell$ is the number of data belonging to the class $\mathcal{C}_\ell(1 \leq \ell \leq K)$, and $n_{k\ell}$ denotes the number of data that are in the intersection between cluster $\mathcal{P}_k$ and class $\mathcal{C}_\ell$.

2.1.5.3 Adjusted Rand Index

The last measure *Adjusted Rand Index* (ARI) is a measure of the similarity between two data clustering partitions. From a mathematical standpoint, the Rand index is related to the accuracy. The adjusted form of the Rand Index is:

$$\text{ARI} = \frac{\sum_{k,\ell}\binom{n_{k\ell}}{2} - \left[\sum_k\binom{n_k}{2}\sum_\ell\binom{\hat{n}_\ell}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_k\binom{n_k}{2}+\sum_\ell\binom{\hat{n}_\ell}{2}\right] - \left[\sum_k\binom{n_k}{2}\sum_\ell\binom{\hat{n}_\ell}{2}\right]/\binom{n}{2}}.$$

2.1.5.4 Silhouette Score

Silhouette index (noted SIL) is a very well-known clustering evaluation approach that introduces clustering quality scores for each individual point and calculates the final quality index as an average of the point-wise quality estimates [Rousseeuw, 1987]. Each point-wise estimate for a point $\mathbf{x}_{.p} \in \mathcal{P}_i$ is derived from two quantities: $a_{i,p}$ and $b_{i,p}$ which correspond to the average distance to other points within the same cluster and the minimal average distance to points from a different cluster, respectively. Formally,

$$a_{i,p} = \frac{1}{|\mathcal{P}_i| - 1} \sum_{\mathbf{x}_{.q} \in \mathcal{P}_i, q \neq p} \|\mathbf{x}_{.q} - \mathbf{x}_{.p}\| \quad \text{and} \quad b_{i,p} = \min_{j=1..k, j \neq i} \frac{1}{|\mathcal{P}_j|} \sum_{\mathbf{x}_{.q} \in \mathcal{P}_j} \|\mathbf{x}_{.q} - \mathbf{x}_{.p}\|$$

$$\text{For each data point } \mathbf{x}_{.p} : \quad SIL(\mathbf{x}_{.p}) = \frac{a_{i,p} - b_{i,p}}{\max(a_{i,p}, b_{i,p})}$$

$$\text{The Silhouette Score} : \quad SIL = \frac{1}{n} \sum_{p=1}^{n} SIL(\mathbf{x}_{.p})$$

2.2 Co-clustering

Co-clustering consists in performing clustering simultaneously on the sets of samples and features. Because of its potential benefit of discovering latent local patterns, in recent years, has recently received a lot of attention in varied practical applications. In *Text mining* to identify document and word clusters from a bag-of-words model represented in a vector space in the form of word-by-document matrix [Dhillon, 2001b; Dhillon et al., 2002, 2003b; Gao et al., 2005a; Takamura and Matsumoto, 2003]. In *Web mining* to extract subsets of user sessions and Web pageviews to construct a variety of co-clusters [Charrad et al., 2009]. In *Bioinformatics* to identify groups of similar genes and similar conditions based on their expression levels [Madeira and Oliveira, 2004]. In *Natural language processing*, to construct new features in a more compact but highly informative representation from co-cluster centroids [Freitag, 2004; Li and Abe, 1998]. In *Image retrieval* to perform better retrieval performance [Dhillon, 2001b]. In *Video content recognition and Auditory scene categorization* to detect unusual activity in a large video set using many simple features [Cai et al., 2005; Guan et al., 2005; Qie, 2004]. In *Users and movies in recommender systems* to simultaneously obtain user and item neighborhoods via the co-clustering and generate predictions based on the average ratings of the co-clusters; see for instance [Banerjee et al., 2007].

Co-clustering is not a recent topic. Hartigan [1972] proposed the direct clustering approach (*Block Clustering*) where the initial data matrix is divided into several sub-matrices which correspond to blocks. The division of a block depends on the variance of its values. Indeed, the lower the variance and the more constant are the values in the block. The partition's quality is estimated by the sum of the variances of the blocks. Hartigan [1975] proposed two other algorithms of bi-clustering: the first algorithm (*One-Way Splitting*) focuses mainly on the examples, by trying to build a partition with features having an intra-class variance superior to a certain threshold to cut the associated class. As any technique of direct clustering, a minimal threshold yields a significant number of classes of low density and vice-versa. The second algorithm (*Two-Way Splitting*) proceeds by successive divisions of lines and columns. Since the work of Hartigan, several papers have appeared often referring to biclustering, co-clustering, two-mode clustering. In the sequel, we briefly describe the popular methods throughout the last three decades.

Govaert [1983] proposed three algorithms of co-clustering, *Croeuc* for continuous data, *Crobin* for binary data and *Croki2* for contingency tables. The three algorithms are based on an alternated *nuées dynamiques* of Diday [1971]. With the advent of bioinformatics appeared the interest of bi-clustering. Cheng and Church [2000] proposed $\delta-clusters$, another greedy search based on the creation of bi-clusters by adding lines (or columns) so as to maximize a local gain. This approach uses the Residual Mean Square as a measure of similarity. Inspired by $\delta-clusters$, Yang et al. [2003] proposed *FLOC* (*Flexible Overlapped Clusters*). They introduced an additional function relative to the processing of missing data and the overlapping. Moreover, Tanay et al. [2002] proposed a graph-based method called *Samba*, which enumerates exhaustively all the cliques modeling the possible bi-clusters in a bipartite graph from the data matrix. Other approaches in this field exist; see for instance [Klugar et al., 2003; Lazzaroni and Owen, 2002; Pensa et al., 2010; Pensa and Boulicaut, 2008].

Because the interest of text-mining, probabilistic approaches are appeared. In Hoffman and Puzicha [1999], the authors 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 [Kemp et al., 2006], Mixed Membership Blockmodel [Airoldi et al., 2008] and Bayesian co-clustering [Shan and Banerjee, 2008], were introduced recently using different inference engines. Furthermore, Long et al. [2007] proposed the Mixed Membership Relational Clustering (MMRC) model in which parametric soft clustering results are derived

using Expectation Maximization (EM) for a large number of exponential family distributions. Note that the lalent block models developed by Govaert and Nadif [2003, 2008, 2010, 2014] can be used in this context.

Placing the text-mining area in graph context, Dhillon [2001b] proposed the spectral learning, such as Bipartite Spectral Graph Partitioning (BSGP) 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. In the same manner, Gao et al. [2005b] 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. 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. [2006] 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.

Initially applied to the image and video the matrix factorisation is increasingly popular in the field of clustering and co-clustering. As in the thesis we focus on both topics under the matrix factorisation, in the sequel we describe this approach.

2.3 Matrix Factorization Based Clustering and Co-clustering

The aim of Matrix factorisation is to factorise a given matrix into two smaller matrices of lower rank, so that their product reconstructs the original matrix. k-means can also be seen as a matrix factorisation method, where the cluster centroids are stored in one matrix and the cluster indicators in the other. The use of matrix factorisation as a standalone method became popular when several experiments consistently showed that Non-negative Matrix Factorisation (NMF) gives better clustering results than k-means.

2.3.1 NMF Formulations

In [Lee and Seung, 1999], the authors formulated NMF as a model based on minimizing a cost function based on a Poisson likelihood. They also introduced two further cost functions based on the Frobenius norm and I-divergence (or generalised KL-divergence) [Lee and Seung, 2001]. Algorithms were based on multiplicative updates became the standard in the field of

Non-negative Matrix Factorisation. Specifically, NMF of a matrix $X \in \mathbb{R}_+^{d\times n}$ was formulated as follows. We wish to compute non-negative (unobserved or latent) matrices $F \in \mathbb{R}_+^{d\times k}$ and $G \in \mathbb{R}_+^{n\times k}$ such that $X \approx FG$. We do this either by minimising the total square error of predictions, also called the Frobenius norm:

$$\left\|X - FG^\top\right\|_F^2 = \sum_{j,i}(X_{ji} - (FG^\top)_{ji})^2. \tag{2.9}$$

or by minimising the I-divergence:

$$J_{basic}(X \parallel FG^\top) = \sum_{j,i}\left(X_{ji}\log\frac{X_{ji}}{(FG^\top)_{ji}} - X_{ji} + (FG^\top)_{ji}\right) \tag{2.10}$$

subject to the constraints $F \geq 0, G \geq 0$.

The following multiplicative updates can be shown to be correct and to converge for the two cost functions (2.9) and (2.10):

$$F_{jk} = F_{jk} \odot \frac{(XG)_{jk}}{(FG^TG)_{jk}} \qquad G_{ik} = G_{ik} \odot \frac{(X^TF)_{ik}}{(GF^TF)_{ik}} \tag{2.11}$$

$$F_{jk} = F_{jk} \odot \frac{\sum_i G_{ik}X_{ji}/(FG^\top)_{ji}}{\sum_i G_{ik}} \qquad G_{ik} = G_{ik} \odot \frac{\sum_j F_{jk}X_{ji}/(FG^\top)_{ji}}{\sum_j F_{jk}}. \tag{2.12}$$

They showed that these multiplicative updates are essentially gradient descent updates where the step size is chosen accordingly. At each iteration, we can normalise the rows of one of the matrices to sum to one. In that case, that matrix gives the cluster indicators, and the other matrix the cluster centroids. An example of NMF is illustrated as follows:

$$\begin{aligned} X &= \begin{bmatrix} 0.185 & 0.326 & 0.761 & 2.799 & 2.375 & 2.970 & 2.585 \\ 0.508 & 0.380 & 0.884 & 2.134 & 2.374 & 2.342 & 2.524 \\ 0.452 & 0.887 & 0.457 & 2.065 & 2.484 & 2.253 & 2.163 \\ 1.486 & 1.843 & 1.858 & 0.566 & 0.103 & 0.417 & 0.269 \\ 1.496 & 1.806 & 1.610 & 0.612 & 0.158 & 0.560 & 0.784 \end{bmatrix} \\ &\approx FG^T = \begin{bmatrix} 1.762 & 0.217 \\ 1.516 & 0.301 \\ 1.439 & 0.310 \\ 0.000 & 1.042 \\ 0.133 & 0.989 \end{bmatrix} \times \begin{bmatrix} 0.000 & 0.000 & 0.052 & 0.474 & 0.507 & 0.520 & 0.494 \\ 0.492 & 0.610 & 0.569 & 0.160 & 0.021 & 0.124 & 0.142 \end{bmatrix} \end{aligned} \tag{2.13}$$

In Equation 2.13, based on the membership indicator G, clearly the first three columns form one cluster, and the last four columns give another.

2.3.2 Variants of NMF

A lot of application papers followed soon after, often compared the popular clustering methods (k-means, SVD, spectral clustering) and showed that NMF achieved better clustering performances. In addition, NMF has been proved to be very useful for applications such as face recognition, text mining and DNA gene expression grouping. Many reviews of NMF exist already. See for example [Berry et al., 2007; Gillis, 2014; Wang and Zhang, 2013].

Local Nonnegative Matrix Factorization. In [Li et al., 2001], the authors proposed the Local Nonnegative Matrix Factorization (LNMF). Using the I-divergence as cost function, they added constraints that aim to minimise the elements in one matrix, and maximise the diagonal elements of the other. In simple terms, LNMF imposes the sparseness constraints on G and locally constraints on F based on the following three considerations: 1) Maximizing the sparseness in G; 2) Maximizing the expressiveness of F; 3) Maximizing the column orthogonality of F. The objective function in the model of LNMF can take the following form:

$$J_{basic}(X \parallel FG^{\top}) + \alpha \sum_{i,j} (F^T F)_{ij} - \beta \sum_{i} (G^T G)_{ii}.$$

More recently, Xu et al. [2003] use the same model as [Li et al., 2001] but to make the solution unique, they required that the Euclidean length of the column vector in matrix F is one. They then used this to cluster documents, it is shown that NMF outperforms spectral methods, achieving higher clustering accuracy, less computation cost and more intuitive interpretability.

$$F_{jk} = \frac{F_{jk}}{\sqrt{\sum_{j'} F_{j'k}^2}} \qquad G_{ik} = G_{ik} \sqrt{\sum_{j'} F_{j'k}^2}.$$

Sparse Nonnegative Matrix Factorization. First of all, Hoyer [2002] proposed the Nonnegative Sparse Coding (NSC) which only maximizes the sparseness in G. The objective function to be minimized can be written as:

$$\left\|X - FG^T\right\|_F^2 + \lambda \sum_{i,j} G_{ij}.$$

Since the objective function in the above model NSC can be separated into a least squares error term $\left\|X - FG^T\right\|_F^2$ and an additional penalty term $\sum_{i,j} G_{ij}$, Liu et al. [2003] replaced the least squares error term with the KL-divergence to get the following new objective function:

$$J_{basic}(X \parallel FG^{\top}) + \lambda \sum_{i,j} G_{ij}.$$

Furthermore, Pauca et al. [2004] also used NMF for text mining, but added the L2 norm on rows in G to enforce sparsity. The updates for F are the multiplicative updates for minimising $\left\|X - FG^\top\right\|_F^2$, but the updates for G are given by the minimisation problem

$$\min_{G_i} \|X_{:i} - FG_i\|_F^2 \; + \; \lambda \|G_i\|_F^2 \, .$$

This idea was used by Gao et al. [2005b] to classify cancer types from gene expression profiles and by Shahnaz et al. [2006] for document clustering and topic detection. Recently, Pauca et al. [2006] used NMF for spectral data, but add new sparsity constraints that are simply the Frobenius norm on F and G.

$$\left\|X - FG^\top\right\|_F^2 \; + \; \alpha \|F\|_F^2 \; + \; \beta \|G\|_F^2 \, .$$

Semi Nonnegative Matrix Factorization. Semi-NMF [Wang et al., 2008a] is designed for the data matrix X that has mixed signs. In semi-NMF, G is restricted to be nonnegative while the other factor matrix F can have mixed signs, i.e., semi-NMF can take the following form

$$X_\pm \approx F_\pm G_\pm^T$$

This model is motivated from the perspective of data clustering. When clustering the columns of data matrix X, the columns of F can be seen as the cluster centroids and the rows of G denote the cluster indicators, i.e., the column j of X belongs to cluster k if $k = \arg\max_p \{G_{jp}\}$. Hence the nonnegative constraint on F can be relaxed such that the approximation FG^T is tighter and the results are more interpretable.

Wang et al. [2008a] have also considered semi-NMF (constraining G to be non-negative but X and F can be negative) and introduced constraint matrices Θ indicating rewards (if negative) or penalties (if positive) for clustering two data points together, making the cost function.

$$\left\|X - FG^\top\right\|_F^2 \; + \; \mathrm{Tr}(G^\top \Theta G).$$

Nonnegative Matrix Factorization on manifold. In [Kim and Park, 2007], the authors introduces a model minimising the Frobenius norm, adding the Frobenius norm on F and the L_1 norm on rows in G . They give an algorithm based on Alternating Least Squares. The proposed method was used for gene expression data analysis and cancer-class discovery.

$$\left\|X - FG^\top\right\|_F^2 \; + \; \alpha \|F\|_F^2 \; + \; \beta \sum_i \|G_i\|_1^2 \, .$$

Next, Shen and Si [2010] started from the model of Kim and Park [2007] and extended it by adding multiple manifolds, capturing intrinsic geometrical structure of data. They did this by first obtaining a manifold matrix S that captures geometrical structure, and then finding a G such that $G \approx GS$. We add the term $\sum_i \|G_i - SG_i\|$ extra penalisation. Recently, Huang et al. [2014] extended the model of [Kong et al., 2011] by adding manifold regularisation as follows: we construct a graph in which, an edge exists between two data points $\mathbf{x}_i$, $\mathbf{x}_j$ if $\mathbf{x}_i$ is one of the K nearest neighbors of $\mathbf{x}_j$, or vice versa. Let W be the affinity matrix of the graph. then they compute the graph Laplacian L. The objective function take the following form:

$$\left\|X - FG^\top\right\|_{2,1} + \alpha \mathrm{Tr}(G^\top LG).$$

2.3.3 Relations among NMF and popular clustering models

Ding et al. [2005] studied the problem of symmetric NMF ($d = n$), they proved that the decomposition into $X \approx FF^\top$ is equivalent to kernel k-means clustering and (Laplacian-based) spectral clustering. They also introduced multiplicative updates for "weighted NMF" which is effectively symmetric non-negative matrix tri-factorisation (see later section), and discussed why this is a better model than $X \approx FF^\top$. Next, Ding et al. [2006b] proved that orthogonal NMF is equivalent to k-means clustering. Finally, Gaussier and Goutte [2005] and Ding et al. [2006a] showed that probabilistic latent semantic indexing (PLSI) and NMF (with L_1 normalization) optimize the same objective function, although PLSI and NMF are different algorithms as verified by experiments.

NMF and k-means Clustering. Theoretically, NMF is inherently related to (kernel) k-means clustering [Li and Ding, 2006]. Indeed, NMF has clustering capabilities which is generally better than k-means. In k-means, an exact orthogonality of columns of cluster indicator G implies that each row of G can have only one nonzero element, which implies that each data object belongs only to one cluster. While in NMF, the near-orthogonality condition of 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. There are three popular objective functions in spectral clustering: the Ratio Cut [Hagen and Kahng, 1992], the Normalized Cut [Shi and Malik, 2000], and the MinMax Cut [Ding et al., 2001]. Compared to the spectral graph model, NMF does not require the derived cluster indicator space G to be orthogonal, and it guarantees that each data takes only non-negative values; these two characteristics make NMF interesting.

2.3.4 Non-negative Matrix Tri-Factorisation

Unlike NMF which decompose a matrix into two matrices, Non-negative Matrix Tri-Factorisation (NMTF) decompose it into three matrices,

$$X \approx FSG^\top$$

If we constrain the problem to be non-negative, we have $F \in \mathbb{R}_+^{d\times\ell}$, $G \in \mathbb{R}_+^{n\times k}$ and $S \in \mathbb{R}_+^{\ell\times k}$ where k and ℓ are respectively the number of sample and feature clusters. We now have F indicating the clustering of data features, and simultaneously G the clustering of data samples. S relates data features and data samples clusters. Li and Ding [2006] includes a good discussion as to why matrix tri-factorisation is interesting for detecting biclusters (co-clusters). Note that, NMTF for symmetrical matrices was introduced by Ding et al. [2005] who called it "weighted NMF", and involved the following decomposition $FSF^\top$. The aims of NMF and NMTF are illustrated in Figure 2.2.

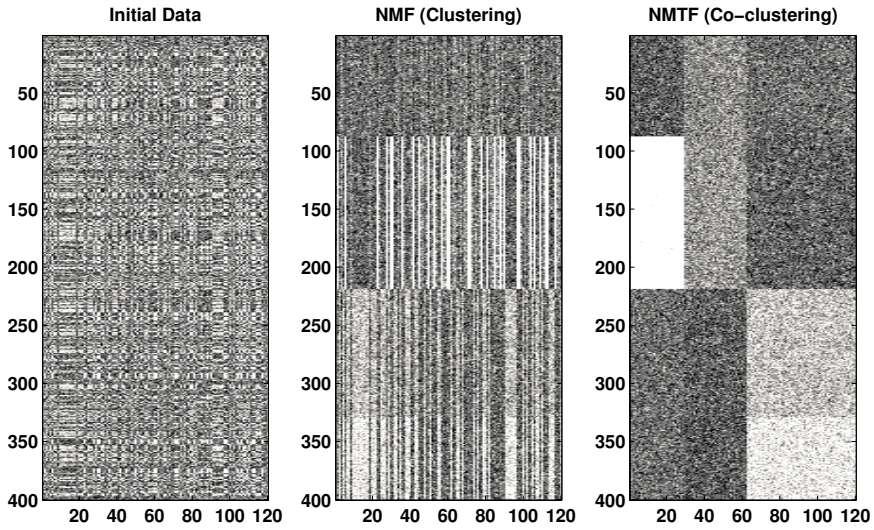


Figure 2.2 – Illustration clustering vs co-clustering

In a similar way to NMTF, Singular Value Decomposition (SVD) decompose a matrix into three matrices, $X = U\Sigma V^*$, for $U \in \mathbb{R}_+^{d\times d}$, $V \in \mathbb{R}_+^{n\times n}$ and $\Sigma \in \mathbb{R}_+^{d\times n}$. But here we have U and V both as orthogonal matrices ($UU^\top = I$, $VV^\top = I$), and Σ is a diagonal matrix where its diagonal entries are non-negative values giving the eigenvalues of X. In contrast, with NMTF we allow the off-diagonal entries to be non-zero so that rows and columns can be assigned to different clusters. Furthermore, ℓ and k can be chosen to be much lower than d and n, to provide a low-rank approximation of the data set.

Nonnegative Matrix Tri-Factorization on manifold. Ding et al. [2006b] introduced the more general case of NMTF $X \approx FSG^\top$, note that without any constraint this is equivalent to NMF, but with orthogonality constraints on F, G this gives a very different solution. The papers that followed often added some forms of regularisation, and applied their methods to clustering data sets. Approaches for manifold regularisation include orthogonality constraint ($FF^\top = I$), constraint matrices (Θ measures dissimilarity between objects or features; then add $Tr(F^\top \Theta F)$), and Laplacian graphs (we again add $Tr(F^\top LF)$).

Furthermore, Gu and Zhou [2009] proposed the Dual Regularized Co-Clustering (DRCC) method, based on graph-regularized semi-NMTF models. The DRCC algorithm inherits the advantages of NMTF and, in addition, takes into account the manifold structures in both data and feature spaces (added penalisation based on Laplacian graphs both over the rows/data points and columns/features). However, the high computational complexity of DRCC usually makes it unsuitable for large-scale problems.

To reduce the computational complexity of DRCC, Wang et al. [2011b] introduced a faster algorithm for semi-NMTF by constraining the F and G to be in the cluster indicator space: each row vector has to contain exactly one 1-entry, and for the rest only 0's. Laplacian graph penalisation is added with reducing the computational cost of the eigendecomposition of the graph Laplacian. The proposed algorithms are referres as to Fast NMTF (FNMTF) and Locally Preserved FNMTF (LPFNMTF).

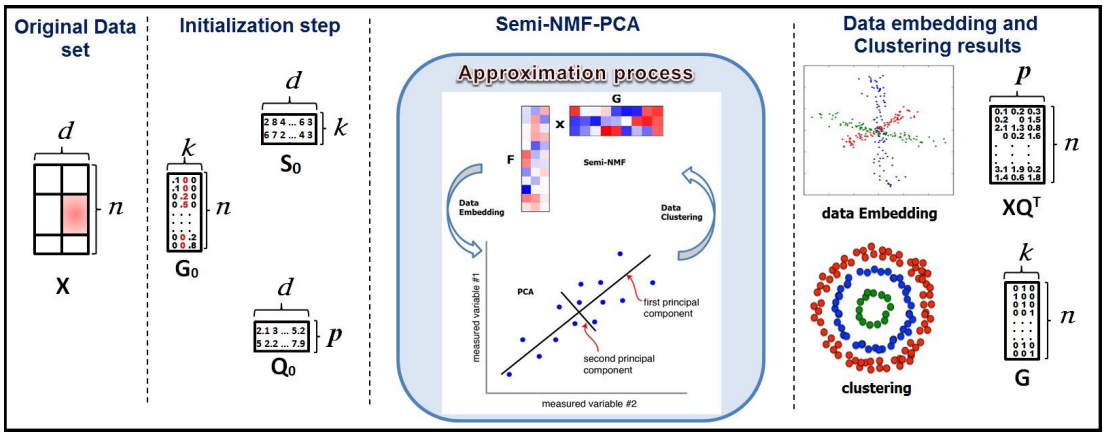
Semi-Supervised Co-Clustering. Semi-supervised co-clustering aims to incorporate the priory knowledge in co-clustering. The partial knowledge can be formulated as *must-link* and *cannot-link* constraints on both data samples and data features. Let A^x contain the *must-link* pairs for samples (A^y for features), and B^x contain the *cannot-link* pairs for samples (B^y for features). Then, the semi-supervised co-clustering problem can be formulated as

$$\min_{F \geqslant 0, G \geqslant 0} \left\| X - FSG^T \right\|^2 + \mathrm{Tr}\left[\alpha F^T (A^x - B^x) F + \beta G^T (A^y - B^y) G\right].$$

where α, β are parameters to control the effects of different types of constraints [Wang et al., 2008b]. On the other hand, Li et al. [2008] proposed several constrained nonnegative tri-factorization knowledge transformation method to use the partial knowledge (such as instance-level constraints and partial class label information) from one type of objects (e.g., terms) to improve the clustering of another type of objects (e.g., documents). Their models bring together semi-supervised clustering/co-clustering and learn from labeled features [Sindhwani et al., 2008]. Another semi-supervised co-clustering method has been proposed in [Chen et al., 2010] using symmetric NMTF.

Chapter 3

Simultaneous Data Embedding and Clustering

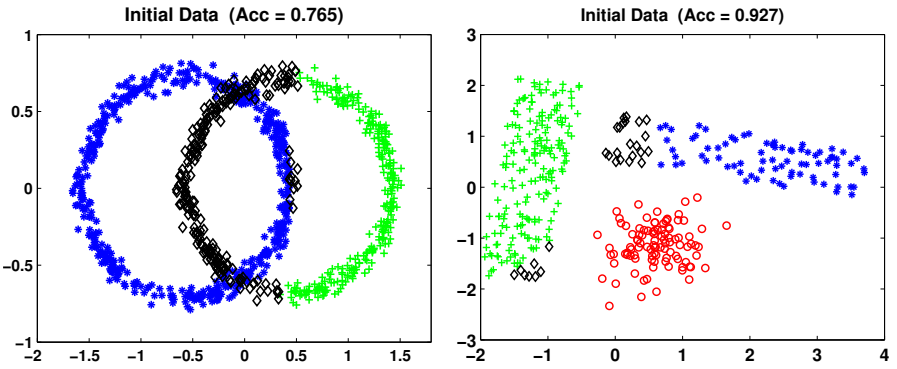


3.1 Introduction

The aim of cluster analysis is the discovery of a finite number of homogeneous classes from data. These classes can be assumed to lie in a low-dimensional subspace of data. Generally when users aim to cluster data, they seek to visualize the clusters in a reduced dimension space. This procedure can be carried out into two simple steps:

- Step 1. PCA is performed, and the first few components are retained.
- Step 2. k-means clustering is performed on these first principal components.

This two-step procedure is called tandem clustering by Arabie and Hubert [Arabie and Hubert, 1994] and has been discouraged by several authors [Arabie and Hubert, 1994; Vichi and Kiers, 2001]. Because the first few principal components of PCA do not necessarily reflect the cluster structure in data, the appropriate clustering result may not be obtained by using the tandem clustering approach. In order to illustrate the weakness of tandem clustering to preserve the initial topology and its capability to separate classes, we used Lsun and Chainlink FCPS data sets. Applying the tandem clustering on these data sets, in Fig. 3.1 we note that the 2D representation of the obtained clusters does not reflect the real cluster structure.



(a) Chainlink data set. (b) Lsun data set.

Figure 3.1 – Data representation of the FCPS data sets: Lsun and Chainlink. For Chainlink, the data points are projected into the factorial plane spawned by the two first components obtained by PCA. Black points represent the misclassified objects obtained by k-means and Acc denotes the accuracy which is the percentage of objects well classified.

In this chapter we propose a novel approach to finding an optimal subspace of multi-dimensional variables for identifying a partition of the set of objects. The use of a low-dimensional representation can be of help in providing simpler and more interpretable solutions. We show that by doing so, our model is able to learn low-dimensional representations that are better suited for clustering.

Cluster analysis is often carried out in combination with dimension reduction. For instance, Semi-*Non-negative Matrix Factorization* (SemiNMF) that learns a low-dimensional representation of a data set lends itself to a clustering interpretation. Indeed, PCA and SemiNMF can be integrated into a single framework of simultaneous data clustering and visualization. Specifically:

- Unlike to known methods that combine the objective function of PCA and the objective function of k-means separately, we propose a new single framework to perform SemiNMF via PCA for dimension reduction and data clustering.
- We show that the objective learning of SemiNMF-PCA can be decomposed into two terms, the first one is the objective function of PCA and the second is the SemiNMF criterion in a low-dimensional space. This allows a better approximation of data reduction by a clustering solution.
- We developed an efficient Fast SemiNMF-PCA based procedure to find simultaneously the optimal partition and reduced features space.
- We further developed our method to incorporate manifold information and proposed the graph regularized Fast SemiNMF-PCA method.

The rest of chapter is organized as follows. Section 2 introduces the clustering problem and the dimension reduction in factorization framework. Section 3 provides a sound SemiNMF-PCA framework for clustering. Section 4 focuses on some details concerning the proposed Graph Regularized Fast SemiNMF-PCA algorithm and on the connection between them and other state of the art clustering methods. Section 5 is devoted to numerical experiments. Finally, the conclusion summarizes the advantages of our contribution.

3.2 SemiNMF via Principal component analysis (SemiNMF-PCA)

Let $X = (x_{ij})$ be a $(n \times d)$ positive data matrix; we assume that X is provided by a collection of n data row vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, each with d features.

SemiNMF [Ding et al., 2010]: SemiNMF relaxes the non-negativity constraint of NMF and allows the data matrix X and the matrix S to have mixed signs, while it restricts only the data factor matrix G so that it comprises of strictly non-negative components. It thus approximates the following factorization:

$$\min_{G,S} \left\| X - GS^\top \right\|^2 \quad \text{s.t.} \quad G \geq 0. \tag{3.1}$$

This is motivated by a clustering perspective. If we view S as the cluster centroids, then G can be viewed as the cluster indicators for each data point. In fact, if we had a matrix G that was not only non-negative but also orthogonal, then every column vector would have only one positive element, making SemiNMF equivalent to k-means. SemiNMF, which does not impose an orthogonality constraint on its features matrix, can be seen as a soft clustering method where the features matrix describes the compatibility of each component with a cluster centroid and a base in S.

Principal Component Analysis [Collins et al., 2001; Jolliffe, 2002]: PCA enabled us to find the optimal low-dimensional subspace defined by the principal directions Q. The projected data points in the new subspace are U. PCA finds U and Q by minimizing

$$\min_{U,Q} \left\| X - UQ^\top \right\|^2. \tag{3.2}$$

Solving for the optimal U while fixing Q is given by $U = XQ$. Plugging U in equation 3.2, holds

$$\min_{Q} \left\| X - XQQ^\top \right\|^2. \tag{3.3}$$

In addition, PCA relates closely to k-means clustering naturally [Ding and He, 2004]. The principal components U are actually the continuous solution of the membership indicators in the k-means clustering method. This provides a motivation to relate PCA to Laplacian embedding whose primary purpose is clustering. Next, we see how we can combine simultaneously both clustering and dimension reduction approaches.

3.2.1 SemiNMF-PCA objective function

Let k be the number of clusters and p the number of components to which the features are reduced. SemiNMF-PCA clustering is defined as the minimizing problem of the following criterion:

$$\min_{G,S,Q} \left\| X - GSQ^\top \right\|^2 \quad \text{s.t.} \quad G \geq 0, \quad Q^\top Q = I. \tag{3.4}$$

where $||.||$ denotes the Frobenius norm.

The binary matrix $G = (g_{ij})$ of size $(n \times k)$ specifies cluster membership for each object, $Q = (q_{ij})$ of size $(p \times d)$ is a column-wise orthonormal loading matrix, $S = (s_{k'j})$ of size $(k \times d)$ is a centroid matrix while $\mathbf{s}_{k'}$ is a centroid of the $(k')^{th}$ cluster for each $k' = 1, \ldots, k$.

To solve the problem (3.4), we rely on the following proposition

Proposition 1. *Given $G \geq 0$ and $Q^\top Q = I$, the objective function of SemiNMF-PCA can be decomposed into two terms:*

$$\left\|X - GSQ^\top\right\|^2 = \left\|X - XQQ^\top\right\|^2 + \|XQ - GS\|^2 \tag{3.5}$$

Proof. We first expand the matrix norm of the left term of Eq. (3.5)

$$\left\|X - GSQ^\top\right\|^2 = \|X\|^2 + \left\|GSQ^\top\right\|^2 - 2Tr(X^\top GSQ^\top) \tag{3.6}$$

In a similar way, from the two terms of the right term of Eq. (3.5), we obtain

$$\begin{aligned} \left\|X - XQQ^\top\right\|^2 &= \|X\|^2 + \left\|XQQ^\top\right\|^2 - 2Tr(XQQ^\top X^\top) \\ &= \|X\|^2 + \left\|XQQ^\top\right\|^2 - 2||XQ||^2 \\ &= ||X||^2 - ||XQ||^2 \quad \text{due to } Q^\top Q = I \end{aligned} \tag{3.7}$$

$$\text{and} \quad \|XQ - GS\|^2 = \|XQ\|^2 + \|GS\|^2 - 2Tr(X^\top GSQ^\top)$$

Due also to $Q^\top Q = I$, we have

$$\|XQ - GS\|^2 = ||XQ||^2 + ||GSQ^\top||^2 - 2Tr(X^\top GSQ^\top) \tag{3.8}$$

Summing the two terms Eq. (3.7) and Eq. (3.8) leads to the left term of Eq. (3.5).

$$\|X\|^2 + \|GS\|^2 - 2Tr(X^\top GSQ^\top) = \left\|X - GSQ^\top\right\|^2 \tag{3.9}$$

□

Using proposition 1, the objective function of SemiNMF-PCA (3.4) can be decomposed into two terms: the first one is the objective function of PCA, and the second is the SemiNMF criterion in a low-dimensional subspace.

3.2.2 Relationships among SemiNMF-PCA and other state-of-the-art clustering methods

Hereafter we establish the relationships among our proposed approach SemiNMF-PCA and some various clustering methods.

3.2.2.1 Relationships with SemiNMF and k-means

The objective function of the SemiNMF method is given by

$$\min_{G\geq 0,F} \left\| X - GF^\top \right\|^2, \tag{3.10}$$

where F is a $(d \times k)$ cluster center matrix. $U\Lambda V^\top$ is expressed as the SVD of F where U is a $(d \times k)$ orthonormal matrix, Λ is a $(k \times k)$ diagonal matrix, and V is a $(k \times k)$ column-wise orthonormal matrix. The function (3.10) can be expressed as

$$\left\| X - GF^\top \right\|^2 = \left\| X - GU\Lambda V^\top \right\|^2. \tag{3.11}$$

Considering $U\Lambda$ as a low-dimensional centroid matrix S and V as a loading matrix (we replace V by Q), the objective function (3.10) is equivalent to that of SemiNMF-PCA (3.4). Thus, SemiNMF-PCA includes SemiNMF where $G \geq 0$, and k-means where $G \in \{0,1\}^{n\times k}$, as particular cases.

3.2.2.2 Relationship with Projective NMF

For fixed values of G and Q, the minimization of the SemiNMF-PCA objective function Eq. (3.4) leads to the optimal S given by

$$S = (G^\top G)^{-1} G^\top XQ.$$

Plugging now S in Eq. (3.4) leads to

$$\left\| X - GSQ^\top \right\|^2 = \left\| X - XQQ^\top \right\|^2 + \left\| XQ - G(G^\top G)^{-1} G^\top XQ \right\|^2. \tag{3.12}$$

Taking $\tilde{G} = G(G^\top G)^{-1/2}$, we obtain

$$\left\| X - GSQ^\top \right\|^2 = \left\| X - XQQ^\top \right\|^2 + \left\| XQ - \tilde{G}\tilde{G}^\top XQ \right\|^2. \tag{3.13}$$

The first term of equation (3.13) is the objective function of PCA and the second is the Semi-Projective NMF criterion in a low-dimensional subspace. The latter is equivalent to the objective function of PNMF [Zhirong and Laaksonen, 2007], it relaxes the non-negativity constraint of PNMF and allows the reduced data matrix XQ to have mixed signs.

3.2.3 Optimization

To solve (3.4), we use an alternated iterative method.

Computation of S First, fixing G and Q, by setting the derivative of the second term in (3.4) with respect to S as 0, we obtain:

$$S = (G^\top G)^{-1} G^\top XQ \tag{3.14}$$

Computation of Q Secondly, fixing G and S, we can rewrite (3.4) as:

$$\min_{Q^\top Q=I} \left\| X - BQ^\top \right\|^2 \text{ where } B = GS. \tag{3.15}$$

To solve (3.15) we rely on the following theorem.

Theorem 1. *Let $X_{n\times d}$ and $B_{n\times k}$ be two matrices. Consider the constrained optimization problem*

$$Q_* = \arg\min_{Q} \left\| X - BQ^\top \right\|^2 \quad s.t \quad Q^\top Q = I \tag{3.16}$$

$$= \arg\max_{Q} Tr(X^\top BQ^\top) \quad s.t \quad Q^\top Q = I \tag{3.17}$$

The solution of Eq. (3.17) comes from the singular value decomposition (SVD) of $X^\top B$. Let $UDV^\top$ be the SVD for $X^\top B$, then $Q_* = UV^\top$.

Remark 1. *Note that the problem in Eq. (3.17) can be considered as a special case of the Orthogonal Procrustes Problem (OPP) [Schonemann, 1966] in which Q is a square orthogonal rotation matrix (i.e $Q^\top Q = QQ^\top = I$).*

Proof. We expand the matrix norm

$$\left\| X - BQ^\top \right\|^2 = Tr(X^\top X) - 2Tr(X^\top BQ^\top) + Tr(QB^\top BQ^\top) \tag{3.18}$$

Since $Q^\top Q = I$, the last term is equal to $Tr(B^\top B)$ and hence the original minimization problem (3.16) is equivalent to the maximization of the middle term, i.e (3.17). With the SVD of $X^\top B = UDV^\top$, this middle term becomes

$$\begin{aligned} Tr(X^\top BQ^\top) &= Tr(UDV^\top Q^\top) \\ &= Tr(UD\hat{Q}^\top) \quad \text{where} \quad \hat{Q} = QV \\ &= Tr(\hat{Q}^\top UD). \end{aligned} \tag{3.19}$$

Denoting $U = [\mathbf{u}_1|\dots|\mathbf{u}_k] \in \mathbb{R}^{d\times k}$, $D = Diag(d_1,\dots,d_k) \in \mathbb{R}_+^{k\times k}$ and $\hat{Q} = [\hat{\mathbf{q}}_1|\dots|\hat{\mathbf{q}}_k] \in \mathbb{R}^{d\times k}$, applying the Cauchy-Shwartz inequality and since $U^\top U = I$, $\hat{Q}^\top\hat{Q} = I$ due to $VV^\top = I$, we have

$$Tr(\hat{Q}^\top UD) \leq \sum_i d_i||\mathbf{u}_i|| \times ||\hat{\mathbf{q}}_i|| = \sum_i d_i = Tr(D).$$

Then the upper bound is clearly attained by setting $\hat{Q} = U$. This leads to $\hat{Q} = QV = U$ and $QVV^\top = UV^\top$. Hence we obtain $Q_* = UV^\top$. □

Due to Theorem 1, applying SVD to $X^\top B$ we obtain the expression of $Q = UV^\top$.

Computation of G Thirdly, we update G by keeping S and Q fixed at the value computed in the above steps, as in [Ding et al., 2006b] we obtain

$$G = G\circ\sqrt{\frac{[XB^\top]^+ + G[BB^\top]^-}{G[BB^\top]^+ + [XB^\top]^-}} \tag{3.20}$$

where $B = SQ^\top$, M^+ and M^- correspond respectively to positive and negative parts of the matrix M given by

$$M_{ik}^+ = \frac{1}{2}(|M_{ik}| + M_{ik}) \quad \text{and} \quad M_{ik}^- = \frac{1}{2}(|M_{ik}| - M_{ik})$$

In summary, the steps of the SemiNMF-PCA algorithm can be deduced in Algorithm 2.

Algorithm 2: SemiNMF-PCA algorithm.

Input: Data matrix X, k and p
Initialize: - G using k-means, Q arbitrary orthonormal matrix.
repeat
  (a) - Update S by Eq. (3.14);
  (b) - Update G by Eq. (3.20)
  (c) - Update Q by solving Eq. (3.15)
until *convergence*;
Output: Indicator matrices G for data points and Q for features subspace

3.2.4 Fast SemiNMF-PCA

Despite its mathematical elegance, Eq. (3.4) suffers from two problems that impede its practical use. First, similar to Eq. (2), the relaxations on G make the immediate outputs of Eq. (3.4) are not cluster labels and the solution is often not unique. To this end an additional post-processing step is required.

Secondly, and more important, Eq. (3.4) is usually solved by an alternately iterative algorithm, and in each iteration step, the intensive matrix multiplications are involved [Ding et al., 2005, 2010, 2006b; Gu and Zhou, 2009]. Hence the scalability for such algorithms is problematic due to the expensive computational cost.

In order to tackle the difficulties mentioned above, instead of solving the relaxed clustering problems as in Eq. (3.4), we propose to solve the following clustering problem.

$$\min_{G,S,Q} \left\| X - GSQ^\top \right\|^2, \qquad \text{s.t.} \quad G \in \{0,1\}^{n\times k}, Q^\top Q = I. \tag{3.21}$$

Specifically, we constrain the factor G of SemiNMF-PCA to be a cluster indicator matrix. Similar to SemiNMF-PCA, the objective function of Fast SemiNMF-PCA can be decomposed into two terms as in Eq. (3.5). The first term is the objective function of the PCA, and the second is the k-means criterion in a low-dimensional subspace.

The optimization of F-SemiNMF-PCA leads to the similar updating formulas as in SemiNMF-PCA; S is obtained using Eq. (3.14) and Q is obtained by solving Eq. (3.15).

As G is now a binary cluster indicator matrix, its computation is done as follows: We fix S, Q and calculate

$$g_{ik} = \begin{cases} 1 & k = \arg\min_{k'} ||(XQ)_i - \mathbf{s}_{k'}||^2 \\ 0 & \text{otherwise.} \end{cases} \tag{3.22}$$

The steps of F-SemiNMF-PCA are summarized in Algorithm 3.

Algorithm 3: F-SemiNMF-PCA algorithm.

Input: Data matrix X, k and p
Initialize: G using k-means, Q arbitrary orthonormal matrix.
repeat
  (a) - Update S by Eq. (3.14);
  (b) - Update G by Eq. (3.22)
  (c) - Update Q by solving Eq. (3.15)
until *convergence*;
Output: Indicator matrices G for data points and Q for features subspace

Furthermore, we known that PCA provides an embedding for the data lying on a linear manifold. However, in many applications, data lie in a non-linear manifold. One popular method is to use the graph Laplacian based embedding. Next we propose a regularized version of Algorithm 3.

3.3 Regularized Fast SemiNMF-PCA (RF-SemiNMF-PCA)

3.3.1 Manifold Embedding using Graph Laplacian

We first construct a K-nearest neighbor data graph whose vertices correspond to the n data samples $[\mathbf{x}_1, \ldots, \mathbf{x}_n]$. We use the 01 weighting scheme to construct the K-nearest neighbor graph, and define the data weight matrix W as follows,

$$W_{ij} = \begin{cases} 1, & \text{if} \quad \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i); \quad i, j = 1, \cdots, n; \; i \neq j. \\ 0, & \text{otherwise} \end{cases}$$

where $\mathcal{N}(\mathbf{x}_i)$ represents the set of K-nearest neighbors of $\mathbf{x}_i$.

The Laplacian embedding [Belkin and Niyogi, 2001; Zhang and Zha, 2004] preserves the local geometrical relationships and maximizes the smoothness with respect to the intrinsic manifold of the data set in the low-embedding space. Let $\mathcal{G}^*$ be the embedding coordinates of the n data points. They are obtained by minimizing $\min_{\mathcal{G}} \sum_{i,j=1}^{n} W_{ij} \left\| g_i - g_j \right\|^2$. It is easy to show that:

$$\mathcal{G}^* = \arg\min_{\mathcal{G}} Tr(\mathcal{G}^\top (D - W)\mathcal{G}) \tag{3.23}$$

where D is a diagonal matrix the entries of which are row sums of the weight matrix W given by $D_{ii} = \sum_j W_{ij}$. The Laplacian embedding is closely connected with graph clustering. In fact, the embedding vectors of Eq. (3.23) provides an approximation solution for the Ration Cut Spectral Clustering [Chan et al., 1994], i.e., they can be seen as the relaxation solution of the cluster indicators (g_i for data i) in the spectral clustering objective function. This is similar to PCA being the spectral relaxation of k-means clustering [Ding and He, 2004].

3.3.2 RF-SemiNMF-PCA objective function

Graph Laplacian Regularized Fast SemiNMF-PCA clustering is defined as the minimizing problem of the following criterion:

$$\min_{G,S,Q} \left\| X - GSQ^\top \right\|^2 + \alpha Tr(G^\top (D - W)G), \quad G \in \{0, 1\}^{n \times k}, \; Q^\top Q = I. \tag{3.24}$$

where the parameter α is used to trade-off the contribution of the graph regularizing. Note that the objective function of RF-SemiNMF-PCA can be decomposed into three terms:

$$\left\| X - XQQ^\top \right\|^2 + \|XQ - GS\|^2 + \alpha Tr(G^\top (D - W)G) \tag{3.25}$$
$$G \in \{0, 1\}^{n \times k}, \; Q^\top Q = I.$$

The first term of equation (3.25) is the objective function of PCA, the second term is the k-means criterion in a low-dimensional subspace and the third term is the graph Laplacian regularization. Because G is constrained to be a cluster indicator matrix, it is often difficult to solve the objective function of our problem (3.24). It is, therefore, important that (3.24) be reformulated and simplified. To this end, we rely on the following proposition.

Proposition 2. *Given a symmetric and positive semi-definite similarity matrix A and its eigen-decomposition $U_A \Lambda U_A^\top$, where $\Lambda \in \mathbb{R}^{k\times k}$ is a diagonal matrix with diagonal elements as the k largest eigenvalues, and U_A is the corresponding eigenvector matrix. Let $B = U_A \Lambda^{0.5}$ and G is a non-negative partition matrix of size $n \times k$. Consider the orthonormal matrix Q, the following two optimization problems are equivalent:*

$$\min_G \left\| GG^\top - A \right\|^2 \Leftrightarrow \min_G \left\| G - BQ^\top \right\|^2 \quad s.t. \quad Q^\top Q = I.$$

Proof. Given a symmetric positive semi-definite similarity matrix A and its eigendecomposition $A = U_A \Lambda U_A^\top$. Further, we consider $A = GG^\top$ as a NMF of A. If $G = U_G \Sigma V_G^\top$ be the Singular Value Decomposition (SVD) of G, Then $A = GG^\top = U_G \Sigma^2 U_G^\top = U_A \Lambda_A U_A^\top$.

Consequently, we have $U_G = U_A$ and $\Lambda = \Sigma^2$. Let now consider $B = U_A \Lambda^{0.5}$, then there exists an orthonormal matrix Q such that $BQ^\top \geq 0$, thus finding G can be posed as the following optimization problem: $\min_G \left\| G - BQ^\top \right\|^2$ s.t. $Q^\top Q = I$. □

Using proposition 2 and considering the orthonormal matrix Q_g such as $Q_g^\top Q_g = I$, the expression (3.24) can be written as:

$$\begin{aligned} &\min_{G,S,Q,Q_g} \left\| X - GSQ^\top \right\|^2 + \alpha \left\| G - BQ_g \right\|^2 \\ &\text{s.t} \quad G \in \{0,1\}^{n\times k}, \quad Q^\top Q = I, \quad Q_g^\top Q_g = I. \end{aligned} \tag{3.26}$$

The second term in Eq. (3.27) can be written as

$$\|G - BQ_g\|^2 = \|G\|^2 + \|BQ_g\|^2 - 2Tr(G^\top BQ_g^\top) \tag{3.27}$$

Since G is a cluster indicator matrix, B and Q_g are both orthogonal matrices, the two first terms of Eq. (3.27) are both constant. Then, the optimization problem given in Eq. (3.27) is equivalent to

$$\begin{aligned} &\min_{G,S,Q,Q_g} \left\| X - GSQ^\top \right\|^2 - 2\alpha Tr(G^\top BQ_g^\top) \\ &\text{s.t} \quad G \in \{0,1\}^{n\times k}, \quad Q^\top Q = I, \quad Q_g^\top Q_g = I. \end{aligned} \tag{3.28}$$

Hereafter we present the computation of all matrices and parameters. The optimization of RF-SemiNMF-PCA leads to the similar updating formulae as in SemiNMF-PCA, S is obtained using Eq. (3.14) and Q by solving Eq. (3.15).

To calculate Q_g, we fix G, Q and S, and solve the following problem:

$$\max_{Q_g^\top Q_g = I} Tr[G^\top B Q_g] \tag{3.29}$$

Due to Theorem 1, applying SVD on $G^\top B$ we obtain: $Q_g = U_g V_g^\top$.

Next, because G is a cluster indicator matrix and is related to the graph Laplacian regularization term, its computation is done as follows: We fix S, Q and Q_g, and let $\widetilde{B}_g = B_g Q_g$. Each element of G is defined by

$$g_{ik} = \begin{cases} 1 & k = \texttt{arg}\min_{k'} ||(XQ)_i - \mathbf{s}_{k'}||^2 - 2\alpha(\widetilde{B}_g)_{ik'} \\ 0 & \text{otherwise.} \end{cases} \tag{3.30}$$

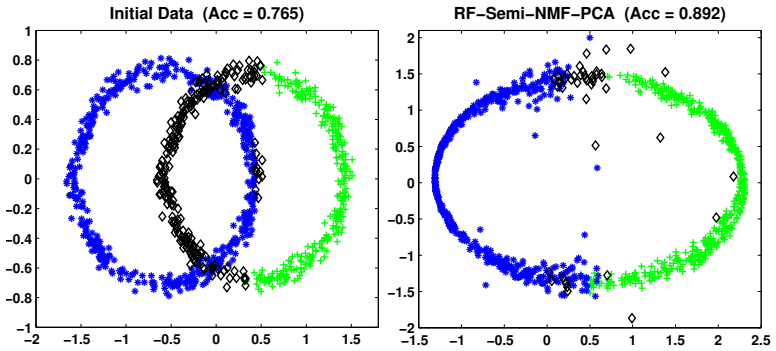
3.3.3 RF-SemiNMF-PCA algorithm

In summary, the steps of the RF-SemiNMF-PCA algorithm can be deduced in Algorithm 4.

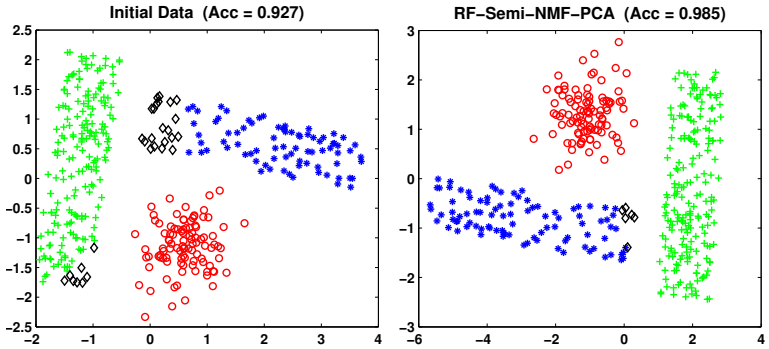
Algorithm 4: RF-SemiNMF-PCA algorithm.

Input: Data matrix X, k and p
Initialize: - G using k-means,
- Q and Q_g with arbitrary orthonormal matrices.
repeat
(a) - Update S by Eq. (3.14);
(b) - Update G by (3.30)
(c) - Update Q by solving Eq. (3.15)
(d) - Update Q_g by solving Eq. (3.29)
until *convergence*;
Output: Indicator matrices G for data points and Q for features subspace

Illustration Before investigating the behaviors of all proposed algorithms on real data sets, we propose to illustrate this last version. We applied RF-SemiNMF-PCA on the FCPS data sets previously defined, i.e, Tetra, Lsun and Chainlink. In Fig. 3.2, we observe the ability of RF-SemiNMF-PCA to propose a good solution in terms of accuracy and separability between the clusters.



(a) Chainlink data set.



(b) Lsun data set.

Figure 3.2 – RF-SemiNMF-PCA performances on FCPS data sets. Black points represent the misclassified objects. Acc denotes the accuracy (the percentage of objects well classified).

3.4 Experiments

In the following subsections we discuss some of the advantages of our contribution against two kinds of clustering methods:

1) NMF-based methods including NMF, PNMF and SemiNMF [Ding et al., 2010].

2) Two-steps-based methods including LDA-k-means, PCA-k-means and Ncut-k-means. These methods are based on two steps performed separately. The first step consists in performing an eigendecomposition-based dimensional reduction of the features space and the second step in applying k-means on the first few principal components.

Performance metrics. To measure the clustering performance of the proposed algorithms we use the commonly adopted metrics, the accuracy, the Normalized Mutual Information [Strehl and Ghosh, 2002] and the Adjusted Rand Index [Hubert and Arabie, 1985]. For these three metrics (Acc, NMI and ARI), a value close to 1 means a good clustering result.

Parameter settings. We run each method under different parameter settings 50 times and we report the best result for each method. For all the compared methods, we set the number of sample clusters equal to the true number of classes in data sets (k) and we use k-means or *spherical k-means* (Sk-means) [Dhillon, 2001a] to initialize the sample partition matrix G according the type of data.

- For NMF, PNMF and SemiNMF the best parameters are used, as suggested in each of the reference articles (see for details [Ding et al., 2005, 2010; Zhirong and Laaksonen, 2007]). For Ncut-k-means, we have used the code of Ncut provided by Zhirong et al. [Zhirong and Laaksonen, 2007]. For, LDA-k-means, we have used the code of LDA provided by Deng Cai [Cai et al., 2006]. Note that LDA is a supervised method, where its components are computed using the partition obtained by $k$means rather than the true cluster label that is assumed to be unknown.
- In order to assess the number of components, for SemiNMF-PCA, F-SemiNMF-PCA, RF-SemiNMF-PCA, Ncut and PCA, we varied the number of components p between 2 and $10k$ and retained the one that optimizes the criterion. For LDA, we set $p = k - 1$.
- For RF-SemiNMF-PCA, the graph Laplacian is constructed using the K-Nearest Neighbors (K-NN) in which the neighborhood size is fixed to 10. The regularization parameter α is searched from the grid (0.01, 0.1, 1, 10, 100, 500, 1000).
- To evaluate all studied methods, we consider three types of data, with different characteristics: sparsity rates, sizes (where $n << d$ and $n >> d$) and balances. Thus 10 sparse data sets and 14 not sparse data sets will be considered.

3.4.1 Results on sparse data sets

Data sets. These experiments were performed using some benchmark Document-term data sets from the clustering literature. Table 3.1 summarizes the characteristics of these data sets.

Table 3.1 – Description of Document-term Data sets

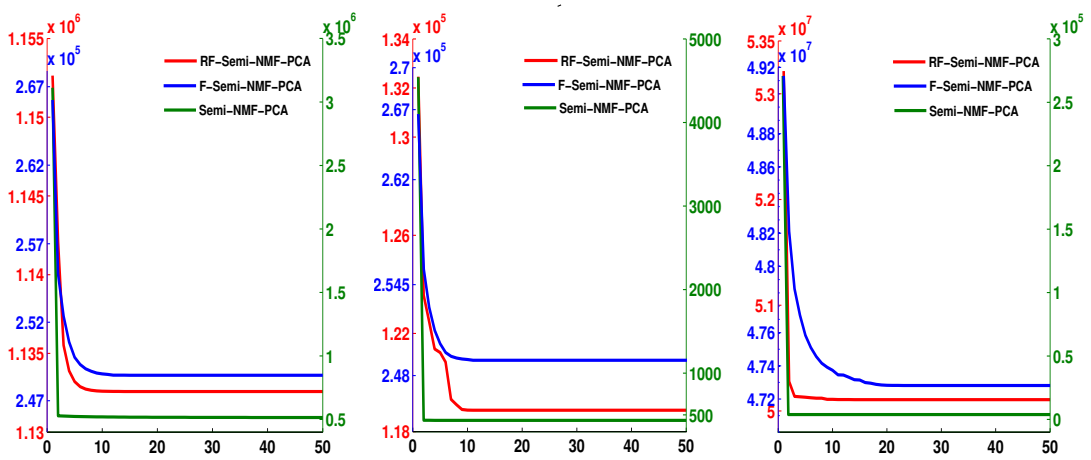
Data sets	**Characteristics**				
	#Documents	#Terms	#Clusters	Sparsity (%)	Balance
CSTR	475	1000	4	96.60	0.399
WebKB4	4199	1000	4	93.90	0.307
WebACE	2340	1000	20	91.83	0.169
NG10	500	2000	10	0.858	1
NG20	19949	43586	20	99.99	0.991
RCV1	9625	29992	4	99.75	0.766
Reviews	4069	18483	5	99.99	0.098
Sports	8580	14870	7	99.99	0.036
Classic3	3891	4303	3	98.0	0.710
Classic4	7095	5896	4	99.41	0.323

Note that, for all the used document-term data sets, we apply the TF-IDF transformation on all the document-term frequency matrices. We used the TF-IDF weighting scheme proposed in scikit-learn [Pedregosa et al., 2011] which is defined by $w_{ij} = tf_{ij}(1 + \log(\frac{1+n}{1+d_j}))$, where w_{ij} is the weight of term i in document j, tf_{ij} is the frequency of term i in document j, n is the total number of documents and d_j is the number of documents containing term j.

Computation speed. In order to study experimentally the asymptotic behavior of our proposed algorithms and them potential to converge and compare its computation speeds, we repeat the clustering 50 times using the different methods with the optimal parameters. The average computation time of our three proposed methods, i.e, SemiNMF-PCA, F-SemiNMF-PCA and RF-SemiNMF-PCA, applied to the different text data sets are reported in Table 3.2.

Table 3.2 – Average computation time for convergence on document-term data sets.

	Algorithms		
Data sets	**SemiNMF-PCA**	**F-SemiNMF-PCA**	**RF-SemiNMF-PCA**
CSTR	0,159	0,142	0,185
WebKB4	1,430	1,235	1,549
WebACE	1,293	1,047	1,323
NG10	0,190	0,151	0,202
NG20	75,300	67,752	80,149
RCV1	15,74	14,462	15,908
Reviews	3,207	2,824	3,263
Sports	15,183	12,840	15,413
Classic3	1,696	1,584	1,718
Classic4	7,804	6,181	8,224



(a) WebACE (b) Classic3 (c) Reviews

Figure 3.3 – Illustration of the convergence study of our proposed algorithms on document-term data sets. ”x” axis is the iteration number and ”y” axis represents the criterion.

The obtained results illustrate the monotonic (see Fig.3.3) and rapid convergence of our algorithms (as reported in Table 3.2), and show that F-Semi-NMF-PCA requires less time to converge than Semi-NMF-PCA. Furthermore, we can observe that, in general, the more the data dimensions increase, the more the gain in terms of computation time increases too. For instance, the gain is 0.382 seconds (sec) for Reviews data set (4069×18483), 2.343 sec for Sports data set (8580×14870) and 7.548 sec for NG20 data set (19949×43586). These results are consistent with our theoretical analysis in which we have chosen F-SemiNMF-PCA (faster and more efficient) to be extended to RF-SemiNMF-PCA by introducing the locality preserving.

Document-term data Clustering performances The assessment of all the algorithms in terms of Acc, NMI and ARI are reported in Table 3.3. The main comments arising from our experiments are the following. First, the NMF-based methods including NMF, PNMF and the SemiNMF give similar results none among them outperforms the others. However, the Two-steps based methods including Ncut, PCA and LDA are equivalent with a slight advantage for LDA-k-means. Finally, our proposed algorithms give better results than both NMF-based methods and Two-steps based methods. We observe that RF-SemiNMF-PCA is better than F-SemiNMF-PCA that is itself better than SemiNMF. In addition we have noted that RF-SemiNMF-PCA allows a good separability between the clusters as illustrated in Fig. 3.4 for CSTR data set. We observe a good solution obtained by the simultaneous combination of SemiNMF and PCA.

Table 3.3 – Results obtained by the compared methods on sparse (document-term) data sets in terms Acc, NMI and ARI. The *spherical k-means* (Sk-means) is the most efficient on sparse data sets.

			NMF-based Methods			**Two-Steps-based Methods**			**Our proposed Methods**		
Data Set	**Metric**	**Sk-means**	**NMF**	**PNMF**	**SemiNMF**	**Ncut-k-means**	**PCA-k-means**	**LDA-k-means**	**SemiNMF-PCA**	**F-SemiNMF-PCA**	**RF-SemiNMF-PCA**
CSTR	Acc	0.894	0.903	**0.905**	0.798	0.771	0.901	**0.905**	0.905	0.916	**0.924**
	NMI	0.761	0.776	**0.778**	0.729	0.708	0.765	**0.772**	0.762	0.803	**0.810**
	ARI	0.798	0.807	**0.814**	0.747	0.686	0.805	**0.811**	0.812	0.839	**0.847**
WebKB4	Acc	0.717	0.796	**0.799**	0.790	0.737	0.762	**0.789**	0.790	0.795	**0.807**
	NMI	0.490	0.535	**0.545**	0.496	0.455	0.488	**0.521**	0.525	0.539	**0.554**
	ARI	0.477	0.555	**0.562**	0.537	0.485	0.509	**0.547**	0.545	0.558	**0.572**
WebACE	Acc	0.572	0.650	**0.651**	0.626	0.512	0.547	**0.654**	0.641	0.654	**0.658**
	NMI	0.629	0.652	**0.665**	0.556	0.517	0.518	**0.650**	0.643	0.652	**0.666**
	ARI	0.518	0.677	**0.681**	0.518	0.405	0.563	**0.689**	0.688	0.689	**0.693**
Ng10	Acc	0.531	**0.543**	0.541	0.474	0.442	0.416	**0.474**	0.601	0.651	**0.764**
	NMI	0.455	**0.489**	0.465	0.434	0.472	0.440	**0.489**	0.506	0.587	**0.657**
	ARI	0.337	0.317	**0.352**	0.265	0.242	0.194	**0.291**	0.383	0.454	**0.568**
Ng20	Acc	0.548	0.449	0.550	**0.622**	0.619	0.596	**0.641**	0.619	0.643	**0.654**
	NMI	0.547	0.414	0.593	**0.615**	0.608	0.591	**0.617**	0.612	0.619	**0.631**
	ARI	0.408	0.283	0.417	**0.461**	0.459	0.431	**0.471**	0.450	0.495	**0.499**
RCV1	Acc	0.681	0.737	**0.757**	0.756	0.744	**0.768**	0.716	0.795	0.807	**0.822**
	NMI	0.443	0.502	**0.588**	0.561	0.543	**0.595**	0.480	0.636	0.644	**0.661**
	ARI	0.425	0.493	**0.531**	0.530	0.500	**0.540**	0.477	0.584	0.608	**0.615**
Reviews	Acc	0.706	**0.827**	0.751	0.747	0.742	**0.777**	0.758	0.762	0.779	**0.814**
	NMI	0.548	**0.693**	0.672	0.671	0.644	**0.641**	0.672	0.652	0.646	**0.700**
	ARI	0.484	**0.631**	0.630	0.600	0.625	**0.663**	0.637	0.645	0.665	**0.725**
Sports	Acc	0.678	0.700	0.691	**0.738**	0.647	0.658	**0.685**	0.721	0.737	**0.752**
	NMI	0.665	0.645	0.677	**0.678**	0.678	0.654	**0.672**	0.687	0.751	**0.802**
	ARI	0.521	0.523	0.528	**0.566**	0.507	0.516	**0.538**	0.545	0.627	**0.628**
Classic3	Acc	0.886	**0.909**	0.905	0.901	0.920	0.901	**0.912**	0.948	0.965	**0.992**
	NMI	0.749	0.768	**0.777**	0.767	0.801	0.765	**0.778**	0.902	0.944	**0.956**
	ARI	0.776	**0.826**	0.821	0.806	0.844	0.805	**0.826**	0.898	0.943	**0.975**
Classic4	Acc	0.708	0.719	**0.812**	0.774	0.745	0.802	**0.827**	0.829	0.836	**0.852**
	NMI	0.674	0.687	**0.765**	0.729	0.694	0.736	**0.774**	0.780	0.788	**0.808**
	ARI	0.468	0.518	**0.635**	0.582	0.540	0.602	**0.681**	0.689	0.702	**0.735**

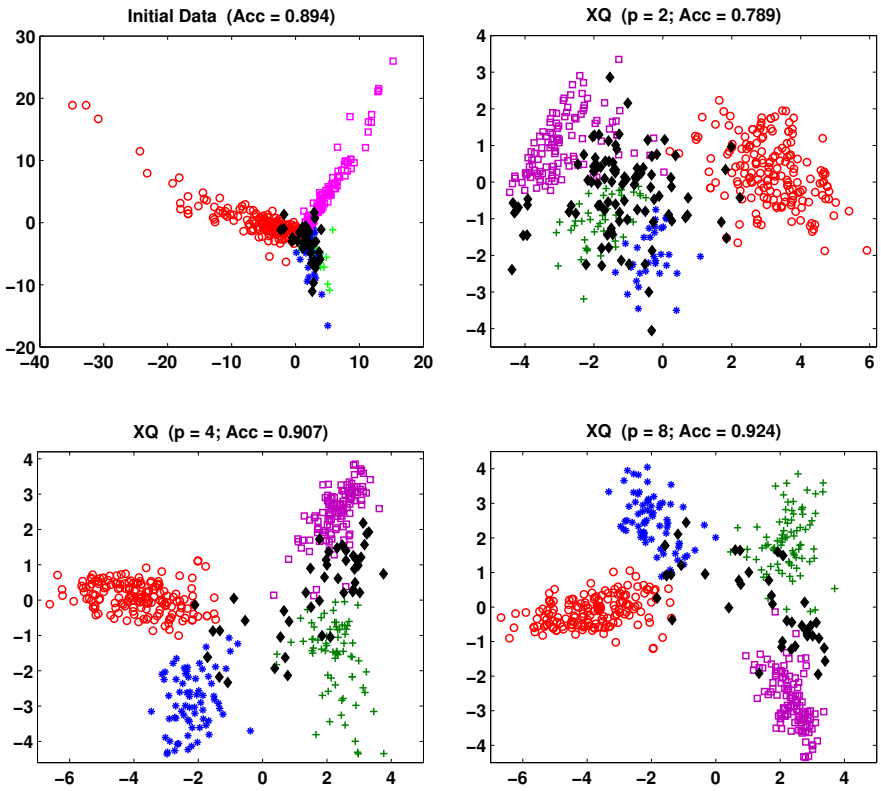


Figure 3.4 – CSTR data set: Projection of the objects into the factorial plane spawned by the two first components. Initial data X while the clustering is obtained by Sk-means and XQ of size $n \times p$ while the clustering is obtained by RF-SemiNMF-PCA. Black points represent the misclassified objects obtained by k-means and Acc denotes the accuracy which is the percentage of objects well classified. The best Accuracy is obtained with $(p = 8)$.

3.4.2 Results on Non-sparse data sets

To assess our approach on other data types, experiments were performed using some benchmark image and microarray data sets from the clustering literature. Table 3.4 summarizes the characteristics of these data sets.

Like for sparse data, in table 3.5, we report the performances of our best method RF-SemiNMF-PCA against the best compared methods of each category, i.e, PNMF (NMF-based methods) and Ncut-k-means (Two-steps-based methods), in terms of Acc, NMI and ARI. We observe the good performance of our approach for all data sets. It is clear that RF-SemiNMF-PCA is most effective; the regularization always brings some improvement. First, as illustrated in Fig. 3.5 and Fig. 3.6, RF-SemiNMF-PCA has a high capability to separate the obtained clusters.

Table 3.4 – Image and microarray data sets description.

Data Sets	Characteristics				
	Type	#samples	#features	#classes	Sparsity(%)
Coil20	Image	1440	1024	20	34.38
Coil100	Image	7200	1024	20	0
ORL	Image	400	1024	40	0
Yale	Image	165	1024	15	30.54
USPS	Image	9298	256	10	0
PIE	Image	2856	1024	68	8.53
MNIST	Image	70000	784	10	80.85
Leukemia	Microarray	72	1762	2	0
Lung	Microarray	203	12600	5	0
Colon	Microarray	62	2000	2	0
Breast	Microarray	106	9	6	0.21
Yeast	Microarray	1484	8	10	12.41
Isolet	Microarray	1559	617	26	0.35

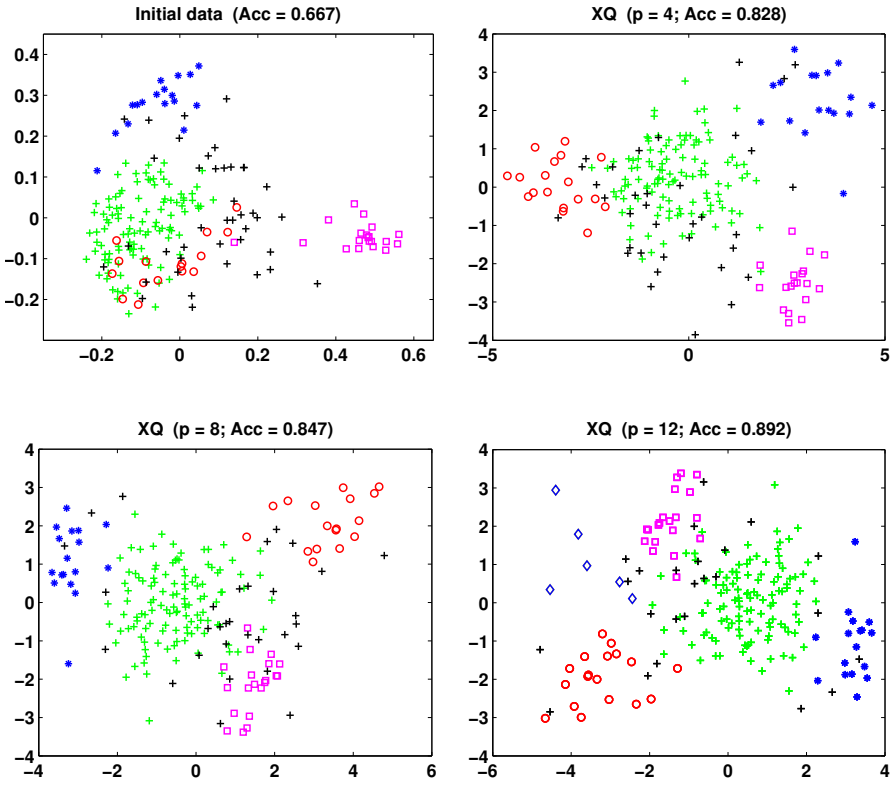


Figure 3.5 – Lung: Projection of the objects into the factorial plane spawned by the two first components. Initial data X while the clustering is obtained by Sk-means and XQ of size $n \times p$ while the clustering is obtained by RF-SemiNMF-PCA. Black points represent the misclassified objects obtained by k-means and Acc denotes the accuracy which is the percentage of objects well classified. The best Accuracy is obtained with $(p = 12)$.

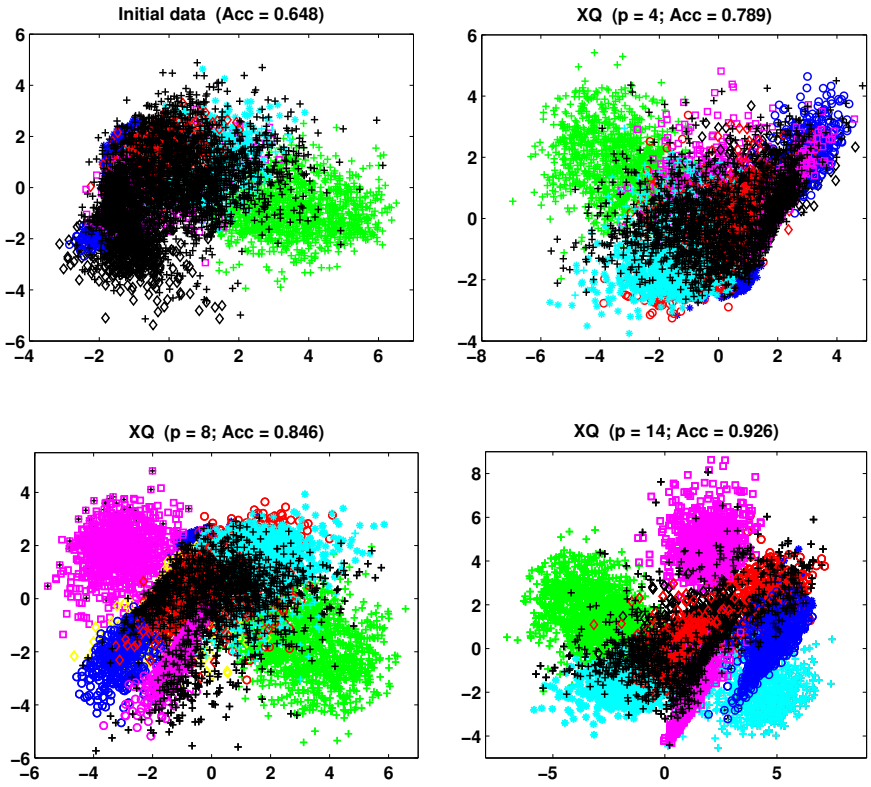


Figure 3.6 – USPS: Projection of the objects into the factorial plane spawned by the two first components. Initial data X while the clustering is obtained by Sk-means and XQ of size $n \times p$ while the clustering is obtained by RF-SemiNMF-PCA. Black points represent the misclassified objects obtained by k-means and Acc denotes the accuracy which is the percentage of objects well classified.The best Accuracy is obtained with $(p = 14)$.

Table 3.5 – Results obtained by the compared methods on non sparse data sets in terms Acc, NMI and ARI. k-means is the most efficient on image and microarray data sets.

			NMF-based Methods			Two-Steps-based Methods			Our proposed Methods		
Data Set	**Metric**	**k-means**	**NMF**	**PNMF**	**SemiNMF** *(Ding)*	**Ncut-k-means**	**PCA-k-means**	**LDA-k-means**	**SemiNMF-PCA**	**F-SemiNMF-PCA**	**RF-SemiNMF-PCA**
Coil20	Acc	0.608	0.637	**0.676**	0.666	0.668	0.690	**0.698**	0.701	0.740	**0.762**
	NMI	0.727	0.746	**0.769**	0.741	0.744	0.760	**0.781**	0.794	0.813	**0.813**
	ARI	0.550	0.552	**0.597**	0.589	0.604	0.599	**0.611**	0.625	0.679	**0.678**
Coil100	Acc	0.450	0.519	**0.536**	0.488	0.486	0.543	**0.551**	0.559	0.561	**0.583**
	NMI	0.744	0.750	**0.779**	0.746	0.749	0.764	**0.768**	0.780	0.782	**0.790**
	ARI	0.409	0.436	**0.472**	0.410	0.427	0.481	**0.480**	0.507	0.510	**0.521**
ORL	Acc	0.487	0.595	**0.603**	0.580	**0.608**	0.525	0.583	0.636	0.662	**0.674**
	NMI	0.713	0.750	**0.774**	0.761	**0.780**	0.737	0.766	0.782	0.805	**0.819**
	ARI	0.318	0.392	**0.468**	0.453	**0.469**	0.389	0.436	0.476	0.511	**0.525**
Yale	Acc	0.430	**0.491**	0.442	0.448	**0.509**	0.489	0.497	0.521	0.545	**0.570**
	NMI	0.501	**0.526**	0.508	0.522	**0.555**	0.521	0.530	0.554	0.579	**0.598**
	ARI	0.227	**0.267**	0.241	0.243	**0.296**	0.266	0.236	0.293	0.316	**0.361**
USPS	Acc	0.648	0.666	**0.873**	0.677	**0.852**	0.619	0.840	0.880	0.903	**0.926**
	NMI	0.607	0.608	**0.826**	0.615	**0.772**	0.548	0.746	0.780	0.828	**0.842**
	ARI	0.521	0.528	**0.798**	0.533	**0.730**	0.470	0.715	0.779	0.820	**0.828**
PIE	Acc	0.261	0.303	0.274	**0.344**	**0.392**	0.338	0.311	0.395	0.419	**0.427**
	NMI	0.560	0.558	0.583	**0.598**	**0.704**	0.660	0.641	0.711	0.730	**0.739**
	ARI	0.156	0.156	0.167	**0.231**	**0.247**	0.192	0.158	0.271	0.316	**0.322**
MNIST	Acc	0.570	0.636	**0.643**	0.606	0.621	0.532	**0.638**	0.637	0.654	**0.656**
	NMI	0.582	0.541	**0.564**	0.622	0.671	0.457	**0.696**	0.703	0.718	**0.720**
	ARI	0.415	0.435	**0.443**	0.515	0.536	0.351	**0.581**	0.561	0.583	**0.591**
Leukemia	Acc	0.722	0.778	**0.798**	0.753	0.889	**0.917**	0.792	0.903	0.931	**0.958**
	NMI	0.194	0.202	**0.270**	0.212	0.592	**0.632**	0.269	0.619	0.675	**0.740**
	ARI	0.135	0.280	**0.300**	0.226	0.637	**0.685**	0.307	0.644	0.734	**0.837**
Lung	Acc	0.667	0.695	0.675	**0.700**	**0.823**	0.734	0.793	0.847	0.867	**0.892**
	NMI	0.572	0.618	0.587	**0.619**	**0.638**	0.639	0.563	0.639	0.678	**0.728**
	ARI	0.388	**0.447**	0.392	0.444	**0.575**	0.468	0.555	0.612	0.741	**0.752**
Colon	Acc	0.629	0.677	0.661	**0.687**	**0.726**	0.694	0.710	0.721	0.726	**0.742**
	NMI	0.077	0.133	0.108	**0.133**	**0.190**	0.133	0.139	0.175	0.190	**0.220**
	ARI	0.031	0.113	0.094	**0.142**	**0.169**	0.137	0.143	0.152	0.169	**0.208**
Breast	Acc	0.491	0.519	**0.528**	0.502	0.529	0.462	**0.557**	0.566	0.575	**0.585**
	NMI	0.334	0.431	**0.461**	0.422	0.440	0.419	**0.507**	0.515	0.530	**0.562**
	ARI	0.182	0.236	**0.251**	0.189	0.280	0.232	**0.339**	0.374	0.388	**0.408**
Yeast	Acc	0.456	0.547	**0.566**	0.541	0.535	0.540	**0.559**	0.556	0.559	**0.569**
	NMI	0.137	0.277	**0.280**	0.261	0.261	0.269	**0.299**	0.287	0.299	**0.304**
	ARI	0.079	0.150	**0.166**	0.142	0.137	0.139	**0.181**	0.162	0.181	**0.188**

Furthermore, by using the row clusters obtained by PNMF, Ncut-k-means and RF-SemiNMF-PCA, Fig. 3.7 shows the reorganized images of the USPS data set. It reveals the good result of our proposed method in image data sets classification.

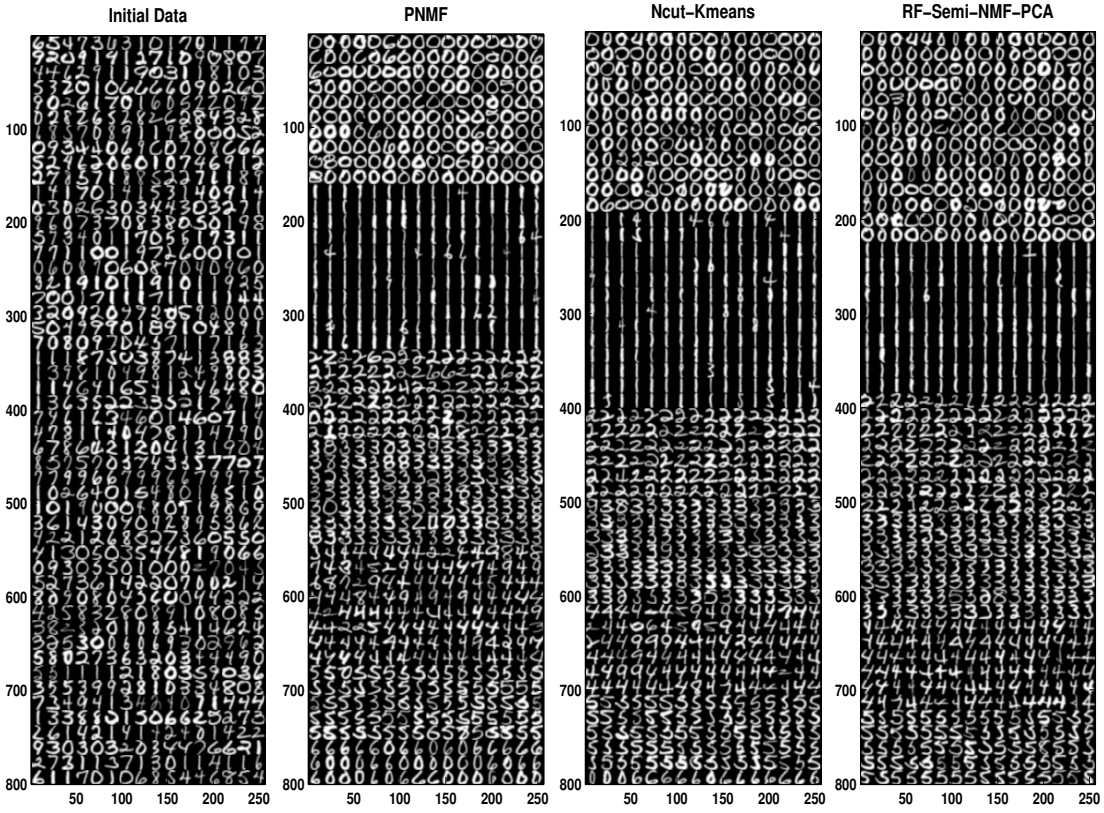


Figure 3.7 – Performances of the compared methods, k-means, Ncut-k-means and RF-SemiNMF-PCA, on USPS image data set.

3.4.3 Statistical tests

The first question addressed is whether there were any significant differences among our three proposed methods including RF-SemiNMF-PCA, F-SemiNMF-PCA and SemiNMF-PCA? To this end, we first test for the significance of performance differences between RF-SemiNMF-PCA, F-SemiNMF-PCA and SemiNMF-PCA. We used the analysis of variance (ANOVA) and Kruskal-Wallis (KW) tests. The obtained p-values are reported in Table 3.6[1]. As it can be seen in table 3.6, for each data set the difference among the compared methods is statistically significant; all p-values are less than 1%.

[1]We selected two datasets of image, text and microarray to illustrate our comparisons

Furthermore, we exploit the statistics obtained by ANOVA in applying a post-hoc analysis of RF-SemiNMF-PCA, F-SemiNMF-PCA and SemiNMF-PCA. The Scheffé's procedure is the most popular of the post-hoc procedures (see for instance [Scheffé, 1959]). The obtained results for studied data sets show that RF-SemiNMF-PCA almost always significantly outperform F-SemiNMF-PCA and SemiNMF-PCA, we illustrate this performance in Table 3.7. Furthermore, the Scheffé tests confirm the performance differences between these compared methods; most of the p-values are less than 5%. The same observations are verified from other data sets.

Table 3.6 – Variance analysis of RF-SemiNMF-PCA, F-SemiNMF-PCA and SemiNMF-PCA Accuracy's using ANOVA and Kruskal-Wallis (KW) tests (with $\alpha = 0.05$) performed on 50 random initialisations.

Data set	P-Value	
	ANOVA	KW
CSTR	2.68e-29	9.50e-18
WebAce	4.02e-12	4.83e-05
Coil20	2.14e-72	4.59e-29
USPS	1.64e-45	8.69e-22
Leukemia	3.23e-25	1.53e-19
Lung	1.14e-04	7.24e-04

Table 3.7 – Post-hoc analysis of RF-SemiNMF-PCA, F-SemiNMF-PCA and SemiNMF-PCA Accuracy's using Scheffé test (with $\alpha = 0.05$) performed on 50 random initialisations.

Data set	Methods	P-Value
CSTR	F-SemiNMF-PCA SemiNMF-PCA	1.70e-22
	RF-SemiNMF-PCA SemiNMF-PCA	1.02e-25
	RF-SemiNMF-PCA F-SemiNMF-PCA	0.047
Webace	F-SemiNMF-PCA SemiNMF-PCA	6.35e-02
	RF-SemiNMF-PCA SemiNMF-PCA	9.40e-02
	RF-SemiNMF-PCA F-SemiNMF-PCA	0.043
Coil20	F-SemiNMF-PCA SemiNMF-PCA	1.39e-59
	RF-SemiNMF-PCA SemiNMF-PCA	4.60e-68
	RF-SemiNMF-PCA F-SemiNMF-PCA	2.31e-05
USPS	F-SemiNMF-PCA SemiNMF-PCA	2.22e-02
	RF-SemiNMF-PCA SemiNMF-PCA	8.15e-42
	RF-SemiNMF-PCA F-SemiNMF-PCA	3.79e-35
Leukemia	F-SemiNMF-PCA SemiNMF-PCA	0.018
	RF-SemiNMF-PCA SemiNMF-PCA	3.01e-16
	RF-SemiNMF-PCA F-SemiNMF-PCA	1.14e-23
Lung	F-SemiNMF-PCA SemiNMF-PCA	0.011
	RF-SemiNMF-PCA SemiNMF-PCA	0.079
	RF-SemiNMF-PCA F-SemiNMF-PCA	1.15e-04

Secondly, to confirm the performance of RF-Semi-NMF-PCA compared with the best Two-steps-based method on some representative data sets, i.e, LDA-k-means for document-term data sets and Ncut-k-means for image and microarray data sets, for each data set, we perform pairwise t-tests on 50 random initialisations.

Table 3.8 – RF-SemiNMF-PCA vs LDA-k-means: Evaluation on document-term data sets in terms of Acc, NMI and ARI; using t-tests performed on 50 random initialisations.

Data set		Metric	LDA-k-means	RF-SemiNMF-PCA	P-values
Document-term	CSTR	Acc	0.786 ±0.055	0.894 ±0.024	$< 0.1\%$
		NMI	0.575 ±0.069	0.762 ±0.034	$< 0.1\%$
		ARI	0.571 ±0.114	0.798 ±0.031	$< 0.1\%$
	WebKB4	Acc	0.789 ±0.005	0.786 ±0.034	0.704
		NMI	0.521 ±0.001	0.532 ±0.024	$< 0.1\%$
		ARI	0.546 ±0.002	0.546 ±0.038	0.519
	WebACE	Acc	0.613 ±0.045	0.618 ±0.027	0.278
		NMI	0.629 ±0.044	0.643 ±0.028	0.042
		ARI	0.560 ±0.041	0.613 ±0.039	$< 0.1\%$
	NG10	Acc	0.466 ±0.004	0.744 ±0.006	$< 0.1\%$
		NMI	0.407 ±0.003	0.636 ±0.005	$< 0.1\%$
		ARI	0.276 ±0.006	0.549 ±0.006	$< 0.1\%$
	Reviews	Acc	0.598 ±0.034	0.673 ±0.052	$< 0.1\%$
		NMI	0.620 ±0.013	0.639 ±0.025	0.001
		ARI	0.583 ±0.014	0.649 ±0.029	0.005
	Sports	Acc	0.528 ±0.039	0.707 ±0.034	$< 0.1\%$
		NMI	0.603 ±0.028	0.771 ±0.025	$< 0.1\%$
		ARI	0.378 ±0.027	0.569 ±0.026	$< 0.1\%$
	Classic3	Acc	0.903 ±0.018	0.967 ±0.008	$< 0.1\%$
		NMI	0.767 ±0.027	0.912 ±0.012	$< 0.1\%$
		ARI	0.812 ±0.029	0.928 ±0.015	$< 0.1\%$
	Classic4	Acc	0.725 ±0.046	0.818 ±0.039	$< 0.1\%$
		NMI	0.685 ±0.063	0.731 ±0.035	$< 0.1\%$
		ARI	0.566 ±0.083	0.701 ±0.021	$< 0.1\%$

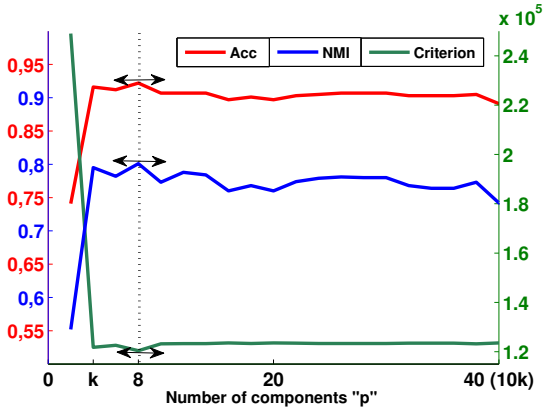
Table 3.9 – RF-SemiNMF-PCA vs Ncut–k-means: Evaluation on image and microarray data sets in terms of Acc, NMI and ARI; using t-tests performed on 50 random initialisations.

Data set		Metric	Ncut-k-means	RF-SemiNMF-PCA	P-values
Image	Coil20	Acc	0.639 ±0.044	0.745 ±0.004	$< 0.1\%$
		NMI	0.754 ±0.021	0.804 ±0.005	$< 0.1\%$
		ARI	0.562 ±0.045	0.666 ±0.007	$< 0.1\%$
	ORL	Acc	0.566 ±0.040	0.659 ±0.012	$< 0.1\%$
		NMI	0.755 ±0.031	0.808 ±0.013	$< 0.1\%$
		ARI	0.419 ±0.045	0.514 ±0.016	$< 0.1\%$
	Yale	Acc	0.426 ±0.040	0.537 ±0.018	$< 0.1\%$
		NMI	0.483 ±0.037	0.592 ±0.013	$< 0.1\%$
		ARI	0.195 ±0.049	0.342 ±0.015	$< 0.1\%$
	USPS	Acc	0.789 ±0.044	0.894 ±0.024	$< 0.1\%$
		NMI	0.722 ±0.039	0.806 ±0.029	$< 0.1\%$
		ARI	0.641 ±0.071	0.779 ±0.065	$< 0.1\%$
Microarray	Leukemia	Acc	0.772 ±0.060	0.886 ±0.045	0.009
		NMI	0.263 ±0.243	0.582 ±0.137	0.002
		ARI	0.276 ±0.257	0.622 ±0.177	$< 0.1\%$
	Lung	Acc	0.750 ±0.049	0.813 ±0.022	0.014
		NMI	0.466 ±0.083	0.624 ±0.033	0.006
		ARI	0.391 ±0.092	0.638 ±0.037	0.003
	Colon	Acc	0.706 ±0.018	0.721 ±0.011	$< 0.1\%$
		NMI	0.110 ±0.034	0.131 ±0.024	$< 0.1\%$
		ARI	0.120 ±0.028	0.156 ±0.016	$< 0.1\%$
	Breast	Acc	0.463 ±0.084	0.531 ±0.035	$< 0.1\%$
		NMI	0.418 ±0.126	0.491 ±0.066	$< 0.1\%$
		ARI	0.236 ±0.099	0.304 ±0.040	$< 0.1\%$

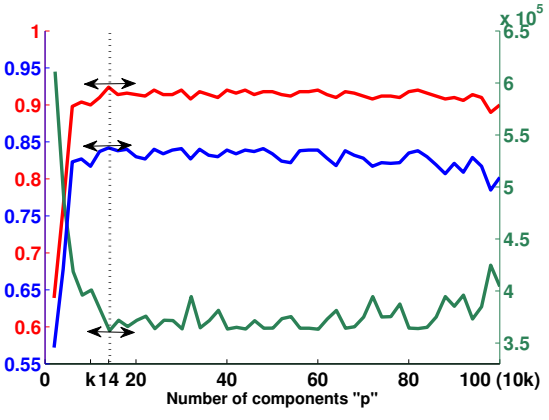
In Table 3.8, we show that, for document-term data sets, the improvement between RF-Semi-NMF-PCA and LDA-k-means is statistically significant; most of the p-values are less than 0.1%. Similarly, in Table 3.9, we show that, for image and microarray data sets, the improvement between RF-Semi-NMF-PCA and Ncut-k-means is statistically significant; most of the p-values are less than 0.1%.

3.4.4 Assessing the number of components

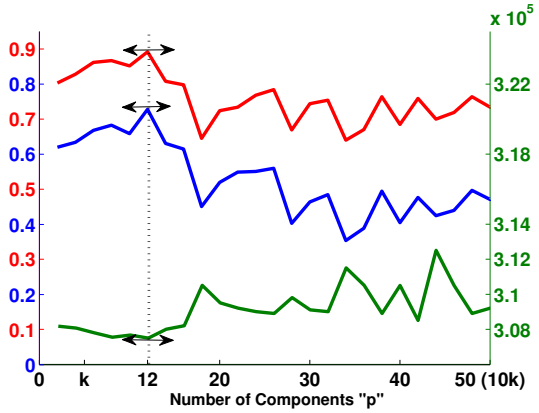
In our experiments and in order to assess the number of components, we varied p between 2 and $10k$, and retained the one that optimizes the criterion. The questions that naturally arises is: how many components are necessary to give a good result?



(a) CSTR ($k = 4$).



(b) USPS ($k = 10$).



(c) Lung ($k = 5$).

Figure 3.8 – Performances of RF-SemiNMF-PCA according to the number of components "p" in terms of Acc and NMI.

In all our experiments, the obtained results in terms of clustering and visualization are very encouraging and specifically with RF-Semi-NMF-PCA. For all data sets, the retained p optimizing the criterion (3.28) corresponds to the best result in terms of Acc, NMI and ARI as illustrated in Fig. 3.8. To confirm these results on these three data sets, we report in Table 3.10 the performances recorded by LDA-k-means for $p = k-1$, Ncut-k-means and RF-semi-NMF-PCA for $p = k-1$, $p = k$ and $p = p^*$ where p^* is equal to the value optimizing the criterion (3.28).

Table 3.10 – RF-Semi-NMF-PCA vs LDA-k-means vs Ncut-k-means: Comparison of performances according to the number of components p in terms of Acc, NMI and ARI. The considered values of p are k, $k-1$ and p^* where p^* denotes the value of p optimizing the criterion. k is the true number of clusters.

Data set	Metric	**LDA-k-means** (or Sk-means)	**Ncut-k-means** (or Sk-means)			**RF-SemiNMF-PCA**		
		$p = k-1$	$p = k-1$	$p = k$	$p = p^*$	$p = k-1$	$p = k$	$p = p^*$
CSTR	Acc	0.905	0.718	0.762	0.771	0.897	0.907	0.924
$k = 4$	NMI	0.772	0.601	0.695	0.708	0.766	0.773	0.810
	ARI	0.811	0.558	0.657	0.686	0.794	0.822	0.847
USPS	Acc	0.740	0.818	0.823	0.852	0.904	0.900	0.926
$k = 10$	NMI	0.676	0.754	0.761	0.772	0.823	0.817	0.842
	ARI	0.595	0.715	0.719	0.730	0.808	0.804	0.828
Lung	Acc	0.793	0.808	0.793	0.823	0.828	0.862	0.892
$k = 5$	NMI	0.563	0.631	0.620	0.638	0.634	0.668	0.728
	ARI	0.555	0.550	0.536	0.575	0.663	0.728	0.752

In this way, we can compare the three methods and we observe the substantial interest of RF-semi-NMF-PCA with p^*. Hence our strategy appears effective; given the number of clusters it can provide the appropriate number of components. According to our experiments on the 24 data sets, the choice of p between k and $5k$ seems an appropriate way to assess this parameter. However, further theoretical investigations are necessary.

3.5 Conclusion

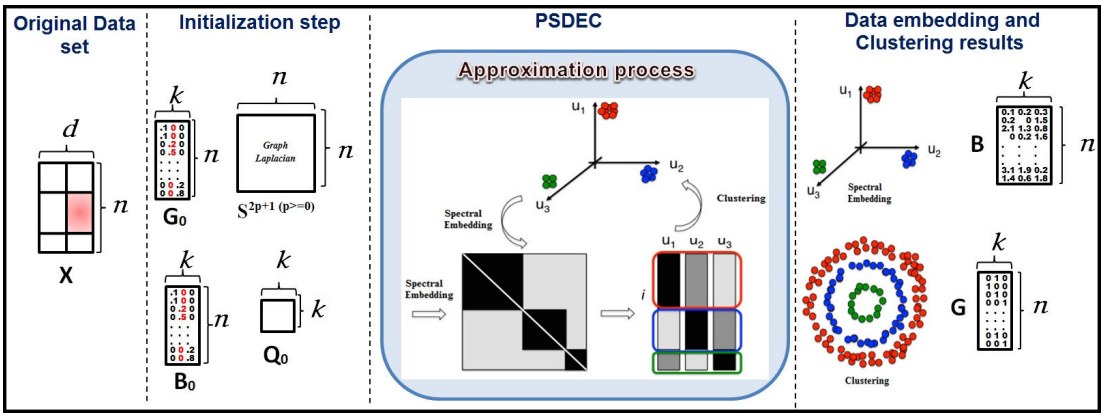
The dual purpose of this paper is to reduce the dimension and the clustering. Based on the decomposition of the objective function of Semi-NMF-PCA into two terms where the first one is the objective function of PCA and the second is the Semi-NMF criterion in a low-dimensional space, we proposed a novel way to consider the clustering and the reduction of the dimension simultaneously. Our approach takes advantage of the mutual reinforcement between data reduction and clustering tasks. Such a solution better approximates the relaxed continuous dimension reduction solution by the true discrete clustering solution. We also establish theoretical connections among our method and NMF, k-means and PNMF; that explains the performance improvement.

Three variants of partitioning algorithms have been proposed. On sparse or not sparse data sets, they give better results in terms of clustering than the state-of-the-art algorithms devoted to similar tasks for data sets with different sizes, degrees of overlapping and balances. In addition, They offer good performances in terms of separability between clusters, hence they can also be beneficial for visualization.

Certainly we proposed a solution to choose of the number of components, the used strategy relies to a certain extent on the number of classes. It would therefore be interesting to investigate the simultaneous choices of the number of classes and the number of components. This objective is our major ongoing research activity.

Chapter 4

Simultaneous Spectral Data Embedding and Clustering



4.1 Introduction

Clustering is widely used for exploratory data analysis, with applications ranging from artificial intelligence, statistics to social sciences. Among various clustering methods in the literature, spectral clustering is a popular choice. It is easy to implement efficiently and often outperforms traditional clustering methods such as k-means. In recent years, the interest in spectral clustering suitable for various data mining problems has increased. The method has been well studied in the literature [Bach and Jordan, 2006; Ng et al., 2001], many tutorials giving an introduction to spectral clustering are available; see for instance [Luxburg, 2007]. The spectral clustering methods based on the graph partitioning theory focus on finding the best cuts of a graph that optimize certain predefined criterion functions. Their optimization usually leads to the computation of singular vectors or eigenvectors of certain graph affinity matrices. Many criterion functions have been proposed such as the average cut [Chan et al., 1994], the average association [Shi and Malik, 2000], the normalized cut [Shi and Malik, 2000], and the min-max cut [Ding et al., 2001]. On the other hand, connections between spectral clustering and other clustering methods have been established; see for instance [Dhillon et al., 2004; Ding et al., 2005; Luo et al., 2010; Nie et al., 2010].

Our focus is the area of spectral clustering which uses graph cuts as objective functions for nonlinear data separation. Spectral clustering algorithms represent data as a graph where data samples are vertices and edge weights represent the similarity between data samples. Then data are partitioned by finding a k-way graph cut in two steps:

1. finding a spectral embedding by using an eigendecomposition of the graph Laplacian matrix; and
2. based on the embedding, finding a partition via a simplified clustering algorithm such as k-means.

Spectral clustering has the advantage of requiring weak assumptions regarding the shapes of clusters. Moreover, it is applicable to a wide variety of data types and similarity functions. However, classical spectral methods such as Ratio Cut [Hagen and Kahng, 1992] and Normalized Cut [Ng et al., 2001; Shi and Malik, 2000] generally use k-means to perform the clustering on the relaxed continuous spectral vectors in order to obtain the final clusters. The disadvantage of this approach is that it consists in optimizing two different objectives. Hence, spectral low-dimensional embedding and clustering are successively and not simultaneously used. For this reason, certain obtained continuous low-dimensional embedding can deviate far from the clustering solution, thereby affecting the partition quality. Finally, due to the computational complexity of $O(n^3)$ in general, with n the number of data points, the applicability of spectral clustering for large-scale problems remains limited.

we address this problem using simultaneous spectral dimensionality reduction and clustering. We first propose a novel framework referred to as Simultaneous Spectral data Embedding and Clustering (SDEC) which alternates both tasks iteratively. SDEC relies on a matrix decomposition technique to simultaneously learning a spectral data embedding B, a clustering matrix G and a rotation matrix Q which closely maps the continuous spectral embedding to the clustering solution. As we will show, this usually leads to a better embedding approximation and improvement in clustering accuracy. It is worth highlighting the novelty of our proposed framework. It allows

- to propose a unified framework for spectral clustering combining low dimensional embedding learning and clustering in a common procedure. Then the optimization of a single learning objective function is necessary to achieve simultaneously spectral embedding and clustering tasks.
- to apply the spectral rotation technique to get the continuous spectral vector which is closer to the cluster membership indicator than existing results.
- to be less costly than traditional spectral clustering and to be better than existing methods commonly used for the same tasks.

Then, we propose a novel framework, referred to as Power Spectral Data Embedding and Clustering (PSDEC), which alternates the spectral clustering and the dimensionality reduction while relying on the classical *Power method* [Golub and van Loan, 1996]. It is worth highlighting the novelty of PSDEC which make it possible to

- propose a unified framework for spectral clustering combining low-dimensional embedding learning and clustering in a common procedure. Then the optimization of a single learning objective function is necessary to achieve spectral embedding and clustering tasks simultaneously.
- perform on a stochastic powered matrix; the purpose of the use of such matrix is twofold. First, it allows to the use of *Power method* inside our algorithm in order to speed up the eigenvectors computation, and secondly to explore the similarity matrix structure via a random walk process and then make the similarity matrix more suitable for the clustering task (quasi block diagonal matrix).

The rest of this chapter is organized as follows. Section 2 provides notation and related works. We formulate the proposed PSDEC framework and provide an effective method to solve this problem in Section 3. Then, we describe several experiments we have run, compare our algorithm to other algorithms from the literature on several benchmark image data sets. Finally, we conclude with additional observations and future work.

4.2 Spectral and Symmetric NMF approaches

Our work is inspired by spectral clustering for which we review some related work in this sequel.

4.2.1 Spectral clustering.

Spectral clustering can be presented from different points of views [Luxburg, 2007]; in this Chapter, we focus on the graph partitioning viewpoint. Given a set of n data samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with each $\mathbf{x}_i$ a column vector in R^d, and given a set of similarities, $\{k_{ij}\}$, between all pairs $\mathbf{x}_i$ and $\mathbf{x}_j$, where $k_{ij} \geq 0$. Let $\mathcal{G} = (V, E)$ be a graph, where $V = \{v_1, \ldots, v_n\}$ is the set of vertices and E the set of edges. Each vertex v_i in the graph represents a data sample $\mathbf{x}_i$, with the similarities k_{ij} treated as edge weights. If there is no edge between v_i and v_j then $k_{ij} = 0$. Let the matrix K with elements k_{ij} be the similarity matrix. This matrix is generally obtained from a kernel function, an example of which is the Gaussian kernel $k(\mathbf{x}_i; \mathbf{x}_j) = exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2)$.

The aim of spectral clustering is to partition the data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ into k disjoint clusters $\{P_1, \ldots, P_k\}$, such that the similarity of the samples between clusters is low, and the similarity of the samples within clusters is high. There are several objective functions that capture this desideratum; in this subsection we focus on the normalized cut objective.

4.2.1.1 K-way normalized cut.

The k-way normalized cut, Ncut($\mathcal{G}$), is defined as

$$Ncut\{P_1, \ldots, P_k\} = \sum_{c=1}^{k} \frac{cut(P_c, V \setminus P_c)}{vol(P_c)}$$

where the cut between sets $A, B \in V$ is defined by $cut(A; B) = \sum_{v_i \in A, v_j \in B} k_{ij}$, the degree d_i of vertex $v_i \in V$ is defined as $d_i = \sum_{j=1}^{n} k_{ij}$, the volume of set $A \subseteq V$ is defined by $vol(A) = \sum_{i \in A} d_i$, and $V \setminus A$ denotes the complement of A. In this objective function, note that $cut(P_c, V \setminus P_c)$ measures the between cluster similarity and the within cluster similarity captured by the normalizing term $vol(P_c)$.

The next step is to rewrite Ncut($\mathcal{G}$) using an indicator matrix B of cluster membership of size $n \times k$ and to note that Ncut($\mathcal{G}$) takes the form of a Rayleigh quotient in B. Relaxing the indicator matrix to allow its entries to take on any real value, we obtain a generalized eigenvector problem.

The problem reduces to the following relaxed Ncut minimization:

$$\min_{B \in R^{n \times k}} Tr(B^\top LB) \text{ s.t. } B^\top B = I \tag{4.1}$$

where L is the normalized graph Laplacian, with $L = I - D^{-0.5}KD^{-0.5}$, I is an identity matrix, D is a diagonal matrix whose diagonal entries are the degree d_i, B is the spectral embedding matrix and Tr denotes the trace of a matrix.

Minimizing the relaxed Ncut objective is equivalent to maximizing the relaxed normalized association as follows:

$$\max_{B \in R^{n \times k}} Tr(B^\top D^{-0.5}KD^{-0.5}B) \quad \text{s.t.} \quad B^\top B = I. \tag{4.2}$$

The solution is to set B equal to the k eigenvectors corresponding to the largest k eigenvalues of the normalized similarity, $W = D^{-0.5}KD^{-0.5}$. This yields the spectral embedding. Based on this embedding, the discrete partitioning of the data is obtained from a "rounding" step. One specific rounding algorithm, due to [Ng et al., 2001], is based on renormalizing each row of B to have unit length and then applying k-means on the rows of the normalized matrix. We then assign each x_i to the cluster that the row b_i is assigned to.

4.2.1.2 Random walk view.

Different algorithms use matrix K with different ways to derive an affinity matrix W. In this paper, we adopt the random walk view [Meila and Shi, 2001] for the definition of the affinity matrix; W is given by $W = D^{-1}K$. Note that the sum of each row of W is equal to 1, thus W_{kl} can be interpreted as the probability for a random walk that begins at w_k and ends up at w_l after a single step. More formally, if we let c_j be the location of the walk at time j, then $W_{kl} = P(c_{j+1} = w_k | c_j = w_l)$.

The proposed algorithm by Meila and Shi [Meila and Shi, 2001] assumes that K and the number of clusters k are given with the data. First the affinity matrix $W = D^{-1}K$ is computed. Then the top k eigenvectors of W of the generalized eigensystem $Kv = \lambda Dv$ or equivalently $Wv = \lambda v$ are used to cluster the data. Furthermore, we can observe that the Ng-Jordan-Weiss (NJW) algorithm uses the top k eigenvectors of $W = D^{-0.5}KD^{-0.5}$ of the eigensystem $Wu = \lambda u$ to map data. Indeed, we can relate U to V by $U = D^{0.5}V$. Since D is a diagonal matrix, when the row is normalized to length 1, the B obtained from V is identical to the B obtained from U.

4.2.1.3 Spectral clustering algorithm.

Spectral clustering is a technique that exploits the properties of the Laplacian of the graph, whose edges denote the similarities between the data points. The top k eigenvectors of the normalized graph Laplacian are the relaxations of the indicator vectors that assign each node in the graph to one of the k clusters. Apart from being theoretically well-motivated, spectral clustering has the advantage of performing well on arbitrary shaped clusters, which is otherwise a shortcoming of several other clustering algorithms such as the k-means algorithm.

Here we briefly outline the spectral clustering algorithm due to [Meila and Shi, 2001].

Algorithm 5: : Spectral Clustering

Input: Initial data matrix X, cluster number k.
Output: A Partition: $\Delta = \{A_1, A_2, ..., A_k\}$.

1. Construct an $n \times n$ positive semi-definite similarity matrix (or kernel) K, where K_{ij} quantifies the similarity between samples i and j;
2. Compute the normalized graph Laplacian matrix denoted $W = D^{-1}K$, where D is a diagonal matrix defined by $D_{ii} = \sum_j K_{ij}$;
3. Compute the top k eigenvectors of W, the obtained low-dimensional matrix is referred as to B;
4. Consider the rows of B as data points, and run k-means algorithm to cluster them into k clusters;

4.2.2 Symmetric NMF.

As shown above different graph cuts can be reduced to the following trace maximization form (see equation 4.2), where $B \in \mathbb{R}^{n \times k}$ satisfies some constraints and indicates the clustering assignment. A group of successful graph clustering methods-spectral clustering, relaxes the constraints on B to $B^\top B = I$. Under such orthogonality constraint on B, we have the equivalence between the two following optimizations [Ding et al., 2005]:

$$\begin{aligned} \max_B Tr(B^\top W B) &\Leftrightarrow \min_B Tr(W^\top W) - 2Tr(B^\top W B) + Tr(I) \\ &\Leftrightarrow \min_B Tr(W - BB^\top)^\top (W - BB^\top) \\ &\Leftrightarrow \min_B ||W - BB^\top||_F^2. \end{aligned}$$

Hence, compared to spectral clustering, Symmetric NMF can be seen as a different relaxation to $\min_B ||W - BB^\top||_F^2$, i.e. relaxing the constraints on B to be $B >= 0$.

According to the constraints on B, spectral clustering and Symmetric NMF have different properties in the clustering results they generate. Spectral clustering leads to eigenvector-based solutions of B, which are not necessarily nonnegative; and k-means or more advanced procedures have to be adopted in order to obtain the final clustering assignments. By contrast, the solution found with Symmetric NMF naturally captures the cluster structure. It also indicates the clustering assignments without additional clustering procedures, which heavily depends on initialization, such as k-means. Therefore, after obtaining B via eigen-analysis, we can formulate the recovery of the cluster membership matrix G as follows: $BQ = G$ where Q is an $(k \times k)$ orthonormal rotation matrix which most closely maps B to G. Specifically,

$$\min_{B} ||BQ - G||_F^2 \quad \text{s.t } Q^\top Q = I, \quad G \geq 0. \tag{4.3}$$

In the following section we propose a new objective function that combines spectral data embedding and clustering in a common framework. Thus, we aim to combine both advantages of spectral clustering and symmetric NMF while avoiding the large computational cost of these methods.

4.3 Simultaneous Spectral Data Embedding and Clustering (SDEC)

In the sequel, we present the details of the proposed algorithm. A fast iterative method is also proposed to solve the objective function.

4.3.1 SDEC objective function

Let k be the number of clusters and the number of components to which the data is embedded. SDEC clustering is defined as the minimizing problem of the following criterion:

$$\min_{B,Q,G} \left\| W - GQB^\top \right\|^2. \quad \text{s.t.} \quad Q^\top Q = I, \quad B^\top B = I, \quad G \geq 0. \tag{4.4}$$

The nonnegative matrix $G = (g_{ij})$ of size $(n \times k)$ is a cluster membership matrix, $B = (b_{ij})$ of size $(n \times k)$ is the embedding matrix and $Q = (q_{ij})$ of size $(k \times k)$ is an orthonormal rotation matrix which most closely maps B to G.

Proposition 3. *Given $G \geq 0$, $B^\top B = I$ and $Q^\top Q = QQ^\top = I$, the objective function of SDEC can be decomposed into two terms:*

$$\left\| W - GQB^\top \right\|^2 = \left\| W - WBB^\top \right\|^2 + \|WB - GQ\|^2 \tag{4.5}$$

Proof. We first expand the matrix norm of the left term of Eq.(4.5)

$$\left\|W - GQB^\top\right\|^2 = \|W\|^2 + \left\|GQB^\top\right\|^2 - 2Tr(WGQB^\top) \tag{4.6}$$

In a similar way, we obtain from the two terms of the right term of Eq.(4.5)

$$\begin{aligned}
\left\|W - WBB^\top\right\|^2 &= \|W\|^2 + \left\|WBB^\top\right\|^2 - 2Tr(WBB^\top W^\top) \\
&= \|W\|^2 + \left\|WBB^\top\right\|^2 - 2||WB||^2 \\
&= ||W||^2 - ||WB||^2 \quad \text{due to } B^\top B = I \qquad (4.7) \\
\text{and} \qquad \|WB - GQ\|^2 &= \|WB\|^2 + \|GQ\|^2 - 2Tr(WBQG^\top) \qquad (4.8)
\end{aligned}$$

Due also to $B^\top B = I$, we have

$$\|WB - GQ\|^2 = ||WB||^2 + ||GQB^\top||^2 - 2Tr(WGQB^\top) \tag{4.9}$$

Summing the two terms Eq. (4.7) and Eq. (4.9) leads to the left term of Eq. (4.5).

$$\|W\|^2 + \|GQ\|^2 - 2Tr(WGQB^\top) = \left\|W - GQB^\top\right\|^2 \tag{4.10}$$

due to $\|GQ\|^2 = \left\|GQB^\top\right\|^2$ □

Using proposition 3, the objective function of SDEC (4.4) can be decomposed into two terms: The first one is the objective function of spectral embedding, and the second is the semi-NMF criterion in a low embedding subspace. To solve the problem (4.4), we rely on the theorem 1 used previously in Section 3.2.3.

4.3.2 Optimization

To solve (4.4), we propose updating Q, B and G in an alternating fashion.

Computation of Q. First, fixing G and B, problem in Eq. (4.4) is equivalent to:

$$Eq.(4.4) \Leftrightarrow \min_Q \left\|\tilde{B} - GQ\right\|^2 \Leftrightarrow \min_Q \left\|G - \tilde{B}Q^\top\right\|^2 \tag{4.11}$$

where $\tilde{B} = WB$. Due to theorem 1, by applying SVD to $G^\top\tilde{B}$, we obtain $Q = UV^\top$. we can observe that this problem turns out to be similar to the well known orthogonal Procrustes problem [Schonemann, 1966].

Computation of B. Secondly, given G and Q, minimizing Eq. (4.4) is equivalent to

$$\max_{B} \; Tr(WGQB^{\top}) \quad \text{s.t.} \quad B^{\top}B = I. \tag{4.12}$$

Due to theorem 1, by applying SVD to $QG^{\top}W$ we obtain $B = UV^{\top}$.

Computation of G. Thirdly, update G by keeping B, and Q fixed at the value computed in the above steps, Eq.(4.4) is equivalent to the following problem

$$\min_{G} \; \left\| WBQ^{\top} - G \right\|^2 \quad \text{s.t.} \quad G \geq 0. \tag{4.13}$$

Since G is non-negative, we simply set

$$G = \max(\mathbf{0}, WBQ^{\top}). \tag{4.14}$$

Since the objective function of SDEC is linear when fixing either of the matrices on the right side, an alternating least square approach can be used to optimize the solution in each iteration. In addition, to keep the non-negativity of elements of matrices, we use idea analogous to the projected gradient methods in iterations of NMF [Lin, 2007]. The projected gradient methods are typical approaches to solve bound-constrained optimization problems, where variables are constrained by certain bounds. NMF is a kind of bound-constrained optimization problem. The basic idea of the projected gradient is to update variables as in normal gradient descent method, but when the variables are out of the bounds, they are pulled back into the bounds by projection.

In summary, the steps of the SDEC algorithm can be deduced in Algorithm 6.

Algorithm 6: : SDEC

Input: Similarity matrix W
Initialize:B and Q with arbitrary orthonormal matrix
repeat
  (a) - Update G by (4.14)
  (b) - Update B by solving Eq. (4.12);
  (c) - Update Q by solving (4.11)
until *convergence*;
Output: - G for data point clustering,
- Q for matrix rotation and
- B for low dimensional embedding.

4.4 Power Spectral Data Embedding and Clustering (PSDEC)

4.4.1 PSDEC objective function.

Let k be the number of clusters and the number of components to which the data is embedded. PSDEC clustering is defined as the minimizing problem of the following criterion:

$$\min_{B,Q,G} \left\| W^{2p+1} - GQB^\top \right\|^2, \quad \text{s.t.} \quad Q^\top Q = I, \quad B^\top B = I, \quad G \geq 0. \tag{4.15}$$

The nonnegative matrix $G = (g_{ij})$ of size $(n \times k)$ is a cluster membership matrix, $B = (b_{ij})$ of size $(n \times k)$ is the embedding matrix and $Q = (q_{ij})$ of size $(k \times k)$ is an orthonormal rotation matrix which most closely maps B to G. To solve the problem (4.15), it suffices to rely on proposition 3 by postulating $S = W^{2p+1}$ (where p is any positive integer). Using proposition 3, the objective function of PSDEC (4.15) can be decomposed into two terms: the first one is the objective function of spectral embedding, and the second is the semi-NMF criterion in a low embedding subspace.

4.4.2 Power Method for speeding up eigenvectors computation.

One way to speed up Algorithm 5 is to use the *Power method* which is a well-known technique to compute the largest left and right eigenvectors of data matrices [Golub and van Loan, 1996; Lin and Cohen, 2010]. In Step 4, to quickly approximate the eigenvectors in B; authors in [Boutsidis et al., 2015] use the *Power method*: First initialize $U \in R^{n \times k}$ with Independent and identically distributed random Gaussian variables. Let $\tilde{B} \in R^{n \times k}$ containing the left singular vectors of matrix $S = (WW^\top)^p WU = W^{2p+1}U$; for some integer p. Then, use $\tilde{B}$ instead of B in step 4 of algorithm 5.

4.4.3 Powered similarity matrix -Random walk Analysis :

The idea behind the use of W^p is to explore the structure of W when random walk takes many steps instead of only one. It is well known, from the theory of Markov chains [Azran and Ghahramani, 2006], that W^p (where p is any odd positive integer) is given by multiplying W with itself p times, so that if $W = V\Lambda V^\top$ then $W^p = V\Lambda^p V^\top$, where V is the matrix whose n^{th} column is v_n. Thus, if λ_n, v_n is the eigensystem of W, then λ_S^p, v_n is the eigensystem of W^p. Next, if p is odd then the ordering of the eigenvalues is left unchanged and the same eigenvectors are picked to cluster the data.

In a prior work by Meila et al [Meila and Shi, 2001], it is noted that for many natural problems, W is an approximately block stochastic matrix, hence the first k left eigenvectors of W are approximate piecewise constant over the g almost invariant rows subsets. The iterative random walk process will converge to the approximated data W^p where each row and each column moves towards their prototypes. In other words, this process converges to an equilibrium (steady) state. Matrix W is composed of $k << n$ quasi similar rows, where each row is represented by its prototype.

Consider using the p^{th} order transition matrix W^p as the affinity matrix. W^p_{mn} gives the total probability that a random walk x_j , beginning at w_m, will end up in $w_n$after p steps, considering all possible paths between the nodes. W^p_{mn} is expected to be high if there is a good path between w_m, w_n and low otherwise, hopefully leading to a block diagonal matrix which is ideal for clustering data [Meila and Shi, 2001] [Azran and Ghahramani, 2006]. However, often in practice we observe a different behavior of W^p. If points i, j are in the same cluster, then often there are values of p for which W^p_i and W^p_j , the i^{th} and j^{th} rows of W^p, become very similar. The intuition here is that if points i, j are similar then after a sufficient number of steps we can expect that a particle that begins a random walk in each of them will have the same distribution for their locations after p steps. Another observation is that by varying the number of steps p we explicitly explore similarities at different scales in the data, and as p increases we expect to find a coarser structure.

To solve the problem (4.15), we rely on the theorem 1 used previously in Section 3.2.3.

4.4.4 Optimization.

To solve Eq.(4.15), we propose updating Q, B and G in an alternating fashion.

Computation of Q. First, fixing G and B, the problem in eq.(4.15) is equivalent to:

$$\text{Eq.(4.15)} \Leftrightarrow \min_{Q} \left\| \tilde{B} - GQ \right\|^2 \Leftrightarrow \min_{Q} \left\| G - \tilde{B}Q^\top \right\|^2 \tag{4.16}$$

where $\tilde{B} = SB$. Due to theorem 1, by applying SVD to $G^\top \tilde{B}$, we obtain $Q = UV^\top$.

Computation of B. Secondly, given G and Q, minimizing eq.(4.15) is equivalent to

$$\max_{B} \; Tr(SGQB^\top) \quad \text{s.t.} \quad B^\top B = I. \tag{4.17}$$

Due to theorem 1, by applying SVD to $QG^\top S$ we obtain $B = UV^\top$.

Computation of G. Since the objective function of PSDEC is linear when fixing either of the matrices on the right side, an alternating least square approach can be used to optimize the solution. In addition, to keep the non-negativity of elements of matrices, we use idea analogous to the projected gradient methods in the iterations of NMF [Lin, 2007]. The projected gradient methods are typical approaches to solve bound-constrained optimization problems, where variables are constrained by certain bounds.

NMF is a kind of bound-constrained optimization problem. The basic idea of the projected gradient is to update variables as in the normal gradient descent method, but when the variables are out of the bounds, they are pulled back into the bounds by projection. Thus, updating G by keeping B, and Q fixed at the value computed in the above steps, eq.(4.4) is equivalent to the following problem

$$\min_{G} \left\| SBQ^{\top} - G \right\|^2 \quad \text{s.t.} \quad G \geq 0. \tag{4.18}$$

Since G is non-negative, we simply set

$$G = \max(\mathbf{0}, SBQ^{\top}). \tag{4.19}$$

In summary, the steps of the PSDEC algorithm can be deduced in Algorithm 7. Note that PSDEC and SDEC are equivalent when $p = 0$.

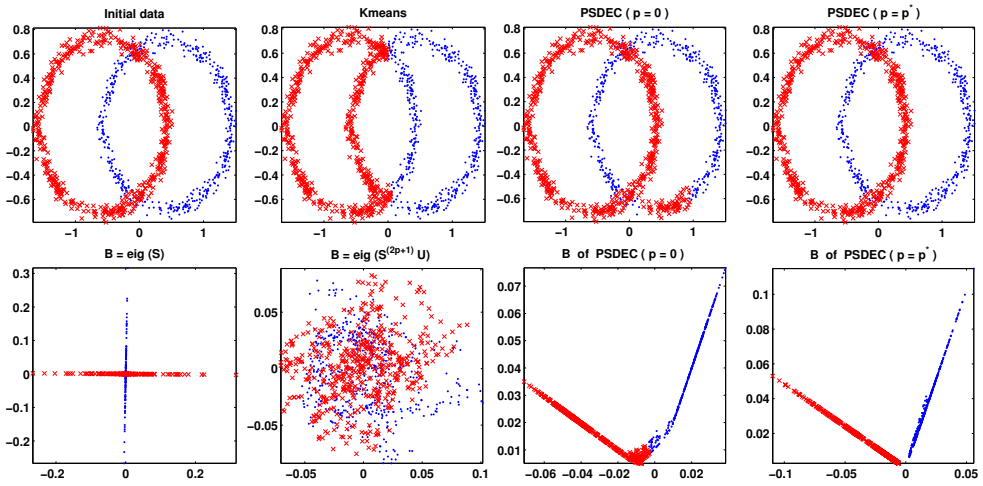
Algorithm 7: : PSDEC algorithm

Input: Powered similarity matrix $S = W^{2\times p+1}$
Initialize: B and Q with arbitrary orthonormal matrix;
repeat
  (a) - Update G by (4.19)
  (b) - Update B by solving Eq. (4.17);
  (c) - Update Q by solving (4.16)
until *convergence*;
Output: - G for data point clustering,
- Q for matrix rotation and
- B for low-dimensional embedding.

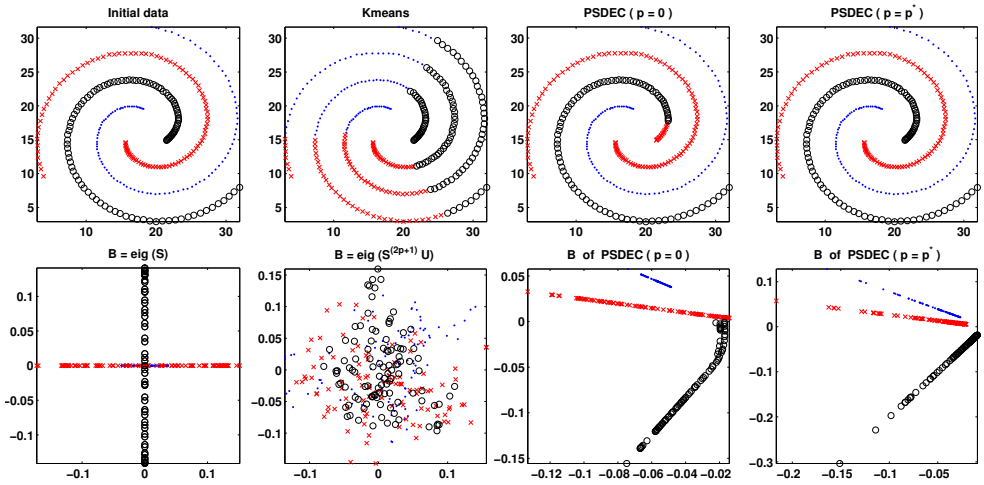
4.4.5 Illustration with synthetic data sets.

In order to illustrate the ability of our method to preserve the initial topology and to separate classes, we used two generated synthetic data sets called: Chainlink (Fundamental Clustering Problem Suite (FCPS)) and Spiral (Shape).

Figure 4.1 illustrates the capability of PSDEC to obtain and separate classes of synthetic data sets commonly used in the spectral clustering context. By applying the PCA, we can show the 2D-projection of the initial data matrix with: 1) the true labels; 2) the labels obtained by k-means; 3) the labels obtained by PSDEC ($p = 0$); 4) the labels obtained by PSDEC ($p = p^*$). Moreover, the 2D-projection of the spectral data embedding matrix B shows clearly the good performances of PSDEC in terms of data dimensionality reduction and clustering.



(a) Chainlink



(b) Spiral

Figure 4.1 – Clustering and visualization with SDEC ($p = 0$), PSDEC ($p = p^*$) and k-means. p^* denotes the value of p optimizing the criterion.

4.4.6 Complexity analysis (Computational cost).

In terms of per-iteration complexity, the matrix multiplications SGQ requires $O(n^2k)$ flops and $G^\top SB$ requires $O(nk^2)$ flops, whereas the SVD performed on the relatively small-sized $k \times k$ matrix $G^\top SB$ requires $O(k^3)$ flops and the SVD performed on matrix SGQ requires $O(kn^2 + nk^2 + k^3)$ flops [Golub and van Loan, 1996]. If we assume $n \gg k$, which is typically the case in practice, then the term $O(n^2k)$ dominates, which results in a per-iteration complexity of our both algorithms SDEC and PSDEC.

4.5 Relationship with other state-of-the-art methods

4.5.1 Matrix decomposition and symmetric NMF.

The objective function in eq.(4.4) is equivalent to the following spectral matrix decomposition problem [Kuang et al., 2012]

$$\begin{aligned} &\min_{B,Q,G} \left\|W - GQB^\top\right\|^2 \Leftrightarrow \min_{B,Q,G} \left\|W - BB^\top\right\|^2 \\ &\text{s.t.} \quad B = GQ \quad Q^\top Q = I, B^\top B = I, G \geq 0. \end{aligned} \tag{4.20}$$

In a similar way we can establish the equivalence to symmetric NMF [Kuang et al., 2012] and we have

$$\begin{aligned} &\min_{B,Q,G} \left\|W - GQB^\top\right\|^2 \Leftrightarrow \min_{B,Q,G} \left\|W - GG^\top\right\|^2 \\ &\text{s.t.} \quad G = BQ \quad Q^\top Q = I, B^\top B = I, G \geq 0. \end{aligned} \tag{4.21}$$

Thus, PSDEC objective includes spectral matrix decomposition when $B = GQ$ and symmetric NMF when $G = BQ$.

4.5.2 Spectral embedding and Reduced Semi-NMF.

As shown in eq.(4.5), the objective of PSDEC is decomposed into two terms,

$$\left\|W - GQB^\top\right\|^2 = \left\|W - WBB^\top\right\|^2 + \|WB - GQ\|^2.$$

The first term ($\left\|W - WBB^\top\right\|^2$) is the objective function of spectral embedding (since S and W have the same eigenvectors). The second term ($\|WB - GQ\|^2$) is the objective function of reduced semi-NMF recently proposed in [Allab et al., 2015b].

4.5.3 Orthogonal Procrustes problem.

For fixed B, and G. The second term in eq.(4.5) turns out to be similar to the well known orthogonal Procrustes problem [Schonemann, 1966].

Power Method [Boutsidis et al., 2015]: We first expand the matrix norm of the left term of eq.(4.5)

$$\min_B \left\|W - GQB^\top\right\|^2 \quad \Leftrightarrow \quad \min_B \|W\|^2 + \left\|GQB^\top\right\|^2 - 2Tr(WGQB^\top)$$

Then, for a fixed G, PSEDC turn out to be equivalent to a trace maximization problem

$$\max_B Tr(WGQB^\top). \tag{4.22}$$

On the other hand, Boutsidis et al. [2015] authors use the left eigenvectors of $B = SVD(WU)$ (where U is $n \times k$ random matrix), which is the solution of the following optimization problem

$$\min_B \left\|WU - B\Lambda V^\top\right\|^2 \quad \Leftrightarrow \quad \min_B \|WU\|^2 + \left\|B\Lambda V^\top\right\|^2 - 2Tr(WU\Lambda V^\top B^\top).$$

For fixed Λ and V, the above problem is equivalent to

$$\max_B Tr(WU\Lambda V^\top B^\top).$$

Let $\tilde{U} = U\Lambda V^\top$, then this problem is equivalent to our trace maximization problem given in Eq.(4.22).

4.6 Experiments

In the sequel, we discuss the advantages of our contribution and we compare it against a variety of state-of-the-art clustering methods.

1. NMF-based methods including NMF and PNMF.
2. Two-steps-based methods including LDA-k-means, PCA-k-means and Ncut-k-means. These methods are based on two steps which are performed separately. The first step consists in performing an eigendecomposition-based dimensional reduction of the features space and the second step is applying k-means on the first few principal components.

Data Sets. To evaluate all studied methods on clustering, we consider various image real data sets of which the characteristics are summarized in Table 4.1.

Table 4.1 – Real data set characteristics.

Data sets	Type	samples	features	classes
Coil20	Image	1440	1024	20
ORL	Image	400	1024	40
Yale	Image	165	1024	15
PIE	Image	2856	1024	68
USPS	Image	9298	256	10
MNIST	Image	70000	784	10

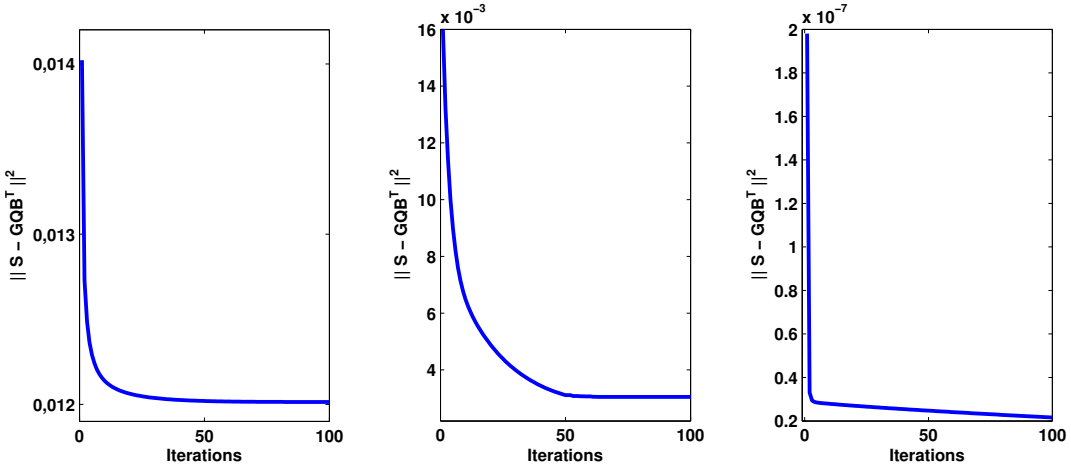
Performance metrics. To measure the clustering performance of the proposed algorithms we use the commonly adopted metrics, the accuracy noted (Acc), the Normalize Mutual Information (NMI) [Strehl and Ghosh, 2002] and the Adjusted Rand Index (ARI) [Hubert and Arabie, 1985]. We only focus on the quality of row clustering. Note that, for these three metrics (Acc, NMI and ARI), a value close to 1 means a good clustering result.

Parameter settings. We run each method under different parameter settings 50 times, and the best, the worse and the average results for each method are computed.

- For all the compared methods, we set the number of sample clusters equal to the true number of classes in each data set (k).
- For NMF and PNMF, the best parameters are used, as suggested in each of the reference articles (see for details [Ding et al., 2005; Zhirong and Laaksonen, 2007].
- For Ncut-k-means, we have used the code of Ncut provided by Zhirong et al. [Zhirong and Laaksonen, 2007]. For, LDA-k-means, we have used the code of LDA provided by Deng Cai [Cai et al., 2006]. Note that LDA is a supervised method, where its components are computed using the partition obtained by k-means rather than the true cluster label that is assumed to be unknown.
- For the eigendecomposition based methods, PSDEC, Ncut-k-means and PCA-k-means, we set the number of components equal to the number of sample clusters k. For LDA-k-means, we set the number of components equal to the number of sample clusters less one ($k-1$).
- For PSDEC, the graph Laplacian is constructed using the Euclidean-distance-based KNN. Also, the neighborhood size is fixed to 5 for the smallest data sets and 10 for the remaining data sets.

4.6.1 Empirical convergence study.

Due to limited space, we will not provide the proof of convergence here. As an illustration, Figure 4.2 shows the empirical convergence behavior of the proposed PSDEC algorithm on some image data sets. We can observe that PSDEC requires few iterations to converge on the five data sets: USPS, PIE and MNIST.



(a) USPS (b) PIE (c) MNIST

Figure 4.2 – Empirical convergence behavior of PSDEC.

4.6.2 Clustering performances.

The assessment of all algorithms in terms of Acc, NMI and ARI are reported in Table 4.2. The main comments arising from our experiments are the following. First, all methods appear almost always better than both k-means in terms of Acc, NMI and ARI. Furthermore, we note that the NMF-based methods including NMF and PNMF give similar results and none among them which outperforms the other. Moreover, the Two-steps based methods including Ncut-k-means, PCA-k-means, LDA-k-means are equivalent with a slight advantage to LDA-k-means. Finally, we can see that PSDEC performs much better than both NMF-based and Two-steps based methods on all real image data sets. We observe that PSDEC allows the best separability between the clusters.

Table 4.2 – Results of compared methods on various image data sets in terms Acc, NMI and ARI. The data sets indicated by (-) are considered small, the neighborhood size is fixed to 5 for the smallest data sets and it is fixed to 10 for the remaining data sets. p^* denotes the value of p optimizing the criterion.

			NMF-based Methods		Two-Steps-based Methods			**Proposed Methods**	
Data set	**Metric**	**k-means**	**NMF**	**PNMF**	**Ncut-k-means**	**PCA-k-means**	**LDA-k-means**	**SDEC** $p=0$	**PSDEC** $p=p^*$
Coil20 (-)	Acc	0.608	0.637	0.676	0.668	0.690	0.698	0.810	0.882
	NMI	0.727	0.746	0.769	0.744	0.760	0.781	0.895	0.965
	ARI	0.550	0.552	0.597	0.604	0.599	0.611	0.750	0.846
ORL (-)	Acc	0.487	0.595	0.603	0.608	0.525	0.583	0.668	0.708
	NMI	0.713	0.750	0.774	0.780	0.737	0.766	0.791	0.827
	ARI	0.318	0.392	0.468	0.469	0.389	0.436	0.499	0.578
Yale (-)	Acc	0.430	0.491	0.442	0.509	0.489	0.497	0.531	0.597
	NMI	0.501	0.526	0.508	0.555	0.521	0.530	0.587	0.635
	ARI	0.227	0.267	0.241	0.296	0.266	0.236	0.324	0.362
PIE	Acc	0.261	0.303	0.274	0.392	0.338	0.381	0.790	0.888
	NMI	0.560	0.558	0.583	0.704	0.660	0.681	0.915	0.956
	ARI	0.156	0.156	0.167	0.247	0.192	0.228	0.803	0.870
USPS	Acc	0.648	0.666	0.873	0.852	0.619	0.840	0.928	0.944
	NMI	0.607	0.608	0.826	0.772	0.548	0.746	0.859	0.876
	ARI	0.521	0.528	0.798	0.730	0.470	0.715	0.862	0.893
MNIST	Acc	0.570	0.636	0.643	0.621	0.532	0.638	0.696	0.759
	NMI	0.582	0.541	0.564	0.540	0.457	0.696	0.621	0.753
	ARI	0.415	0.435	0.443	0.434	0.351	0.581	0.536	0.658

4.6.3 Statistical tests.

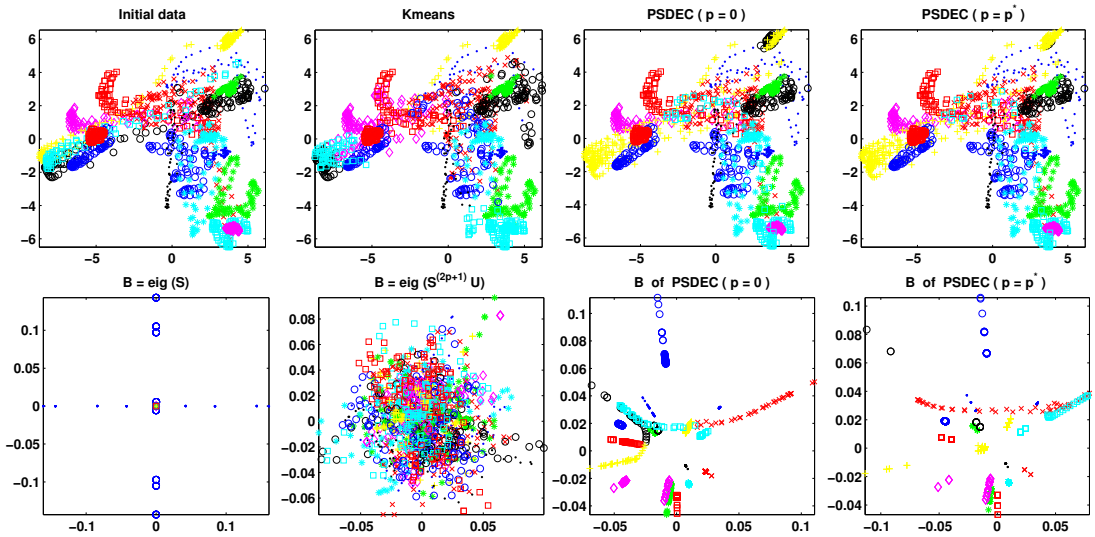
To confirm the performance of PSDEC $(p = 0)$ compared with PSDEC $(p = p^*)$, for each data set, we perform pairwise t-tests on 50 random initialisations. Table 4.3 shows that the improvement is statistically significant; most of the p-values are less than 0.1%.

Table 4.3 – SDEC $(p = 0)$ vs PSDEC $(p = p^*)$: Evaluation on image data sets in terms of Acc, NMI and ARI; using t-tests performed on 50 random initialisations. p^* denotes the value of p optimizing the criterion.

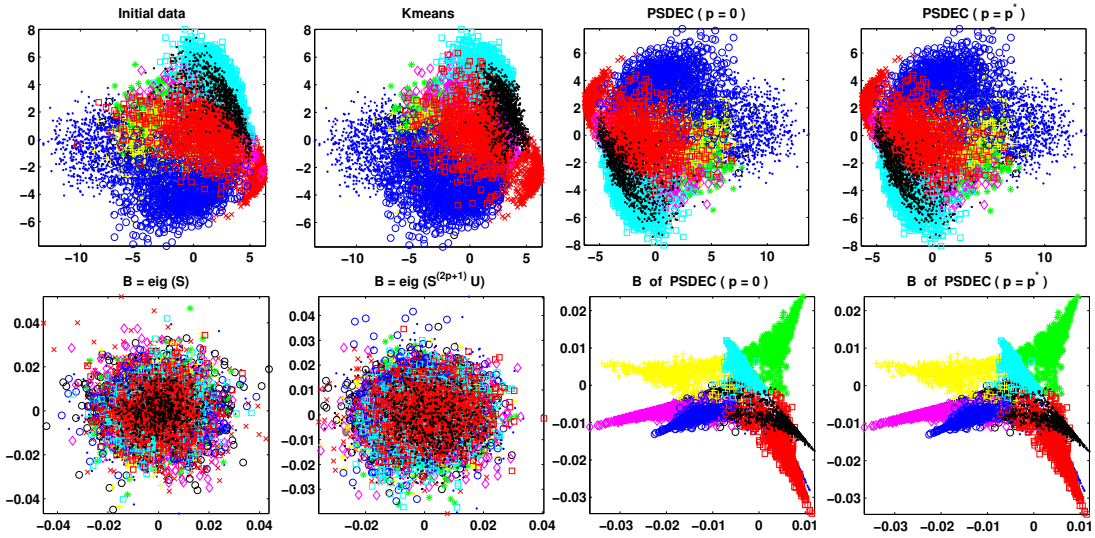
Data set	Metric	PSDEC $(p = 0)$				PSDEC $(p = p^*)$				P-values
		Max	Min	Mean	std	Max	Min	Mean	std	
Coil20 (-)	Acc	0.810	0.606	0.673	0.045	0.882	0.859	0.872	0.004	$< 0.1\%$
	NMI	0.895	0.788	0.837	0.026	0.965	0.945	0.947	0.001	$< 0.1\%$
	ARI	0.750	0.516	0.608	0.057	0.846	0.811	0.824	0.005	$< 0.1\%$
ORL (-)	Acc	0.695	0.685	0.688	0.003	0.708	0.703	0.706	0.002	$< 0.1\%$
	NMI	0.822	0.810	0.815	0.005	0.827	0.826	0.827	0.000	$< 0.1\%$
	ARI	0.552	0.528	0.538	0.010	0.578	0.566	0.575	0.005	$< 0.1\%$
Yale (-)	Acc	0.531	0.470	0.489	0.014	0.597	0.594	0.595	0.001	$< 0.1\%$
	NMI	0.587	0.557	0.571	0.013	0.635	0.633	0.634	0.001	$< 0.1\%$
	ARI	0.324	0.282	0.298	0.015	0.362	0.358	0.360	0.002	$< 0.1\%$
PIE	Acc	0.790	0.671	0.708	0.028	0.888	0.873	0.886	0.002	$< 0.1\%$
	NMI	0.915	0.862	0.871	0.012	0.956	0.952	0.955	0.001	$< 0.1\%$
	ARI	0.803	0.742	0.779	0.022	0.870	0.856	0.868	0.002	$< 0.1\%$
USPS	Acc	0.928	0.761	0.807	0.039	0.944	0.930	0.942	0.002	$< 0.1\%$
	NMI	0.859	0.804	0.814	0.016	0.876	0.855	0.875	0.003	$< 0.1\%$
	ARI	0.862	0.740	0.770	0.029	0.893	0.865	0.892	0.005	$< 0.1\%$
MNIST	Acc	0.696	0.584	0.622	0.028	0.759	0.699	0.721	0.010	$< 0.1\%$
	NMI	0.621	0.573	0.581	0.015	0.753	0.721	0.739	0.008	$< 0.1\%$
	ARI	0.536	0.464	0.482	0.020	0.658	0.611	0.628	0.013	$< 0.1\%$

4.6.4 Real data visualization.

Like in Figure 4.1, we present in Figure 4.3 the obtained clustering results of both k-means, SDEC and PSDEC on some representative real data sets. Using the first two principal components, we can observe the embedding matrix B which is updated and exploited by the approximation process of PSDEC; it reveals a better visualization of clusters. We can note that the clustering result of k-means is not satisfactory for all data sets where we note several misclassified objects. However, thanks to the embedding matrix B which is the low-dimensional data representation obtained by our proposed methods, we can observe that PSDEC allows the best separability between the clusters on both synthetic and real data sets.



(a) Coil20



(b) USPS

Figure 4.3 – Clustering and visualization with PSDEC ($p = 0$), PSDEC ($p = p^*$) and k-means. p^* denotes the value of p optimizing the criterion.

4.7 Conclusion

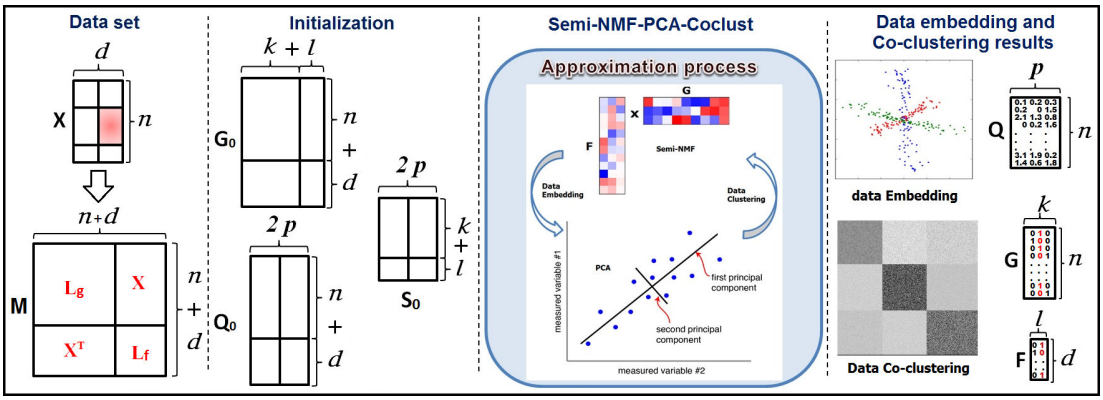
The dual purpose of this paper is spectral embedding and clustering. Based on the decomposition of the objective function of PSDEC into two terms where the first one is the objective function of spectral embedding while the second is the Semi-NMF criterion in a low embedded space; we proposed a novel way to consider clustering and the reduction of the dimension simultaneously.

The proposed PSDEC approach performs on a stochastic powered matrix; the purpose of the use of such matrix is twofold. First, it allows to the use of power method inside our algorithm in order to speed up the eigenvectors computation, and secondly to explore the similarity matrix structure via a random walk process and then make the similarity matrix more suitable for the clustering task (quasi block diagonal matrix). As shown in the experiments section, this usually leads to a better embedding approximation and an improvement in clustering accuracy.

In other words, our approach takes advantage of the mutual reinforcement between data spectral embedding and clustering tasks. Such a solution approximates better the relaxed continuous spectral embedding solution by the good obtained partitions. More interestingly, we established theoretical connections between PSDEC and some well known approaches like spectral matrix decomposition, semi-NMF, Procrustes problem and Power method. This explains the performance improvement.

Chapter 5

Simultaneous Sparse Data Embedding and Co-clustering



5.1 Introduction

In this chapter we proposed a novel way to consider simultaneously the co-clustering and the reduction of the dimension. Our approach takes advantage of the mutual reinforcement between Principal Component Analysis (PCA) which provide a low-dimensional representation of data and Semi-Nonnegative Matrix Factorization (Semi-NMF) that learns this low-dimensional representation and lends itself to a co-clustering interpretation. In other words, the proposed framework aims to find an optimal subspace of multi-dimensional variables for identifying a partition of the set of objects. We show that by doing so, our model is able to learn low-dimensional representations that are better suited for co-clustering, outperforming not only spectral methods, but also co-clustering graph-regularized-based methods. Specifically:

- Unlike to known methods that combine the objective function of PCA and the objective function of k-means separately, we propose a new single framework to perform SemiNMF via PCA for dimension reduction and data co-clustering.
- We show that the objective learning of SemiNMF-PCA-Coclust can be decomposed into two terms, the first one is the objective function of PCA and the second is the Semi-NMF criterion in a low dimensional space. This allows a better approximation of data reduction by a co-clustering solution.
- We further developed our method to incorporate manifold information of both data samples and data features and proposed the graph Regularized SemiNMF-PCA-Coclust method.

In this chapter, we focused our experiments on the sparse document-term databases. The objective of co-clustering is to simultaneously group documents and terms into meaningful clusters. This ability to ”simplify” the data can be used to automatically annotate sets of documents with suitable descriptive terms, thus having applications for automatic indexing, information retrieval as well as exploratory visualization of large document corpuses.

This problem attracted many authors these last years even this problem has been first investigated by Hartigan [1972]. In [Govaert, 1995], the author proposed three algorithms to perform co-clustering on several kinds of data, namely contingency table, binary and continuous data. For continuous data, the author developed the *Croeuc* algorithm which consists in using the principle of double k-means. Later, Dhillon [2001a] proposed the bipartite spectral graph partitioning algorithm referred as (*Spec*) in the sequel. This algorithm has some nice theoretical properties; the singular vectors solve a real relaxation to the NP-complete graph bipartitioning problem. Other algorithms for co-clustering document-term matrices have been proposed in the following years. For example, Li [2005] has proposed a block diagonal clustering algorithm that, given a binary document-term matrix, produces a block diagonal matrix

of ones. This algorithm consists in alternating the clustering of rows and columns minimizing the squared error between the original X data and its approximation $AB^\top$ where A and B are binary matrices. It is worth mentioning the Information-Theoretic Co-clustering (ITCC) algorithm [Dhillon et al., 2003b] which uses as criterion the minimization of the difference in mutual information between the original document-term matrix and the aggregated matrix. More recently, a new direction has been proposed which rely on the maximization of an adapted version of the graph modularity used for community detection. For example, Labiod and Nadif [2011] proposed a co-clustering algorithm maximizing a bipartite modularity by using a spectral approach. According to the experiments carried out by the authors, the developed *Speco* algorithm appears to perform well in the field of document clustering compared to other binary clustering methods based on NMF or NMTF.

Despite the popularity of factorization-based co-clustering methods, one drawback is that they rest on only a global Euclidean geometry, and a local manifold geometry is not fully considered. To address this major limitation, some researchers have sought to take into account a local geometrical structure in matrix-factorization-based co-clustering. For instance, Gu and Zhou [2009] proposed the Dual Regularized Co-Clustering (DRCC) method, which inherits the advantages of NMTF and, in addition, takes into account the manifold structures in both sample and feature spaces. However, when dealing with certain types of data that contain negative values, DRCC is of limited applicability. In addition, its high computational complexity usually makes DRCC unsuitable for large-scale problems. As a consequence of this, in [Wang et al., 2011a] the authors proposed the Locally Preserved Fast Nonnegative Matrix Tri-Factorization algorithm (LpFNMTF), constraining the factors to be cluster indicator matrices and reducing the computational cost of the eigendecomposition of the graph Laplacian.

The rest of chapter is organized as follows. Section 2 introduces the co-clustering problem and the dimension reduction in factorization framework. Section 3 provides a sound SemiNMF-PCA-Coclust framework for co-clustering. Section 4 focuses on some details concerning the proposed Graph Regularized SemiNMF-PCA-Coclust algorithm. Sections 5 and 6 are devoted to numerical experiments on some real document-term data sets. Finally, the conclusion summarizes the advantages of our contribution.

5.2 SemiNMF-PCA Co-Clustering

Let $X = (x_{ij})$ be a $(n \times d)$ positive data matrix; we assume that X is provided by a collection of n data row vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, each with d features. Let k be the number of the sample clusters, ℓ the number of feature clusters and p the number of components to which the features are reduced. We put $nd = n + d$, $k\ell = k + \ell$.

Inspired by Dhillon [2001a], we apply co-clustering to document-term matrices and pose the co-clustering problem as a bipartite graph partitioning problem. SemiNMF-PCA co-clustering is defined as the minimizing problem of the following criterion:

$$\min_{G,S,Q} \left\| M - GSQ^\top \right\|^2 \quad \text{s.t.} \quad G \geq 0, Q^\top Q = I. \tag{5.1}$$

where $\|.\|$ denotes the Frobenius norm. The matrix M of size $(nd \times nd)$ can be written in the form of block matrix as

$M = \begin{bmatrix} 0 & X \\ X^\top & 0 \end{bmatrix}$, where X is the document-term matrix.

The matrix G of size $(nd \times k\ell)$, S of size $(k\ell \times 2p)$ and Q of size $(nd \times 2p)$ are defined as follows:

$$G = \begin{bmatrix} G_g^{(n\times k)} & 0 \\ 0 & G_f^{(d\times \ell)} \end{bmatrix}, \quad Q = \begin{bmatrix} Q_g^{(n\times p)} & 0 \\ 0 & Q_f^{(d\times p)} \end{bmatrix} \text{ and } S = \begin{bmatrix} S_g^{(k\times p)} & \theta_g^{(k\times p)} \\ \theta_f^{(\ell\times p)} & S_f^{(\ell\times p)} \end{bmatrix}.$$

where G_g, and G_f are the label matrices obtained by applying k-means on X and $X^\top$ respectively. $S_g = (s_{k'p})$ and $S_f = (s_{\ell' p})$ are centroid matrices while $\mathbf{s}_{g_{k'}}$ is a centroid of the $(k')^{th}$ sample cluster for each $k' = 1, \ldots, k$ and $\mathbf{s}_{f_{\ell'}}$ is a centroid of the $(\ell')^{th}$ feature cluster for each $\ell' = 1, \ldots, \ell$.

Note that Q_g, Q_f are initialized using Singular Value Decomposition (SVD). Applying the full SVD to X, we obtain $X = U\Sigma V^\top$. Truncating U, Σ and V to the first p singular components and arbitrarily splitting the singular values between the left and right factors yields an optimal rank $-p$ approximation of X in the least-squares sense. In fact, U of size $(n \times p)$ and V of size $(d \times p)$ are orthonormal matrices and Σ of size $(p \times p)$ diagonal containing the p non-negative singular values of X in non-increasing order on its diagonal. Now we simply put $Q_g = U\Sigma$ and $Q_f = V\Sigma$.

5.3 Regularized SemiNMF-PCA for Co-Clustering

Recent research has shown that existing co-clustering algorithms fail to consider the intrinsic geometric structure in the data which is essential for data clustering on manifolds [Gu and Zhou, 2009; Wang et al., 2011b]. Furthermore, we known that PCA provides an embedding for the data lying on a linear manifold. However, in many applications, data lie in a non-linear manifold.

One popular method is to use the graph Laplacian based embedding. Laplacian embedding [Belkin and Niyogi, 2001; Zhang and Zha, 2004] preserves the local geometrical relationships and maximizes the smoothness with respect to the intrinsic manifold of the data set in the low-embedding space. We now formulate the Regularized SemiNMF-PCA co-clustering (R-SemiNMF-PCA-Coclust) problem in the light of this considerations.

We first construct a K-nearest neighbor data graph whose vertices correspond to the n data samples $[\mathbf{x}_1, \ldots, \mathbf{x}_n]$. We use the 01 weighting scheme to construct the K-nearest neighbor graph, and define the data weight matrix W_g as follows,

$$W_{g_{(ij)}} = \begin{cases} 1, & \text{if} \quad \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i); \quad i, j = 1, \cdots, n;\ i \neq j. \\ 0, & \text{otherwise} \end{cases}$$

where $\mathcal{N}(\mathbf{x}_i)$ represents the set of K-nearest neighbors of $\mathbf{x}_i$. Then, we define the weight matrix W_f from the k-nearest neighbor graph whose vertices correspond to the d data features, in an analogous way. Next, we simply compute the normalized graph Laplacians L_g and L_f, respectively.

$$L_g = D_g^{-\frac{1}{2}} W_g D_g^{-\frac{1}{2}} \quad \text{and} \quad L_f = D_f^{-\frac{1}{2}} W_f D_f^{-\frac{1}{2}}$$

where D_g, D_f are diagonal matrices the entries of which are row sums of the matrices W_g and W_f respectively. Introducing the normalized graph Laplacians L_g and L_f in the matrix M we obtain

$$M = \begin{bmatrix} \alpha L_g & X \\ X^\top & \beta L_f \end{bmatrix}$$

where α and β are the to trade-off parameters used to govern the contribution of L_g and L_f, respectively. The Regularized SemiNMF-PCA co-clustering optimization problem (5.1) becomes:

$$\min_{G,S,Q} \left\| M - GSQ^\top \right\|^2 \quad \text{s.t.} \quad G \geq 0, Q^\top Q = I. \tag{5.2}$$

To solve (5.2), we rely on the following proposition

Proposition 4. *Given $G \geq 0$ and $Q^\top Q = I$, the objective function of R-SemiNMF-PCA-Coclust can be decomposed into two terms:*

$$\left\| M - GSQ^\top \right\|^2 = \left\| M - MQQ^\top \right\|^2 + \|MQ - GS\|^2 \tag{5.3}$$

Proof. We expand the matrix norm of the left term of Eq. (5.3)

$$\left\|M - GSQ^\top\right\|^2 = \|M\|^2 + \left\|GSQ^\top\right\|^2 - 2Tr(M^\top GSQ^\top)$$

In a similar way, we obtain from the two terms of the right term of Eq. (5.3)

$$\begin{aligned} \left\|M - MQQ^\top\right\|^2 &= \|M\|^2 + \left\|MQQ^\top\right\|^2 - 2Tr(MQQ^\top M^\top) \\ &= \|M\|^2 + \left\|MQQ^\top\right\|^2 - 2||MQ||^2 \\ &= ||M||^2 - ||MQ||^2, \quad \text{due to } Q^\top Q = I \\ \text{and} \quad \|MQ - GS\|^2 &= \|MQ\|^2 + \|GS\|^2 - 2Tr(M^\top GSQ^\top) \end{aligned} \tag{5.4}$$

Due also to $Q^\top Q = I$, we have

$$\|MQ - GS\|^2 = ||MQ||^2 + ||GSQ^\top||^2 - 2Tr(M^\top GSQ^\top) \tag{5.5}$$

Summing the two terms Eq. (5.4) and Eq. (5.5) leads to the left term of Eq. (5.3).

$$\|M\|^2 + \|GS\|^2 - 2Tr(M^\top GSQ^\top) = \left\|M - GSQ^\top\right\|^2 \tag{5.6}$$

□

Using proposition 4, the objective function of R-SemiNMF-PCA-Coclust (5.2) can be decomposed into two terms: the first one is the objective function of PCA, and the second is the SemiNMF criterion in a low-dimensional subspace.

5.4 Optimization

To solve (5.2), we use an alternated iterative method.

Computation of S. First, fixing G and Q, by setting the derivative of the second term in (5.2) with respect to S as 0, we obtain:

$$S = (G^\top G)^{-1} G^\top MQ \tag{5.7}$$

Computation of Q. Secondly, fixing G and S, we can rewrite (5.2) as:

$$\min_{Q^\top Q = I} \left\|M - BQ^\top\right\|^2 \text{ where } B = GS. \tag{5.8}$$

To solve (5.8) we rely on the Theorem 1. Due to Theorem 1, applying SVD to $M^\top B$ we obtain the expression of $Q = UV^\top$.

Computation of G. Thirdly, update G by keeping S and Q fixed at the value computed in the above steps, as in [Ding et al., 2006b] we obtain

$$G = G \circ \sqrt{\frac{[MH^\top]^+ + G[HH^\top]^-}{G[HH^\top]^+ + [MH^\top]^-}} \tag{5.9}$$

where $H = SQ^\top$, A^+ and A^- correspond respectively to positive and negative parts of the matrix A given by

$$A^+_{ik} = \frac{1}{2}(|A_{ik}| + A_{ik}) \quad \text{and} \quad A^-_{ik} = \frac{1}{2}(|A_{ik}| - A_{ik})$$

In summary, the steps of the R-SemiNMF-PCA-Coclust can be deduced in Algorithm 8.

Algorithm 8: R-SemiNMF-PCA-Coclust algorithm.

Input: Data matrix X, k, l and p.
Initialize: - G using k-means, Q using SVD.
Step 1 : Compute the normalized graph Laplacians L_g and L_f.
Step 2 : Construct the matrix M.
repeat
  (a) - Update S by Eq. (5.7);
  (b) - Update G by Eq. (5.9)
  (c) - Update Q by solving Eq. (5.8)
until *convergence*;
Output: - Sample indicator matrix $G_g = G[1..n, 1..k]$
- Feature indicator matrix $G_f = G[n+1..nd, k+1..k\ell]$
- Sample embedding matrix $Q_g = Q[1..n, 1..P]$

5.5 Experiments

The series of experiments presented in this section is devoted to studying the behavior and performance of *R-SemiNMF-PCA-CoClust*, and comparing it with other algorithms commonly used for revealing block and homogeneous co-clusters in the document clustering context. The competitive algorithms retained for comparison with *R-SemiNMF-PCA-CoClust* are Croeuc [Govaert, 1983], Spec [Dhillon, 2001a], ITCC [Dhillon et al., 2003b], SpecCo [Labiod and Nadif, 2011], DRCC [Gu and Zhou, 2009], FNMTF and LpFNMTF [Wang et al., 2011b].

Performance metrics. To measure the clustering performance of the proposed algorithms we use the commonly adopted external metrics, the accuracy, the Normalize Mutual Information [Strehl and Ghosh, 2002] and the Adjusted Rand Index [Hubert and Arabie, 1985]. We focus only on the quality of row clustering. For these three metrics (Acc, NMI and ARI), a value close to 1 means a good clustering result.

Data sets. The experiments were performed using some benchmark Document-term data sets from the clustering literature. Table 5.1 summarizes the characteristics of these data sets [1].

Table 5.1 – Description of document-term Data sets.

Data sets	**Characteristics**				
	#Documents	#Terms	#Clusters	Sparsity (%)	Balance
CSTR	475	1000	4	96.60	0.399
WebACE	2340	1000	20	91.83	0.169
NG20	19949	43586	20	99.99	0.991
RCV1	9625	29992	4	99.75	0.766
Reviews	4069	18483	5	99.99	0.098
Sports	8580	14870	7	99.99	0.036
Classic3	3891	4303	3	98.0	0.710
Classic4	7095	5896	4	99.41	0.323

Note that, for all the used document-term data sets, we apply the TF-IDF transformation on all the document-term frequency matrices. We used the TF-IDF weighting scheme proposed in scikit-learn [Pedregosa et al., 2011].

Parameter settings. We run each method under different parameter settings 50 times and we report the best result for each method. We set the number of sample clusters equal to the true number of classes (k) for all the data sets. Also, the number of feature clusters (ℓ) is set to be the same as the number of sample clusters.

- For all the compared methods, we use *spherical k-means* (Sk-means) [Dhillon, 2001a] to initialize the sample partition matrix G according the type of data. Furthermore, the best parameters are used, as suggested in each of the reference articles (see for details [Dhillon, 2001a; Dhillon et al., 2003b; Govaert, 1983; Gu and Zhou, 2009; Labiod and Nadif, 2011; Wang et al., 2011b]).
- For graph-regularized-based methods, DRCC, LPFNMTF and R-SemiNMF-PCA-CoClust, the graph Laplacian matrix is constructed using the cosine-distance-based K-Nearest Neighbors in which the neighborhood size is fixed to 5 for the smallest data sets and 10 for the remaining data sets.

[1] The balance coefficient is defined as the ratio of the number of documents in the smallest class to the number of documents in the largest class.

- For R-SemiNMF-PCA-CoClust, we varied the number of components p between 2 and k and retained the one that optimizes the criterion. Furtehermore, the regularization parameter α is searched from the grid (0.01, 0.1, 1, 10, 100, 500, 1000). Also, we set $\beta = 0.1\alpha$.

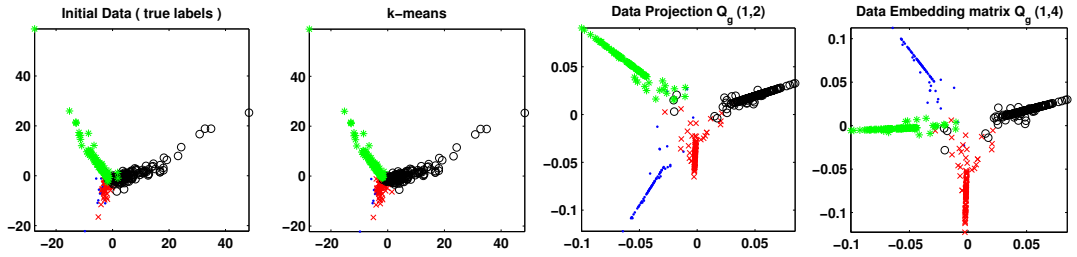
5.5.1 Global Performance Evaluation

The results reported in table 5.2 were obtained by running each algorithm 50 times with random initialization. We retained the solution optimizing the associated criterion. It clearly appears from the results reported in table 5.2 that R-SemiNMF-PCA-CoClust outperforms all the other compared algorithms most of the time.

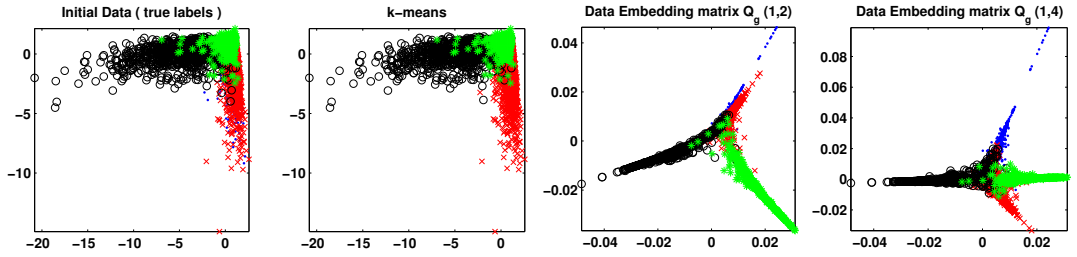
Next, we present in Fig. 5.1 the obtained clustering results of both k-means and R-SemiNMF-PCA-CoClust on some representative real data sets. Furthermore, using the first p principal components for CSTR (best $p = 4$), Classic4 (best $p = 4$), Reviews (best $p = 3$) and RCV1 (best $p = 3$) data sets, we can observe the embedding matrix B which is updated and exploited by the approximation process of R-SemiNMF-PCA-CoClust; it reveals a better separability of clusters.

Table 5.2 – Co-clustering on several data sets. The neighborhood size is fixed to 5 for the smallest data sets indicated by (-) and it is fixed to 10 for the remaining data sets.

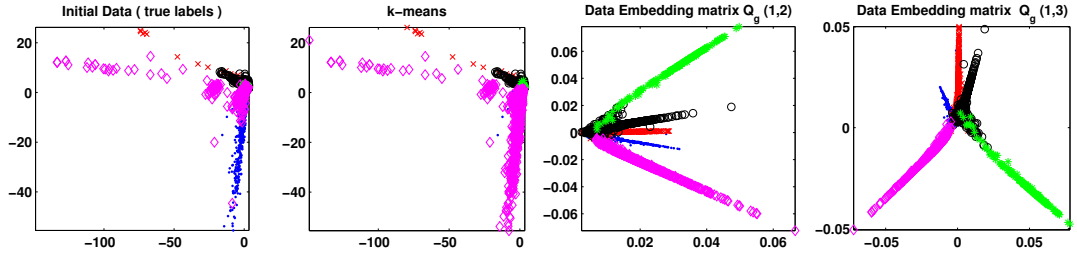
Data sets	Metric	Croeuc	Spec	ITCC	SpecCo	DRCC	FNMTF	LpFNMTF	R-SemiNMF-PCA -Coclust
CSTR (-)	Acc	0.787	0.838	0.663	0.903	0.865	0.903	0.907	**0.924**
	NMI	0.688	0.696	0.672	0.771	0.702	0.778	0.805	**0.831**
	ARI	0.634	0.732	0.578	0.810	0.719	0.810	0.833	**0.862**
WEBACE (-)	Acc	0.545	0.394	0.554	0.542	0.603	0.526	0.639	**0.651**
	NMI	0.614	0.532	0.684	0.635	0.633	0.619	0.646	**0.669**
	ARI	0.456	0.361	0.463	0.440	0.513	0.475	0.609	**0.613**
NG20	Acc	0.548	0.195	0.448	0.375	0.392	0.502	0.556	**0.665**
	NMI	0.547	0.328	0.523	0.464	0.400	0.516	0.533	**0.635**
	ARI	0.408	0.163	0.334	0.262	0.159	0.314	0.347	**0.504**
RCV1	Acc	0.681	0.309	0.673	0.396	0.706	0.553	0.619	**0.752**
	NMI	0.477	0.012	0.440	0.036	0.468	0.304	0.358	**0.517**
	ARI	0.441	0.005	0.408	0.112	0.455	0.282	0.363	**0.509**
REVIEWS	Acc	0.434	0.527	0.711	0.580	0.720	0.510	0.610	**0.750**
	NMI	0.291	0.312	0.569	0.449	0.527	0.373	0.379	**0.657**
	ARI	0.162	0.197	0.589	0.411	0.533	0.264	0.324	**0.638**
SPORTS	Acc	0.486	0.564	0.558	0.613	0.565	0.457	0.653	**0.704**
	NMI	0.316	0.481	0.579	0.659	0.569	0.369	0.552	**0.691**
	ARI	0.178	0.375	0.394	0.471	0.378	0.203	0.461	**0.621**
CLASSIC3 (-)	Acc	0.909	0.832	0.986	0.905	0.981	0.922	0.979	**0.992**
	NMI	0.775	0.717	0.931	0.771	0.909	0.755	0.905	**0.954**
	ARI	0.838	0.724	0.959	0.813	0.942	0.787	0.940	**0.974**
CLASSIC4 (-)	Acc	0.763	0.756	0.722	0.563	0.599	0.686	0.743	**0.772**
	NMI	0.619	0.675	0.593	0.342	0.579	0.470	0.585	**0.686**
	ARI	0.511	0.526	0.445	0.301	0.449	0.358	0.454	**0.541**



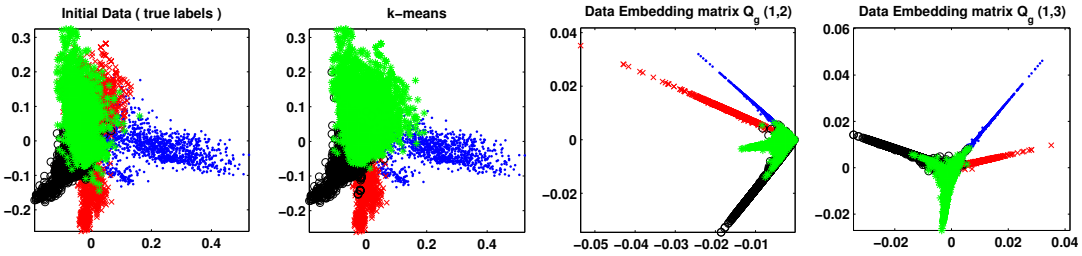
(a) CSTR data set



(b) Classic4 data set



(c) Reviews data set



(d) RCV1 data set

Figure 5.1 – Visualization of the obtained clustering results of both k-means and R-SemiNMF-PCA-CoClust. Visualization of $B(a, b)$: the two selected first principal components a and b of the embedding matrix B obtained by R-SemiNMF-PCA-CoClust.

5.5.2 Statistical tests.

The first question attempted to see if there were any significant differences among R-SemiNMF-PCA-CoClust compared with LpFNMTF?

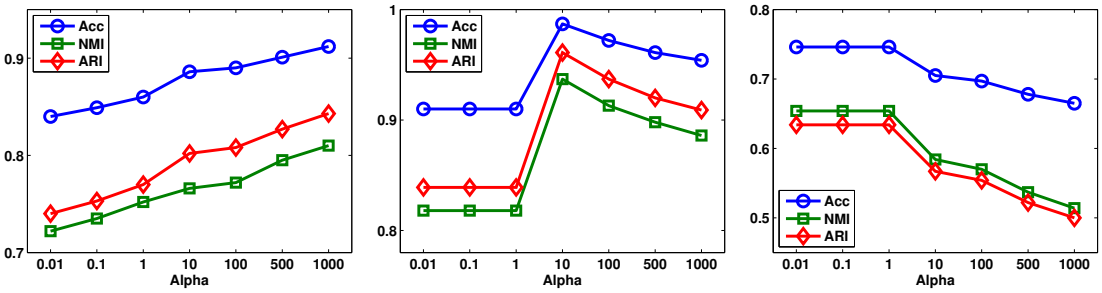
To this end, we first test for the significance of performance differences between R-SemiNMF-PCA-CoClust and LpFNMTF. For each data set, we used analysis of variance (One-way ANOVA) test. We notice that all of the ANOVA p-values are less than 5%. Next, to confirm the performance of R-SemiNMF-PCA-CoClust compared with LpFNMTF, we perform pairwise t-tests. As it can be seen in table 5.3, the improvement is statistically significant; most of the t-test p-values are less than 5%.

Table 5.3 – R-SemiNMF-PCA-CoClust vs LpFNMTF: Evaluation in terms of Acc, NMI and ARI; using t-tests performed on 50 random initialisations.

Data sets	Metric	LpFNMTF	R-SemiNMF-PCA-Coclust	P-values (T-test)
CSTR	Acc	0.885 $\pm$0.013	0.909 $\pm$0.005	$1.193e-20$
	NMI	0.757 $\pm$0.030	0.790 $\pm$0.013	$2.484e-10$
	ARI	0.781 $\pm$0.031	0.828 $\pm$0.010	$1.689e-16$
WEBACE	Acc	0.604 $\pm$0.037	0.622 $\pm$0.015	0.002
	NMI	0.608 $\pm$0.019	0.655 $\pm$0.008	$2.756e-08$
	ARI	0.522 $\pm$0.048	0.543 $\pm$0.037	0.001
NG20	Acc	0.494 $\pm$0.023	0.583 $\pm$0.035	$2.776e-27$
	NMI	0.496 $\pm$0.016	0.599 $\pm$0.015	$5.796e-61$
	ARI	0.312 $\pm$0.020	0.444 $\pm$0.023	$1.474e-51$
RCV1	Acc	0.550 $\pm$0.030	0.747 $\pm$0.004	$3.458e-68$
	NMI	0.225 $\pm$0.063	0.504 $\pm$0.007	$1.950e-52$
	ARI	0.270 $\pm$0.030	0.500 $\pm$0.005	$1.431e-74$
REVIEWS	Acc	0.457 $\pm$0.042	0.678 $\pm$0.039	$6.782e-45$
	NMI	0.271 $\pm$0.040	0.581 $\pm$0.039	$1.570e-58$
	ARI	0.171 $\pm$0.048	0.556 $\pm$0.053	$1.238e-55$
SPORTS	Acc	0.549 $\pm$0.058	0.647 $\pm$0.026	$2.346e-17$
	NMI	0.464 $\pm$0.051	0.650 $\pm$0.025	$5.504e-41$
	ARI	0.348 $\pm$0.064	0.505 $\pm$0.037	$1.111e-26$
CLASSIC3	Acc	0.977 $\pm$0.002	0.991 $\pm$0.001	$1.783e-18$
	NMI	0.900 $\pm$0.002	0.953 $\pm$0.001	$1.346e-12$
	ARI	0.937 $\pm$0.001	0.973 $\pm$0.001	$2.366e-12$
CLASSIC4	Acc	0.729 $\pm$0.010	0.738 $\pm$0.009	$3.021e-06$
	NMI	0.559 $\pm$0.016	0.671 $\pm$0.006	$7.976e-68$
	ARI	0.438 $\pm$0.015	0.472 $\pm$0.013	$3.730e-20$

5.5.3 Study on regularization parameters α and β

The choice of parameters α and β is not easy. However, through our experiments, we can give indications on the appropriate values to be taken for these two parameters. In Figure 5.2, are reported the performances of our algorithm in terms of Acc, NMI and ARI according values of parameters α and β varying in the interval 0.001 to 1000, we took $\alpha = 0.1\beta$.



(a) CSTR data set (b) Classic3 data set (c) Reviews data set

Figure 5.2 – Co-clustering quality for different values of α and β.

For CSTR data set, the performance of R-SemiNMF-PCA-Coclust, grows with α and β, it is the best when α and β are higher ($\alpha = 1000$). Furthermore, we note that, for Classic3 data set, the best performances are obtained with ($\alpha = 10$). However, for Reviews data set, small values of α and β appear more interesting. In fact, the number of data features increases the possibility of the presence of noise. It seems an important element in the choice of α and β. The same observations are verified from the remaining data sets.

5.6 Co-clustering on Pubmed Data

In the preceding sections, we evaluated the document clusters produced by the algorithm. However, R-SemiNMF-PCA-Coclust, as a co-clustering algorithm, also produces corresponding term clusters. It is difficult to assess the quality of these clusters since, contrary to what is the case for documents, we do not have gold standard labels for terms. However, in this section we demonstrate that the term clusters obtained with R-SemiNMF-PCA-Coclust are really meaningful and in good agreement with the corresponding document clusters.

To do so, we use the 10PUBMED data set used in [Chen et al., 2009]. This data set contains more than 15,500 MEDLINE biomedical medical abstracts from Medline database, partitioned across 10 different diseases and published between 2000 and 2008. After pre-processing, the authors obtained a document-terms matrix of the size (15565×22437). The list of diseases is presented in table 5.4. Furthermore, we have used two other variants of 10PUBMED. In PUBMED6 data set, we retain the six largest classes including Migraine (3703 documents), Age-related Macular Degeneration (3283 documents), Otitis (2596 documents), Kidney Calculi (1549 documents), Hay Fever (1517 documents) and Hepatitis A (796 documents). We removed all terms that do not appear in at least one document. Similarly, in PUBMED5 data set, we retain only the five largest classes.

Table 5.4 – Disease clusters in the PUBMED10 data set

Disease class	**Number of documents**
Migraine	3703
Age-related Macular Degeneration	3283
Otitis	2596
Kidney Calculi	1549
Hay Fever	1517
Hepatitis A	796
Chickenpox	732
Gout	543
Jaundice	503
Raynaud Disease	343

Our objective is to illustrate that the corresponding terms of each disease cluster are generally meaningful and can be used to describe the document clusters. We apply the compared methods to co-cluster PUBMED10, PUBMED6 and PUBMED5 data sets. Table 5.5 shows the obtained results. In addition, a reorganisation of the tree matrices according to the co-clusters obtained by our method are illustrated in figure 5.3. Finally, the 10 top terms PUBMED5 data set are presented in Table 5.6. The top 10 terms of each co-cluster (numbered in figure 5.3 (a)) were obtained by keeping only the terms that appear in most documents in the considered cluster.

Table 5.5 – Co-clustering methods on PUBMED data sets

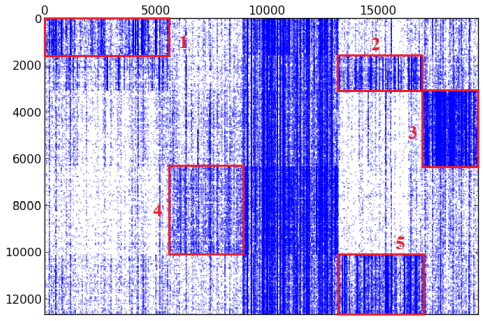
Data sets	Metric	Croeuc	Spec	ITCC	SpecCo	DRCC	FNMTF	LpFNMTF	R-SemiNMF-PCA -Coclust
Pubmed5	Acc	0.343	0.292	0.755	0.388	0.868	0.483	0.894	**0.987**
	NMI	0.068	0.007	0.671	0.170	0.755	0.316	0.758	**0.949**
	ARI	0.056	0.001	0.684	0.116	0.727	0.196	0.774	**0.965**
Pubmed6	Acc	0.297	0.277	0.674	0.433	0.760	0.480	0.810	**0.838**
	NMI	0.053	0.008	0.625	0.182	0.701	0.301	0.772	**0.810**
	ARI	0.052	0.001	0.598	0.116	0.657	0.190	0.728	**0.758**
Pubmed10	Acc	0.212	0.471	0.645	0.429	0.646	0.549	0.696	**0.716**
	NMI	0.039	0.390	0.710	0.166	0.679	0.579	0.712	**0.777**
	ARI	0.032	0.181	0.613	0.122	0.596	0.433	0.627	**0.674**

Table 5.6 – the 10 top Terms of the obtained Clusters on PUBMED5 data set

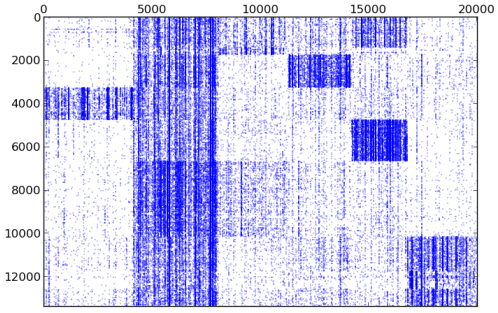
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
allerg	stone	macular	aura	otiti
rhiniti	renal	ey	migraineur	ear
nasal	kidnei	visual	famili	media
pollen	percutan	amd	mechan	children
season	calculi	retin	neurolog	middl
allergen	lithotripsi	acuiti	receptor	antibiot
asthma	shock	degen	pathophysiolog	effu
allergi	uret	neovascular	cortic	aom
immunotherapi	nephrolithotomi	intravitr	hemipleg	om

The main comments arising from our experiments are the following.

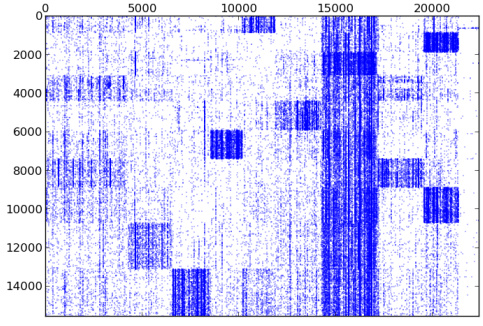
- We have analyzed the five most homogeneous blocks which correspond to the obtained five sample clusters. Since our method is not diagonal, the overlap is not only tolerable, but in addition, it is beneficial. For example, there are many common terms between Kidney Calculi and Otitis disease's abstracts. This explains why the block 2 and the block 5 are overlapped in term of features, although the obtained top terms are different.
- The results show that the column clusters are semantically coherent and highly indicative of the document clusters: Hay Fever (Cluster 1), Kidney Calculi (Cluster 2), Age-related Macular Degeneration (Cluster3), Migraine (Cluster 4) and Otitis (Cluster 5).
- In Figures 5.3(a),(b) and (c), the dense band of variables are the terms cited in the majority of the documents and that can be considered as noise. This did not affect the co-clustering process and has not prevented a correct classification of documents and terms.



(a) Pubmed5 reorganized data.



(b) Pubmed6 reorganized data.



(c) Pubmed10 reorganized data.

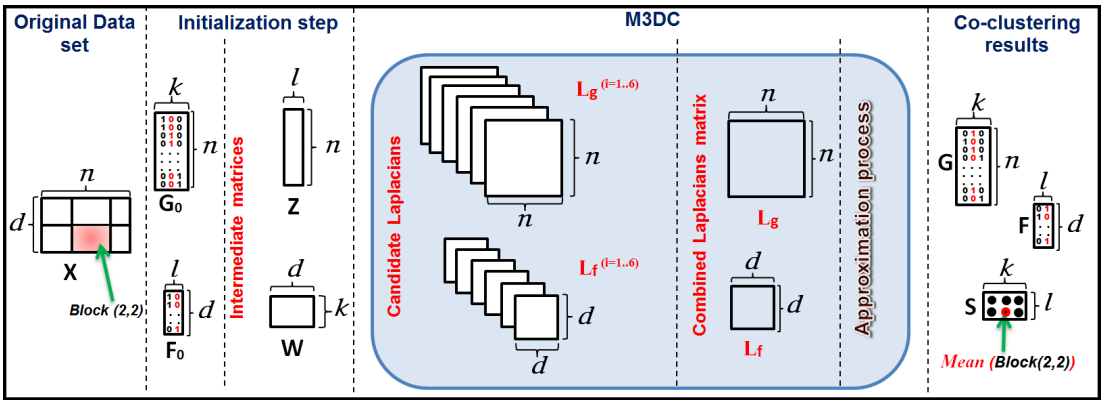
Figure 5.3 – Co-clustering results on PUBMED data sets.

5.7 Conclusion

The dual purpose of this paper is reducing the dimension and co-clustering. Based on the decomposition of the objective function of SemiNMF-PCA-Coclust into two terms where the first one is the objective function of PCA and the second is the SemiNMF criterion in a low-dimensional space, we proposed a novel way to consider simultaneously the Co-clustering and the reduction of the dimension. Our approach takes advantage of the mutual reinforcement between data reduction and Co-clustering tasks. Such a solution better approximate the relaxed continuous dimension reduction solution by the true discrete Co-clustering solution. We also establish theoretical connections among our method and NMF, k-means and PNMF; that explain the performance improvement. On sparse data sets, our partitioning algorithms give better results in terms of clustering than the state-of-art algorithms devoted to similar tasks for data sets with different sizes, degrees of overlapping and balances.

Chapter 6

Multi-Manifold Co-clustering



6.1 Introduction

The co-clustering problem can be formulated as a matrix approximation problem minimizing the approximation error between the original data X and the reconstructed matrix based on the cluster structures. This approximation can be solved by an iterative alternating least-squares optimization procedure (see, for instance [Govaert and Nadif, 2014]). Several algorithms have been proposed using the principle of double *Kmeans* [Rocci and Vichi, 2008]. Here, we propose a new variant formulation of double *Kmeans* concept, referred in the sequel, Matrix Decomposition based Co-clustering algorithm (MDC).

6.2 Matrix Decomposition based Co-clustering

Given a data set $X \in \mathbb{R}^{d\times n}$ and defined by $X := \{x_{ji}; j = 1,\ldots,d; i = 1,\ldots,n\}$, the co-clustering considers simultaneously the set of samples $\{\mathbf{x}_{.1},\ldots,\mathbf{x}_{.n}\}$ where $\mathbf{x}_{.i} = (x_{1i},\ldots,x_{ni})$ and the set of features $\{\mathbf{x}_{1.},\ldots,\mathbf{x}_{d.}\}$ where $\mathbf{x}_{j.} = (x_{j1},\ldots,x_{jd})$ in order to organize data matrix X into homogeneous blocks. This block structure can be obtained by a couple of partitions $\mathcal{P} = \{\mathcal{P}_1,\ldots,\mathcal{P}_k\}$ of columns into k clusters and $\mathcal{Q} = \{\mathcal{Q}_1,\ldots,\mathcal{Q}_\ell\}$ of rows into ℓ clusters. Then a summary defined by a matrix $S := \{(s_{qp}; q = 1,\ldots,\ell; p = 1,\ldots,k\}$ of size $\ell \times k$ can be computed. Each summary s_{qp} corresponding to block (q,p) is a real number and the row and column vectors of S are noted $\mathbf{s}_{.q}$ and $\mathbf{s}_{p.}$.

The partitions $\mathcal{P}$ and $\mathcal{Q}$ can be respectively expressed as binary matrices $G := \{g_{ip}; i = 1,\ldots,n; p = 1,\ldots,k\}$ with $g_{ip} = 1$ if $i \in \mathcal{P}_p$ and $g_{ip} = 0$ otherwise, and $F := \{f_{jq}; j = 1,\ldots,d; q = 1,\ldots,\ell\}$ with $f_{jq} = 1$ if $j \in \mathcal{Q}_q$ and $f_{jq} = 0$ otherwise.

6.2.1 Co-clustering via *double Kmeans*

The co-clustering problem can be formulated as a matrix approximation problem that consists in minimizing the approximation error between the original data X and the reconstructed matrix based on $\mathcal{P}$, $\mathcal{Q}$ and S. An iterative algorithm attempts to identify optimal partitions $\mathcal{P}$ and $\mathcal{Q}$. The most commonly used criterion to measure the deviation between the data matrix $X = (x_{ji})$ and the structure described by $\mathcal{P}$, $\mathcal{Q}$ and S is defined by

$$\begin{aligned}\Psi(X,\mathcal{P},\mathcal{Q},S) &= \sum_{p=1}^{k}\sum_{q=1}^{\ell}\sum_{i\in\mathcal{P}_p}\sum_{j\in\mathcal{Q}_q}(x_{ji}-s_{qp})^2 \\ &= \sum_{i=1}^{n}\sum_{p=1}^{k}\sum_{j=1}^{d}\sum_{q=1}^{\ell} g_{ip}f_{jq}(x_{ji}-s_{qp})^2.\end{aligned}$$

Several algorithms that consist in using the principle of a *double Kmeans* have been proposed to minimise this criterion (see for instance [Baier et al., 1997; Cho et al., 2004; Rocci and Vichi, 2008]). Here, we propose a new parsimonious matrix formulation of this algorithm. First, it is easy to show that the criterion can be rewritten

$$\Psi(X, \mathcal{P}, \mathcal{Q}, S) = \left\| X - FSG^\top \right\|^2 \tag{6.1}$$

where $G \in \{0,1\}^{n \times k}$ is the sample partition matrix of size $(n \times k)$, $F \in \{0,1\}^{d \times \ell}$ is the feature partition matrix of size $(d \times \ell)$ and S represents the block value matrix of size $(\ell \times k)$. $\|.\|$ denotes the the Frobenius norm.

Next, we aim to optimize this criterion. This task is based on the following proposition.

Proposition 5. *Given $D_f = F^\top F$ and $D_g = G^\top G$, the criterion to be optimized can be expressed as the sum of two terms in two different ways:*

$$1. \left\| X - FSG^\top \right\|^2 = \|X - FZ\|^2 + \left\| Z - SG^\top \right\|^2_{D_f} \tag{6.2}$$

$$\textit{where} \quad Z := \{z_{qi} = \frac{\sum_j f_{jq} x_{ji}}{\#\mathcal{Q}_q}; q = 1, \ldots, \ell; i = 1, \ldots, n\}$$

$$2. \left\| X - FSG^\top \right\|^2 = \left\| X - WG^\top \right\|^2 + \|W - FS\|^2_{D_g} \tag{6.3}$$

$$\textit{where} \quad W := \{w_{jp} = \frac{\sum_i g_{jp} x_{ji}}{\#\mathcal{P}_p}; p = 1, \ldots, k; j = 1, \ldots, d\}$$

Proof.

$$\begin{aligned}
\left\| X - FSG^\top \right\|^2 &= \sum_{p=1}^{k} \sum_{q=1}^{\ell} \sum_{i \in \mathcal{P}_p} \sum_{j \in \mathcal{Q}_q} (x_{ij} - s_{qp})^2 \\
&= \sum_{i,p} \sum_{j,q} g_{ip} f_{jq} (x_{ji} - s_{qp})^2 && (6.4) \\
&= \sum_{i,p} \sum_{j,q} g_{ip} f_{jq} (x_{ji} - z_{qi} + z_{qi} - s_{qp})^2 && (6.5) \\
&= \sum_{i,p} g_{ip} \sum_{j,q} f_{jq} (x_{ji} - z_{qi})^2 + \sum_{i,p,q} g_{ip} (z_{qi} - s_{qp})^2 \sum_j f_{jq} && (6.6) \\
&+ 2 \sum_{i,p,q} (z_{qi} - s_{qp}) \sum_j g_{ip} f_{jq} (x_{ji} - z_{qi}). && (6.7)
\end{aligned}$$

It is easy to show that the first term is reduced to

$$\sum_{i,j,q} f_{jq}(x_{ji} - z_{qi})^2 \quad = \quad \|X - FZ\|^2$$

The second term is reduced to

$$\sum_{i,p,q} \#\mathcal{Q}_q g_{ip}(z_{qi} - s_{qp})^2 \quad = \quad \left\|Z - SG^\top\right\|_{D_f}^2 \quad \text{where} \quad D_f = F^\top F$$

and the third term is null since $\sum_j g_{ip} f_{jq}(x_{ji} - z_{qi}) = 0$.

Then we deduce a new formulation of (6.1)

$$\left\|X - FSG^\top\right\|^2 = \|X - FZ\|^2 + \left\|Z - SG^\top\right\|_{D_f}^2 .$$

The second equation (6.3) can be proved in the same way. □

From equation (6.2) we deduce that if $\mathcal{Q}$ is fixed,

$$\min_{G,F,S} \left\|X - FSG^\top\right\|^2 \iff \min_{G,S} \left\|Z - SG^\top\right\|_{D_f}^2$$

Note that this minimization is performed on a reduced matrix Z of size $\ell \times n$. Similarly, from equation (6.3) we deduce that if $\mathcal{P}$ is fixed,

$$\min_{G,F,S} \left\|X - FSG^\top\right\|^2 \iff \min_{F,S} \|W - FS\|_{D_g}^2$$

Note that this minimization is performed on a reduced matrix W of size $d \times k$.

Finally, it is easy to show that with $\mathcal{P}$ and $\mathcal{Q}$ fixed, the optimal values of s_{qp} are the mean values of block clusters.

Hereafter we give the matrix expression of matrices Z, W and S:

$$Z = (F^\top F)^{-1} F^\top X, \tag{6.8}$$

$$W = XG(G^\top G)^{-1}, \tag{6.9}$$

$$S = (F^\top F)^{-1}\ F^\top XG(G^\top G)^{-1}. \tag{6.10}$$

In summary, the steps of the MDC algorithm can be deduced with a matrix formulation in Algorithm 9. MDC can be viewed as a *double Kmeans* but on intermediate reduced matrices Z and W instead the original data matrix X.

Algorithm 9: MDC algorithm .

1. Start from an initial position $(G^{(0)}, F^{(0)})$;
2. Compute $S^{(0)}$ by using eq.(6.10);
3. $t = 0$;

repeat

(a) - Update $G^{(t+1)}$ by using $(Z)^{(t)}$, eq. (6.8)

$$g_{ip}^{(t+1)} = \begin{cases} 1 & p = \texttt{arg}\min_{p'} ||(\mathbf{z}_{.i})^{(t)} - \mathbf{s}_{p'.}^{(t)}||_{D_g}^2 \\ 0 & \text{otherwise.} \end{cases}$$

(b) - Update $F^{(t+1)}$ by using $(W)^{(t+1)}$ eq.(6.9)

$$f_{jq}^{(t+1)} = \begin{cases} 1 & q = \texttt{arg}\min_{q'} ||(\mathbf{w}_{j.})^{(t+1)} - \mathbf{s}_{q'.}^{(t)})||_{D_f}^2 \\ 0 & \text{otherwise.} \end{cases}$$

(c) - Update $S^{(t+1)}$ by using eq.(6.10).

until *convergence*;

Note that MDC and FNMTF proposed in [Wang et al., 2011b] are two equivalent algorithms optimizing the same objective function (both MDC and FNMTF used double Kmeans technique). However, it is important to emphasize that contrary to FNMTF, MDC does not require the calculation of G and F by relying on the original data but only on reduced intermediate matrices Z, W. We will exploit this important advantage of MDC in the sequel.

6.3 MDC algorithm on manifolds

As shown in Section 6.2.1, co-clustering can be formulated as follow

$$\min_{G,F,S} \left\| X - FSG^\top \right\|^2. \quad G \in \{0,1\}^{n\times k}, \quad F \in \{0,1\}^{d\times \ell}. \tag{6.11}$$

Recent research has shown that existing Matrix Tri-Factorization based co-clustering methods fail to consider the intrinsic geometric structure in the data which is essential to data co-clustering on manifolds [Gu and Zhou, 2009; Wang et al., 2011b].

6.3.1 Locality-preserving

In our approach, we aim to find the best partitions that classify data samples and data features as accurately as possible and at the same time preserves the geometry of the underlying manifolds. In order to preserve the geometrical properties of manifold data, two undirected graphs are

constructed to model the local manifold structures. The first is constructed from samples and is denoted $(\eth_g)$. Its vertices correspond to the samples and its edge weights represent the affinity between the samples. The second, denoted $(\eth_f)$, is constructed from features, in an analogous way. Then, and according to the *smoothness Assumption*, two locality-preserving regularization terms are used to measure the smoothness with respect to the intrinsic manifolds of samples and features and are defined by the two following manifold approximation problems

$$\left\|GG^\top - L_g\right\|^2 \text{ and } \left\|FF^\top - L_f\right\|^2$$

Data samples and data features manifolds are expressed by the normalized graph Laplacians L_g and L_f calculated by

$$L_g = D_g^{-\frac{1}{2}} K_g D_g^{-\frac{1}{2}} \quad \text{and} \quad L_f = D_f^{-\frac{1}{2}} K_f D_f^{-\frac{1}{2}}$$

where D_g, D_f are diagonal matrices whose entries are row sums of the affinity matrices K_g and K_f respectively. Introducing these terms in (6.1), the new optimization problem becomes:

$$\min_{G,F,S} \left\|X - FSG^\top\right\|^2 + \alpha \left\|GG^\top - L_g\right\|^2 + \beta \left\|FF^\top - L_f\right\|^2 \qquad (6.12)$$
$$G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times \ell}$$

where α and β are regularization parameters to balance the reconstruction error of co-clustering in the first term, together with labeling smoothness in the sample space and feature space in the second and third terms.

In Table 6.1, we review the graph-regularized-based methods that we will use in the sequel.

Table 6.1 – Compared co-clustering methods. For each type of initialization, we precise below the published paper where the initialization was used.

Method	Objective Function	Initialization
DRCC	$\left\|X - FSG^\top\right\|^2 + \alpha\text{Tr}(G^\top L_g G) + \beta\text{Tr}(F^\top L_f F)$ $G \in \mathbb{R}^{n\times k},\ F \in \mathbb{R}^{d\times \ell},\ S \in \mathbb{R}^{\ell\times k}$	Kmeans [Gu and Zhou, 2009] Random [Wang et al., 2011b]
FNMTF	$\left\|X - FSG^\top\right\|^2,\ G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times \ell},\ S \in \mathbb{R}^{\ell\times k}$	Random[Wang et al., 2011b]
LPFNMTF	$\left\|X - FSG^\top\right\|^2 + \alpha\text{Tr}(G^\top L_g G) + \beta\text{Tr}(F^\top L_f F)$ $G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times \ell},\ S \in \mathbb{R}^{\ell\times k}$	Random[Wang et al., 2011b]
MDC	$\left\|X - FSG^\top\right\|^2 + \alpha \left\|GG^\top - L_g\right\|^2 + \beta \left\|FF^\top - L_f\right\|^2$ $G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times \ell},\ S \in \mathbb{R}^{\ell\times k}$	$k-$means or $Sk-$means

6.3.2 Reformulation of (6.12) as an Orthogonal Procrustes Problem

Because F and G are constrained to be cluster indicator matrices, it is difficult to solve the objective function of our problem (6.12). It is, therefore, important that (6.12) be reformulated and simplified. Next, we propose to model both matrices G and F.

Given a symmetric positive semi-definite similarity matrix L_g, the following decompositions can be proposed $L_g = B_g B_g^\top$ s.t $B_g^\top B_g = I$ and $L_g = GG^\top$ s.t $G \geq 0$ leading a relation between G and B_g. After obtaining B_g via an eigen-analysis, we can formulate the recovery of the cluster membership matrix G as follows $G = B_g Q_g + E$ where Q_g is an $(k \times k)$ orthonormal rotation matrix which most closely maps B_g to G, and E denotes the residual matrix. Specifically, finding G can be posed as the following optimization

$$\min_{G,Q_g} ||G - B_g Q_g||^2 \quad \text{s.t } Q_g^\top Q_g = I, \quad G \geq 0. \tag{6.13}$$

In a similar way, given B_f the eigendecomposition of L_f, the recovery of matrix F can be posed as the following optimization

$$\min_{F,Q_f} ||F - B_f Q_f||^2 \quad \text{s.t } Q_f^\top Q_f = I, \quad F \geq 0. \tag{6.14}$$

Using equations (6.13) and (6.14) , the expression of our new objective can be written as:

$$\begin{aligned} &\min_{G,F,S} \left\|X - FSG^\top\right\|^2 + \alpha \left\|G - B_g Q_g\right\|^2 + \beta \left\|F - B_f Q_f\right\|^2 \\ &G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times \ell}, Q_g^\top Q_g = I,\ Q_f^\top Q_f = I. \end{aligned} \tag{6.15}$$

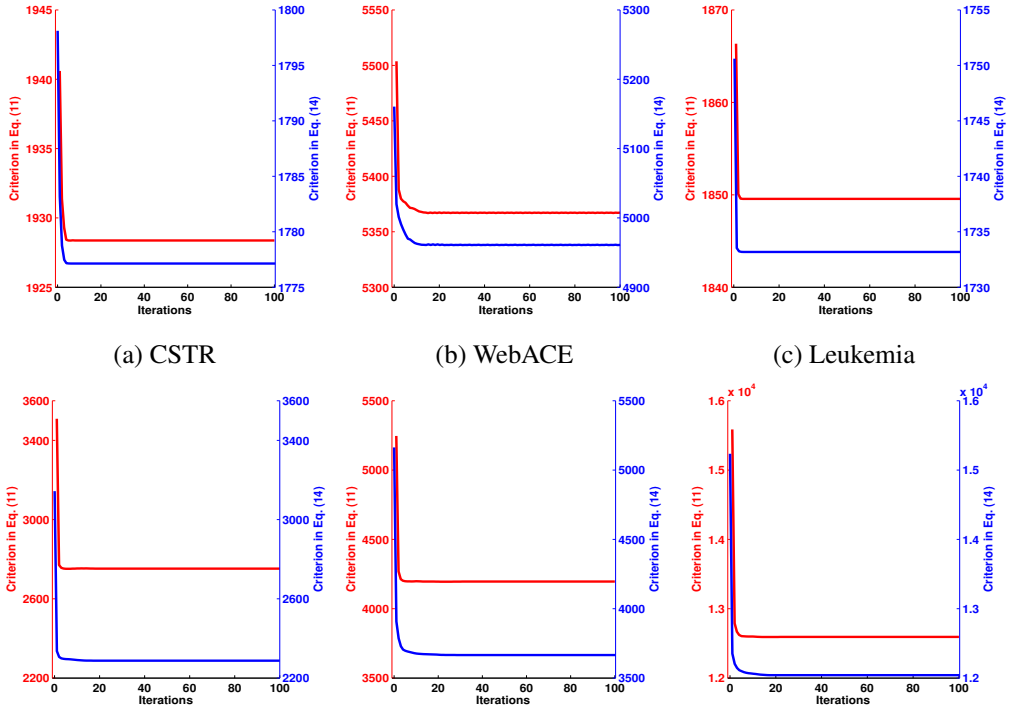
Discussion about the criterion optimized As we have seen below in Eq. 6.12, the MDC algorithm relies on the optimization of

$$\begin{aligned} \min_{G,F,S} \left\|X - FSG^\top\right\|^2 + \alpha \left\|GG^\top - L_g\right\|^2 + \beta \left\|FF^\top - L_f\right\|^2, \\ G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times \ell}. \end{aligned}$$

that we have simplified to Eq. 6.15

$$\begin{aligned} &\min_{G,F,S} \left\|X - FSG^\top\right\|^2 + \alpha \left\|G - B_g Q_g\right\|^2 + \beta \left\|F - B_f Q_f\right\|^2 \\ &G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times \ell}, Q_g^\top Q_g = I,\ Q_f^\top Q_f = I. \end{aligned}$$

Herein, we illustrate the impact of this formulation. In Fig. 6.1, we see that the minimization of (6.12) by our algorithm involves the minimization of (6.15). This reinforces our modeling matrices G and F. Although the results of visualisation and clustering in terms of Acc, NMI, and ARI are very good, further investigation in the approximation of criterion (6.15) could be even more profitable.



(a) CSTR (b) WebACE (c) Leukemia

(d) COIL20 (e) COIL100 (f) USPS

Figure 6.1 – Behaviors of criterion (6.12) and criterion (6.15) during iterations of the MDC

6.4 Single-Manifold Learning

Real-world data in nowadays have high dimensionality. In order to reduce the dimensionality, a manifold learning technique can be used to map a set of high-dimensional data into a low-dimensional space, while preserving the intrinsic structure of the data. In linear methods, the principal component analysis (PCA) is certainly the best known, however, for better representation taking into account clusters of data, the Canonical Discriminant Analysis (CDA) (see for instance, [Gittins, 1985]) is the more appropriate. It is similar to PCA but specialized to the context of discriminant analysis; one primary purpose of CDA is to separate clusters

in a lower dimensional discriminant space. In unsupervised learning, these clusters can be obtained by any clustering algorithm. More efficient in nonlinear cases, a number of techniques have been proposed, including Multi-Dimensional Scaling (MDS), Isometric Feature Mapping (ISOMAP), Locally Linear Embedding (LLE), Locally Preserving Projections (LPP) and Stochastic Neighbor Embedding (SNE). Nevertheless these nonlinear techniques tend to be extremely sensitive to noise, sample size, choice of neighborhood and other parameters (for details see for instance [Engel et al., 2012; Gittins, 1985; van der Maaten et al., 2008]).

These dimensionality reduction methods include different techniques for capturing the nonlinearity of the underlying manifold, and they incorporate local distance information in different ways. Furthermore, the effectiveness of different dimensionality reduction methods varies, and it has been shown that no single method constantly outperforms the others. Rather than choosing a single method, therefore, we seek to apply a set of dimensionality reduction methods and to merge the output of the different methods. Our multi-manifold learning algorithm aims to overcome the drawbacks of single manifold learning methods and to combine the different data structures to which they give rise. In order to illustrate the ability of each of these methods to preserve the initial topology and their capability to separate classes, we used the generated synthetic data set called SwissRoll (1600×3) with 4 classes (400 samples in each class). Figure 6.2 illustrates the obtained manifold projections where the clusters are obtained thanks to *Kmeans*.

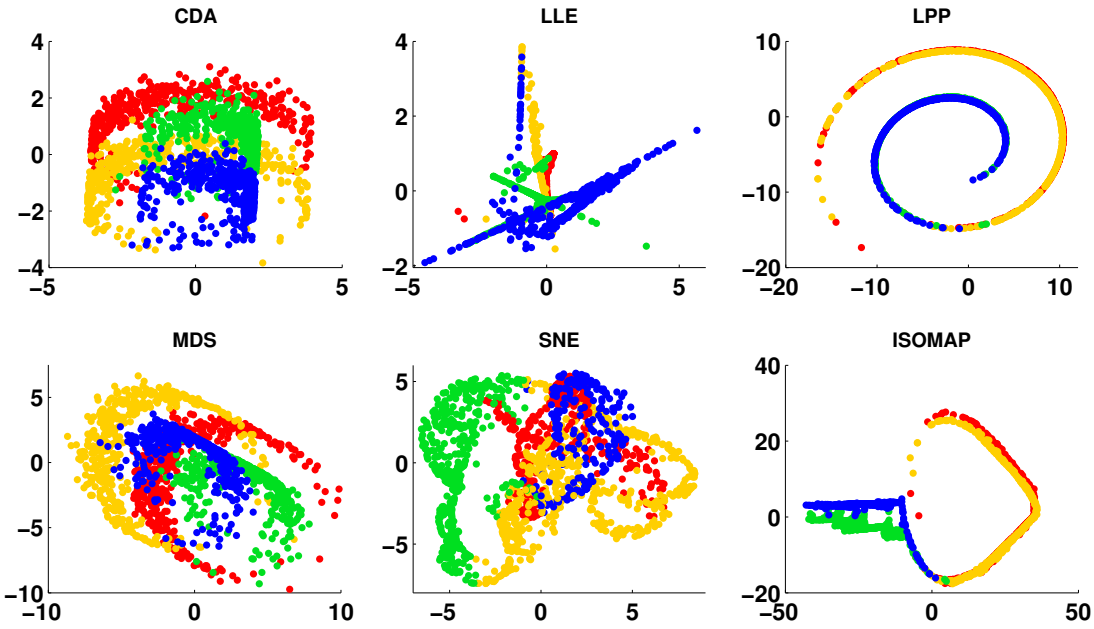


Figure 6.2 – Several low-dimensional manifolds of SwissRoll Data set

Dimensionality reduction methods are restricted to single manifold data sets. However, given the rapid growth in the quantity and complexity of data, multi-manifold learning was proposed to approximate the intrinsic manifold using a subset of candidate manifolds, which can better reflect the local geometrical structure by making use of the graph Laplacian. For example, some linear approaches for multi-manifold learning were proposed in [Fan et al., 2012; Goldberg et al., 2009; Lu et al., 2013; Yang et al., 2011].

6.5 Multi-Manifold Matrix Decomposition

Numerous multi-manifold learning methods have been proposed. k-Manifolds [Souvenir and Pless, 2005] is the first method to classify unorganized data nearly lying on multiple intersecting nonlinear manifolds. Unfortunately, this method is limited to deal with intersecting manifolds since the estimation of geodesic distances will fail when there are widely separated clusters. On the contrary, the Spectral Multi-Manifold Clustering (SMMC), proposed in [Wang et al., 2011c], is able to handle intersections and is well suited to group samples generated from separated manifolds.

In the context of co-clustering that consists in grouping the samples and features simultaneously, in [Li et al., 2012] the authors proposed the Relational Multi-manifold Co-clustering (RMC). With the help of RMC, they showed that the performance of co-clustering can be performed via manifold ensemble learning. However with RMC, the geometric structure modelled by the K-nearest neighbour (KNN) graph learns incomplete and inaccurate intra-type relationships; the KNN graph fails to distinguish the manifolds that are intersecting. To address the above problems, the Robust High-order Co-clustering via Heterogeneous Manifold Ensemble (RHCHME) method has been developed [Jun and Richi, 2015]. RHCHME incorporates multiple subspace learning with a heterogeneous manifold ensemble to learn complete and accurate intra-type relationships. However, the high computational cost of RHCHME, like RMC, makes it unsuitable for large-scale real-world data. Another drawback of both algorithms is that the compromise manifolds are obtained not only from the informative part, but also from the noisy part of the different candidate manifolds. Then to overcome this we propose to rely on dimensionality reduction methods that provide different manifold learning techniques for finding a low-dimensional embedding of the data, while preserving its intrinsic structure.

Motivated by the potential of dimensionality reduction methods, we propose to tackle the aim of co-clustering via an ensemble learning. First, we consider in this work the following well-known dimensionality reduction methods: Canonical Discriminant Analysis (CDA), Multi-Dimensional Scaling (MDS), Isometric Feature Mapping (ISOMAP), Locally Linear Embedding (LLE), Locally Preserving Projections (LPP) and Stochastic Neighbor Embedding

(SNE) (for details see for instance [Engel et al., 2012; Gittins, 1985; van der Maaten et al., 2008]). This choice can be extended to other methods.

Note that in the literature neither of these methods consistently outperforms the others. For this reason, we propose to exploit their strengths according to studied data in an unified framework and we propose a novel Multi-Manifold Co-clustering algorithm referred as M3DC. It attempts to consider simultaneously the diversity of geometric structures in both the sample manifold and the feature manifold, with the aim of discarding the noisy part in each candidate manifold. Instead of choosing a single manifold learning technique, M3DC considers the idea of applying a set of dimensionality reduction methods and extracting the associated manifolds. By considering both sample and feature manifolds, we aim to develop an effective co-clustering algorithm.

We now introduce a Multi-Manifold Matrix-Decomposition-based Co-clustering algorithm (M3DC), which simultaneously considers the geometric structures of both the data manifold and the feature manifold. We present an optimization scheme based on the iterative updating rules of three factor matrices to solve its objective function.

6.5.1 Multi-Manifold Learning

To consider different data manifolds, a set of C candidate graph Laplacians is defined. The intrinsic manifold of the sample or feature space lies in the convex hull of these pre-given candidate manifolds. This assumption can be seen as constraining the search space, since the optimal graph Laplacian is an discrete approximation of the intrinsic manifold.

Sample multi-manifold learning means that the manifold ensemble L_g is represented as a linear combination of the predefined sample candidate manifolds $\left\{L_g^1, \ldots, L_g^C\right\}$. Each candidate L_g^c is linked to a coefficient γ_g^c, which is shown by

$$L_g = \sum_{c=1}^{C} \gamma_g^c \, L_g^c, \quad s.t \sum_{c=1}^{C} \gamma_g^c = 1, \; \gamma_g^c \geq 0. \tag{6.16}$$

Since L_g is in a convex hull of C candidate graph Laplacians, it is itself a graph Laplacian. The coefficients are imposed by the simplex constraints.

Similarly, feature multi-manifold learning means that the manifold ensemble L_f is represented as a linear combination of the predefined feature candidate manifolds $\left\{L_f^1, \ldots, L_f^C\right\}$.

$$L_f = \sum_{c=1}^{C} \gamma_f^c \, L_f^c, \quad s.t \sum_{c=1}^{C} \gamma_f^c = 1, \; \gamma_f^c \geq 0. \tag{6.17}$$

In equations 6.16 and 6.17, if we assume that each candidate graph Laplacian contains an informative part and a noisy part, we may consider that the learned compromise L is made on both the informative and the noisy parts. In order to discard the noisy part in each of the candidate manifolds, we propose taking low-dimensional manifold representations into account using a set of C candidate low-dimensional data representations $\{B_g^1, ..., B_g^C\}$.

Since G is a binary matrix, the following loss function is used as a measure of disagreement between each low-rank manifold representation B_g^c and the factor matrix G with respect to Q_g:

$$\sum_{c=1}^{C} \gamma_g^c \left\| G - B_g^c Q_g \right\|^2, \quad \text{s.t. } \{Q_g^\top Q_g = I\} \tag{6.18}$$

where each candidate distance $\left\| G - B_g^c Q_g \right\|^2$ has a corresponding coefficient γ_g^c.

In the same way, for the feature space, we consider a set of C feature candidate low-dimensional data representations $\{B_f^1, ..., B_f^C\}$. Multiple manifolds are integrated using a similar loss function:

$$\sum_{c=1}^{C} \gamma_f^c \left\| F - B_f^c Q_f \right\|^2 \quad \text{s.t. } \{Q_f^\top Q_f = I\}. \tag{6.19}$$

6.5.2 Candidate manifolds construction

In order to discard the noisy part in each of the candidate manifolds, we use the C selected dimensionality reduction methods and we construct the low-dimensional data representations: $\{B_g^1, ..., B_g^C\}$ for samples and $\{B_f^1, ..., B_f^C\}$ for features.

The low-dimensional data representations $\{B^c\}_{c=1..C}$ are obtained via an eigendecomposition. We distinguish the two cases:

1. For CDA, LLE, LPP, MDS and ISOMAP, we consider B_g^c (resp. B_f^c) as the low-dimensional representation provided by these methods. Note that the sought low-dimensional data set is obtained from solving a trace optimization problem [Kokiopoulou et al., 2011].

2. For SNE, we obtain B_g^c (resp. B_f^c) by performing eignedecomposition on the graph Laplacian L_g^c (resp. L_f^c) which is constructed from the low-dimensional data representation given by SNE.

To preserve the local geometrical structure of the spaces of data samples and data features, we integrate the two multi-manifold regularizer terms defined in equations 6.18 and 6.19. We also introduce the l_2 norm of the variable γ (i.e., $\|\gamma\|^2$) to avoid overfitting on only one manifold.

The M3DC objective function is formulated as:

$$\begin{aligned}\min_{G,F,S} \left\|X - FSG^\top\right\|^2 + \alpha \sum_{c=1}^{C} \gamma_g^c \left\|G - B_g^c Q_g\right\|^2 + \theta_g \left\|\gamma_g\right\|^2 \\ +\beta \sum_{c=1}^{C} \gamma_f^c \left\|F - B_f^c Q_f\right\|^2 + \theta_f \left\|\gamma_f\right\|^2 \\ \text{s.t.}, Q_g^\top Q_g = I, Q_f^\top Q_f = I \end{aligned} \tag{6.20}$$

where the parameters α and β are used to tradeoff the contribution of the multi-manifold regularizer. θ_g and θ_f controls the regularization terms $\left\|\gamma_g\right\|^2$ and $\left\|\gamma_f\right\|^2$, respectively. After some simple algebraic manipulations, the above equation can be rewritten as follows

$$\begin{aligned}\min_{G,F,S} \left\|X - FSG^\top\right\|^2 - 2\,\alpha Tr[G^\top (\sum_{i=c}^{C} \gamma_g^c B_g^c) Q_g] + \theta_g \left\|\gamma_g\right\|^2 \\ -2\,\beta Tr[F^\top (\sum_{c=1}^{C} \gamma_f^c B_f^c) Q_f] + \theta_f \left\|\gamma_f\right\|^2 \\ \text{s.t.}, Q_g^\top Q_g = I, Q_f^\top Q_f = I. \end{aligned} \tag{6.21}$$

6.5.3 Optimization

To solve (6.21), we use an alternated iterative method. The problem is simplified using Theorem 2. Hereafter we present the computation of all matrices and parameters.

Computation of S: Fixing G and F, by setting the derivative of $W(G, F, S)$ with respect to S as 0, we obtain:

$$S = (F^\top F)^{-1}\, F^\top X G (G^\top G)^{-1}. \tag{6.22}$$

Computation of Q_g and Q_f: Fixing G, F and S, we can separate (6.21) into two subproblems:

$$\max_{Q_g^\top Q_g = I} Tr[G^\top (\sum_{c=1}^{C} \gamma_g^c B_g^c) Q_g] \quad \text{and} \quad \max_{Q_f^\top Q_f = I} Tr[F^\top (\sum_{c=1}^{C} \gamma_f^c B_f^c) Q_f].$$

Based on theorem 2, by applying SVD on $G^\top (\sum_{c=1}^{C} \gamma_g^c B_g^c)$, we obtain $Q_g = U_g V_g^\top$. Similarly, applying SVD on $F^\top (\sum_{c=1}^{C} \gamma_f^c B_f^c)$ yields $Q_f = U_f V_f^\top$.

Computation of G: We fix S, F and Q_g, and let be $\widetilde{B}_g = (\sum_{c=1}^{C} \gamma_g^c B_g^c) Q_g$. G can be updated by:

$$g_{ip}^{(t+1)} = \begin{cases} 1 & p = \texttt{arg}\min_{p'} ||(\mathbf{z}_{.i})^{(t)} - s_{p'.}^{(t+\frac{1}{2})}||_{D_g}^2 - 2\alpha(\widetilde{B}_g)_{ip'} \\ 0 & \text{otherwise.} \end{cases} \tag{6.23}$$

Computation of F: We fix S,G and Q_f, and let be $\widetilde{B}_f = (\sum_{c=1}^{C} \gamma_f^c B_f^c) Q_f$. F can be obtained by:

$$f_{jq}^{(t+1)} = \begin{cases} 1 & q = \texttt{arg}\min_{q'} ||(\mathbf{w}_{j.})^{(t+1)} - s_{q'.}^{(t+\frac{1}{2})})||_{D_f}^2 - 2\beta(\widetilde{B}_f)_{jq'} \\ 0 & \text{otherwise.} \end{cases} \tag{6.24}$$

Computation of γ_g and γ_f: Fixing α, β, G and F, the objective function in equation 6.21 reduces to two subproblems:

$$1: \max_{\gamma_g} Tr[G^\top (\sum_{c=1}^{C} \gamma_g^c B_g^c) Q_g] + \theta_g \|\gamma_g\|^2, \text{ s.t., } \sum_{c=1}^{C} \gamma_g^c = 1, \ \gamma_g^c \geq 0.$$

$$2: \max_{\gamma_f} Tr[F^\top (\sum_{c=1}^{C} \gamma_f^c B_f^c) Q_f] + \theta_f \|\gamma_f\|^2, \text{ s.t., } \sum_{c=1}^{C} \gamma_f^c = 1, \ \gamma_f^c \geq 0.$$

To optimize the multi-manifold coefficients γ_g and γ_f, we can use the entropic mirror descent algorithm (EMDA) [Beck and Teboulle, 2003], which is especially well suited for dealing with convex problems. In the interests of simplicity, we present the EMDA process for subproblem 1 only.

If θ_g equals 0, then γ_g will have the trivial solutions 0 and 1. If θ_g approaches infinity, the manifolds L_g^c will be treated equally. Hence, we need to assign a proper value to θ_g to guarantee the effectiveness of multi-manifold learning. EMDA can use a general distance-like function rather than Euclidean squared distance. Since the constraints imposed on γ_g is a unit simplex: $\Delta_g = \left\{\gamma_g \in \mathbb{R}^c, \ \sum_{c=1}^{C} \gamma_g^c = 1, \ \gamma_g \geq 0\right\}$,

EMDA requires the objective function Φ to be a convex Lipschitz continuous function with Lipschitz constant Z_Φ w.r.t. a fixed norm. In our approach, this Lipschitz constant is computed for data samples by $\|\nabla \Phi(\gamma_g)\|_1 \leq 2\theta_g + s_g = Z_\Phi$ where $s_g = \text{Tr}(G^\top (\sum_{c=1}^{C} \gamma_g^c B_g^c) Q_g)$.

The pseudo-code of EMDA is given in Algorithm 10, and the steps of M3DC are shown in Algorithm 11.

Algorithm 10: Entropic Mirror Descent Algorithm .

Input : Lipschitz constant Z_Φ, θ, L, G;
Output : Multi-manifold ensemble coefficient γ;
Initialize : γ_i with identical weights $\frac{1}{C}$;
for $c = 1$ **to** C **do**
  repeat
    (a) - $t_m = \sqrt{\frac{2\,ln\,C}{m\,Z_\Phi^2}}$
    (b) - $\gamma_c^{m+1} \leftarrow \frac{\gamma_c^m \exp\left[-t_m\,\Phi'(\gamma_c^m)\right]}{\sum_{c=1}^{C} \gamma_c^m \exp\left[-t_m\,\Phi'(\gamma_c^m)\right]}$, where $\Phi'(\gamma_c^m) = 2\,\theta\,\gamma_c^m + s_c^m$
  until *convergence*;

Algorithm 11: M3DC algorithm

Input:
- Data matrix X.
- The tradeoff parameters α and β.
- C sample candidate manifolds $\{B_g^1, .., B_g^C\}$.
- C feature candidate manifolds $\{B_f^1, .., B_f^C\}$.
Output: Partition matrices G and F
Initialize: G and F using a clustering algorithm
repeat
  (a) - Update $S^{(t)}$ by (6.22)
  (b) - Compute $Q_g^{(t)}$ and $Q_f^{(t)}$
  (c) - Compute $\gamma_g^{(t)}$ and $\gamma_f^{(t)}$ using the EMDA algorithm.
  (d) - Calculate $(Z)^{(t)}$
  (e) - Update $G^{(t+1)}$ by (6.23)
  (f) - Calculate $(W)^{(t+1)}$
  (g) - Update $F^{(t+1)}$ by (6.24)
until *convergence*;

6.6 Numerical experiments

In this section we investigate the use of our proposed M3DC algorithm for image data clustering. The selected dimensionality reduction methods that we compared and combined are commonly used in image community CDA, LPP, LLE, MDS, ISOMAP and SNE. Note that CDA is a supervised method, which is why its candidate manifold is computed using the partitions obtained by *Kmeans* rather than the correct data set partitions. Table 6.2 summarizes some properties of these techniques.

Table 6.2 – Properties of techniques for dimensionality reduction: "p" is the ratio of nonzero elements in the sparse matrix to the total number of elements, "i" is the number of iterations and "k" is the number of neighbors.

method	Data Linearity	Structure Preservation	Metric (distance)	Computational Complexity
CDA	Linear	Local	Euclidean	$O((n+d^2)d)$
LLE	Nonlinear	Local	Euclidean	$O(pn^2)$
LPP	Linear	Local	Euclidean	$O(kn^2)$
MDS	Nonlinear	Global	Euclidean	$O(n^3)$
ISOMAP	Nonlinear	Global	Geodesic	$O(n^3)$
SNE	Nonlinear	Global	Euclidean	$O(in^2)$

First, we present the performance of M3DC on single manifold. The single candidate manifold is constructed using each of the dimensionality reduction methods in turn. The results obtained can then be compared with those of the graph-regularized-based co-clustering methods DRCC [Gu and Zhou, 2009] and LPFNMTF [Wang et al., 2011b]. Secondly, we evaluate the impact that combining all the manifolds has on the quality of the co-clustering, and we compare the performances of M3DC against the multi-manifold approaches Kmanifolds [Wang et al., 2010], RHCHME [Jun and Richi, 2015], SMMC [Wang et al., 2011c] and RMC [Li et al., 2012].

Evaluation metrics. To measure the clustering performance of the compared algorithms we use the commonly adopted metrics, the accuracy (Acc), the Normalize Mutual Information (NMI) [Strehl and Ghosh, 2002] and the Adjusted Rand Index (ARI) [Hubert and Arabie, 1985]. We focus only on the quality of row clustering.

Parameter settings. For the sake of fairness we adopt an experimental design similar to [Wang et al., 2011b]. We run each method under different parameter settings 50 times, and the average result is computed. We report the best average result for each method.

- We set the number of sample clusters equal to the true number of classes in data sets (k) and we set the number of feature clusters equal to the number of sample clusters.
- For each of the compared approaches: DRCC, LPFNMTF, Kmanifolds, RHCHME, SMMC and RMC, the best parameters are used, as suggested in each of the reference articles (see for details [Gu and Zhou, 2009; Jun and Richi, 2015; Li et al., 2012; Wang et al., 2011b,c]).
- For M3DC, the graph Laplacian is constructed using the distance most suitable for the type of data, i.e. the Euclidean-distance-based k-NN for microarray and image data sets and the Cosine-distance-based k-NN for document-term data sets. The neighborhood size is fixed to 5. Furthermore, the regularization parameter α is searched from the grid (0.01, 0.1, 1, 10, 100, 500, 1000). We set $\beta = \alpha$ for both the sample and feature graphs.

6.6.1 Evaluation of M3DC on real data sets

Data sets Numerical experiments were performed using three types of benchmark data sets from the clustering and co-clustering literature [Ding et al., 2006b; Gu and Zhou, 2009; Wang et al., 2011b], namely image data, document-term data and microarray data. Table 6.3 summarizes the characteristics of these data sets where only the sample classes are known. Note that, even if we are interested in clustering along one dimension of data, when dealing with high-dimensional data, it turns out to be beneficial to employ co-clustering.

Table 6.3 – Data sets description.

Data sets	Type	samples	features	classes	Sparsity (%)
Leukemia	Bio	72	5551	2	0
Lung	Bio	203	2008	5	0
Coil20	Image	1440	1024	20	34.38
Coil100	Image	7200	1024	20	0
ORL	Image	400	1024	40	0
Yale	Image	165	1024	15	30.54
USPS	Image	9298	256	10	0
CSTR	Document-term	1428	1024	4	96.59
WebACE	Document-term	2340	1000	20	91.83
RCV1	Document-term	9625	29992	4	99.75
Ng20	Document-term	19949	43586	20	99.98

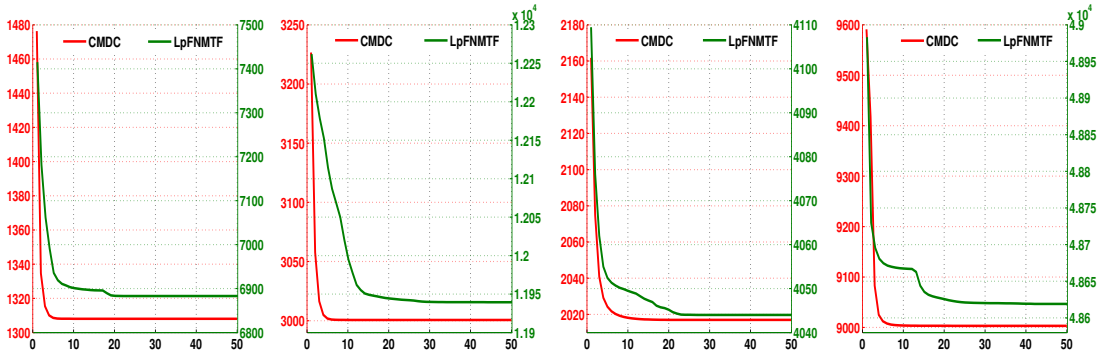
6.6.1.1 Computation time and empirical convergence of MDC

In order to prove the convergence of MDC and compare its computation speeds against the graph-regularized-based co-clustering methods DRCC and LPFNMTF, we repeat co-clustering 50 times, using the different methods, with the optimal parameters for each data set. The average number of iterations (Iter) and computation time (Time) for the different methods on the different data sets are reported in Table 6.4.

Table 6.4 – Average number of iterations and computation time ($\times 10^4$ ms) for convergence. We performed co-clustering on 50 random initialisations.

Data set	Algorithms					
	DRCC		**LPFNMTF**		**MDC**	
	Iter	Time	Iter	Time	Iter	Time
Leukemia	85.2	0.61	5.6	0.21	2.8	0.07
Lung	124.6	20.13	18.5	2.81	15.3	1.65
Coil20	51.6	1.76	15.2	0.74	11.6	0.33
Coil100	36.66	41.83	33.71	36.67	13.89	10.09
ORL	69.4	3.67	48.6	2.87	39.9	2.11
Yale	46.1	1.85	13.8	0.69	7.6	0.29
USPS	18.2	3.63	8.6	0.68	5.3	0.36
CSTR	47.9	0.40	15.6	0.38	6.8	0.08
WebACE	62.3	3.44	16.3	1.33	12.6	0.76
RCV1	84.37	154.48	46.33	78.08	26.01	34.04
NG20	129.96	286.28	139.50	238.06	50.98	57.62

Furthermore, to illustrate the empirical convergence behavior of the proposed MDC, Figure 6.3 shows that MDC, applied on Coil20, USPS, RCV1 and WebACE, requires few iterations to converge. The same observations are verified from other data sets.



(a) Coil20 (b) USPS (c) WebACE (d) RCV1

Figure 6.3 – Empirical convergence behavior of MDC (red line) and LPFNMTF (green line).

The obtained results demonstrate the rapid convergence of our approach, and show that MDC generally requires fewer iterations and less time to converge, not only in comparison to DRCC, but also in comparison to LPFNMTF. These results are consistent with our theoretical analysis.

6.6.1.2 Comparison results

Now, we investigate the effectiveness of M3DC algorithm for document-term, image and microarray data clustering against some other algorithms designed to solve the same tasks in both single manifold and multi-manifold cases.

First, we compare the effectiveness of M3DC on a single manifold. The algorithms we used to construct the candidate manifolds are CDA, LPP, LLE, MDS, ISOMAP (or ISO) and SNE. The results obtained by M3DC can then be compared with those of the co-clustering graph-regularized-based methods DRCC and LPFNMTF.

Secondly, we evaluate the performance of M3DC when these algorithms are combined. The results obtained by M3DC can be compared with those of the multi-manifold approaches Kmanifolds, RHCHME, SMMC and RMC. Note that in these comparison experiments, for M3DC we set the number of feature clusters equal to the number of sample clusters ($\ell = k$).

All these results are reported in Table 6.5. We observe, for all data sets, that RMC outperforms the other compared multi-manifold methods: Kmanifolds, RHCHME and SMMC. We confirm this thanks to some statistical tests.

Table 6.5 – Performances of the compared methods in terms of Acc, NMI and ARI. We used Sk-means ($\circ$) for document-term data sets and k-means ($\star$) for image and microarray data sets.

Data set		Metric	kmeans	Single Manifold								Multi-Manifold				
				M3DC ($C=1$)						DRCC	LP-FNMTF	Kmani-folds	RHC-HME	SMMC	RMC	M3DC ($C=6$)
				CDA	LLE	LPP	MDS	SNE	ISO							
Bio ($\star$)	Leukemia	Acc	0.766	0.767	0.875	0.681	0.736	0.875	0.889	0.806	0.836	0.736	0.833	0.708	0.902	**0.943**
		NMI	0.422	0.432	0.518	0.589	0.606	0.524	0.627	0.462	0.475	0.318	0.419	0.214	0.715	**0.763**
		ARI	0.466	0.485	0.545	0.618	0.650	0.635	0.699	0.566	0.563	0.389	0.428	0.161	0.744	**0.797**
	Lung	Acc	0.708	0.724	0.778	0.591	0.586	0.631	0.690	0.750	0.675	0.660	0.690	0.631	0.808	**0.862**
		NMI	0.586	0.626	0.654	0.436	0.488	0.556	0.576	0.639	0.537	0.521	0.588	0.510	0.584	**0.749**
		ARI	0.433	0.462	0.497	0.282	0.232	0.363	0.421	0.503	0.395	0.377	0.406	0.308	0.553	**0.666**
Image ($\star$)	Coil20	Acc	0.710	0.723	0.859	0.805	0.783	0.817	0.801	0.622	0.723	0.634	0.739	0.697	0.858	**0.889**
		NMI	0.807	0.809	0.904	0.856	0.825	0.862	0.850	0.747	0.796	0.701	0.761	0.721	0.908	**0.920**
		ARI	0.655	0.664	0.767	0.726	0.686	0.752	0.716	0.585	0.643	0.503	0.667	0.594	0.799	**0.806**
	Coil100	Acc	0.548	0.559	0.544	0.761	0.547	0.690	0.658	0.486	0.534	0.619	0.644	0.662	0.728	**0.795**
		NMI	0.783	0.781	0.834	0.913	0.774	0.887	0.843	0.752	0.793	0.798	0.802	0.803	0.839	**0.922**
		ARI	0.495	0.498	0.365	0.725	0.476	0.654	0.589	0.442	0.495	0.555	0.586	0.621	0.698	**0.763**
	ORL	Acc	0.588	0.682	0.783	0.777	0.740	0.795	0.808	0.578	0.618	0.650	0.668	0.658	0.773	**0.822**
		NMI	0.745	0.786	0.841	0.855	0.849	0.858	0.882	0.758	0.766	0.766	0.810	0.779	0.881	**0.901**
		ARI	0.428	0.539	0.740	0.718	0.664	0.725	0.756	0.414	0.426	0.496	0.546	0.513	0.618	**0.788**
	Yale	Acc	0.497	0.646	0.715	0.695	0.827	0.791	0.809	0.406	0.490	0.527	0.470	0.555	0.823	**0.831**
		NMI	0.558	0.715	0.861	0.781	0.915	0.885	0.907	0.463	0.614	0.545	0.519	0.589	0.868	**0.923**
		ARI	0.287	0.577	0.673	0.629	0.773	0.741	0.768	0.195	0.361	0.382	0.344	0.461	0.735	**0.784**
	USPS	Acc	0.699	0.704	0.819	0.705	0.771	0.906	0.897	0.656	0.715	0.693	0.697	0.664	0.764	**0.929**
		NMI	0.652	0.628	0.714	0.632	0.656	0.841	0.808	0.570	0.609	0.606	0.613	0.563	0.721	**0.868**
		ARI	0.558	0.545	0.695	0.558	0.610	0.840	0.813	0.492	0.532	0.546	0.578	0.516	0.658	**0.860**
Document-term ($\circ$)	CSTR	Acc	0.903	0.909	0.929	0.920	0.939	0.922	0.903	0.883	0.908	0.745	0.512	0.806	0.898	**0.946**
		NMI	0.780	0.783	0.915	0.903	0.902	0.814	0.833	0.729	0.784	0.689	0.461	0.714	0.766	**0.906**
		ARI	0.818	0.822	0.852	0.824	0.829	0.795	0.754	0.735	0.763	0.616	0.355	0.739	0.783	**0.830**
	WebACE	Acc	0.653	0.686	0.722	0.776	0.802	0.760	0.735	0.694	0.731	0.602	0.597	0.713	0.718	**0.848**
		NMI	0.698	0.704	0.836	0.818	0.880	0.833	0.790	0.642	0.677	0.640	0.625	0.727	0.766	**0.916**
		ARI	0.586	0.603	0.538	0.616	0.659	0.600	0.578	0.577	0.609	0.399	0.368	0.485	0.530	**0.706**
	RCV1	Acc	0.681	0.715	0.734	0.784	0.759	0.764	0.739	0.726	0.757	0.721	0.556	0.730	0.777	**0.807**
		NMI	0.523	0.512	0.602	0.641	0.615	0.629	0.622	0.468	0.504	0.469	0.287	0.495	0.608	**0.651**
		ARI	0.485	0.565	0.509	0.566	0.523	0.533	0.522	0.455	0.488	0.465	0.230	0.523	0.551	**0.583**
	NG20	Acc	0.388	0.406	0.437	0.517	0.488	0.492	0.456	0.392	0.502	0.414	0.288	0.431	0.504	**0.533**
		NMI	0.365	0.397	0.416	0.491	0.485	0.485	0.450	0.400	0.516	0.392	0.307	0.422	0.493	**0.526**
		ARI	0.148	0.155	0.227	0.286	0.265	0.271	0.243	0.159	0.314	0.162	0.113	0.223	0.266	**0.311**

6.6.1.3 Statistical tests

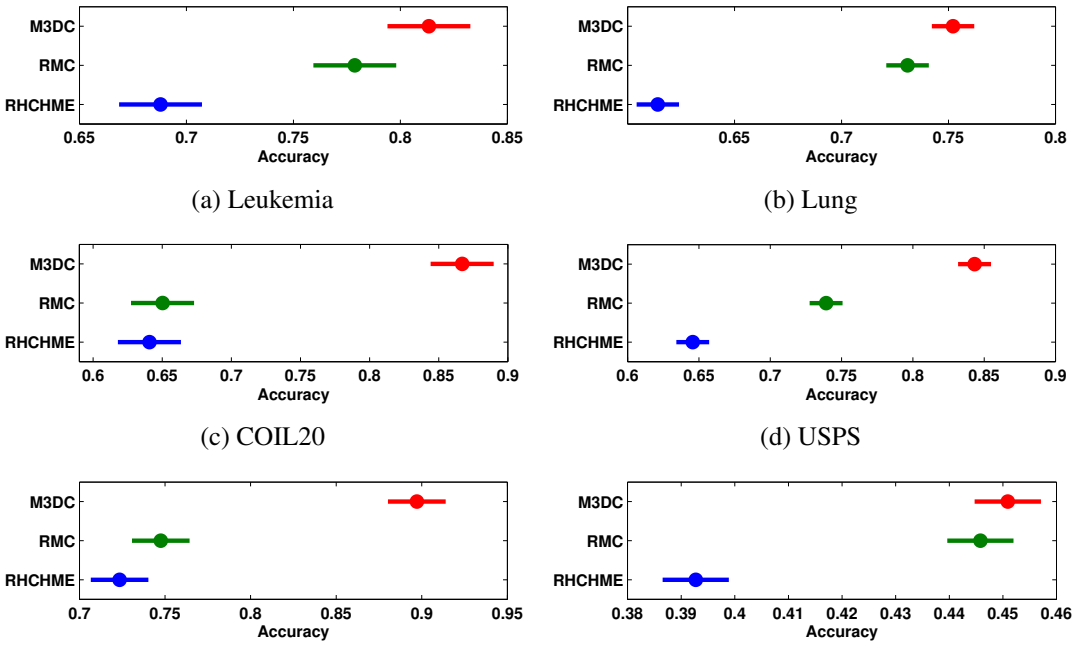
The first question attempted to see if there were any significant differences among the three multi-manifold co-clustering methods including M3DC, RHCHME and RMC? To this end, we first test for the significance of performance differences between M3DC, RHCHME and RMC. We used analysis of variance (ANOVA) and Kruskal-Wallis (KW) tests. The obtained p-values are reported in Table 6.6.

Table 6.6 – Evaluation of co-clustering methods M3DC, RMC and RHCHME using ANOVA test and Kruskal-Wallis (KW) test. Then evaluation of M3DC and RMC using t-tests. These tests are performed on 50 random initializations.

Data set		Metric	RHCHME	RMC	M3DC	P-values		
			mean $\pm std$	mean $\pm std$	mean $\pm std$	ANOVA	KW	t-tests
Microarray ($\star$)	Leukemia	Acc	0.688 $\pm$0.018	0.779 $\pm$0.066	0.813 $\pm$0.115	6.46e-13	1.94e-13	0.0362
		NMI	0.116 $\pm$0.044	0.267 $\pm$0.146	0.378 $\pm$0.240	4.79e-12	3.74e-11	0.0035
		ARI	0.119 $\pm$0.027	0.279 $\pm$0.158	0.419 $\pm$0.272	1.34e-12	6.54e-10	0.0013
	Lung	Acc	0.614 $\pm$0.029	0.731 $\pm$0.041	0.752 $\pm$0.046	5.61e-39	1.16e-21	0.0092
		NMI	0.488 $\pm$0.038	0.455 $\pm$0.077	0.536 $\pm$0.088	5.47e-07	0.58e-03	2.72e-06
		ARI	0.284 $\pm$0.044	0.368 $\pm$0.078	0.503 $\pm$0.054	4.14e-06	1.73e-60	0.0947
Image ($\star$)	Coil20	Acc	0.641 $\pm$0.045	0.690 $\pm$0.094	0.867 $\pm$0.011	2.48e-27	1.87e-21	1.10e-13
		NMI	0.695 $\pm$0.023	0.786 $\pm$0.098	0.899 $\pm$0.020	2.44e-35	9.62e-23	7.97e-11
		ARI	0.546 $\pm$0.041	0.694 $\pm$0.100	0.786 $\pm$0.017	1.63e-23	6.55e-20	6.05e-13
	Coil100	Acc	0.625 $\pm$0.016	0.696 $\pm$0.036	0.774 $\pm$0.033	3.49e-52	9.88e-25	6.69e-19
		NMI	0.794 $\pm$0.005	0.811 $\pm$0.033	0.902 $\pm$0.032	3.15e-45	1.60e-22	9.22e-25
		ARI	0.594 $\pm$0.014	0.651 $\pm$0.044	0.733 $\pm$0.042	3.30e-40	2.34e-23	1.48e-15
	ORL	Acc	0.566 $\pm$0.040	0.722 $\pm$0.036	0.792 $\pm$0.012	2.06e-72	6.31e-29	2.86e-19
		NMI	0.755 $\pm$0.032	0.854 $\pm$0.023	0.886 $\pm$0.008	8.42e-61	1.50e-27	3.70e-13
		ARI	0.420 $\pm$0.045	0.537 $\pm$0.056	0.743 $\pm$0.025	3.15e-74	3.23e-27	3.99e-34
	Yale	Acc	0.515 $\pm$0.027	0.799 $\pm$0.014	0.823 $\pm$0.006	1.74e-13	7.06e-28	4.99e-16
		NMI	0.559 $\pm$0.025	0.852 $\pm$0.013	0.897 $\pm$0.020	6.86e-12	3.21e-28	1.17e-22
		ARI	0.424 $\pm$0.033	0.709 $\pm$0.015	0.752 $\pm$0.022	5.55e-11	2.55e-28	2.49e-19
	USPS	Acc	0.646 $\pm$0.032	0.739 $\pm$0.046	0.843 $\pm$0.057	5.41e-45	4.37e-27	1.57e-16
		NMI	0.557 $\pm$0.024	0.710 $\pm$0.036	0.770 $\pm$0.050	8.24e-60	3.88e-24	5.55e-10
		ARI	0.509 $\pm$0.027	0.644 $\pm$0.040	0.744 $\pm$0.075	3.21e-48	6.83e-24	3.88e-12
Document-term ($\circ$)	CSTR	Acc	0.723 $\pm$0.080	0.747 $\pm$0.084	0.897 $\pm$0.010	3.79e-27	2.74e-20	5.49e-17
		NMI	0.507 $\pm$0.114	0.667 $\pm$0.064	0.766 $\pm$0.016	3.03e-35	2.07e-25	8.43e-15
		ARI	0.476 $\pm$0.126	0.648 $\pm$0.098	0.802 $\pm$0.019	4.16e-36	2.93e-24	4.35e-15
	WebACE	Acc	0.612 $\pm$0.036	0.687 $\pm$0.026	0.812 $\pm$0.031	1.68e-66	2.10e-26	1.11e-38
		NMI	0.699 $\pm$0.016	0.755 $\pm$0.021	0.901 $\pm$0.024	7.03e-92	1.36e-27	1.31e-52
		ARI	0.408 $\pm$0.047	0.516 $\pm$0.011	0.654 $\pm$0.041	1.20e-68	2.06e-28	5.36e-30
	RCV1	Acc	0.569 $\pm$0.054	0.655 $\pm$0.048	0.785 $\pm$0.033	1.25e-49	2.46e-25	6.54e-27
		NMI	0.326 $\pm$0.079	0.511 $\pm$0.044	0.652 $\pm$0.025	1.77e-62	8.80e-28	7.90e-32
		ARI	0.293 $\pm$0.092	0.353 $\pm$0.067	0.523 $\pm$0.034	1.60e-35	1.67e-21	2.34e-25
	NG20	Acc	0.393 $\pm$0.023	0.446 $\pm$0.022	0.461 $\pm$0.019	7.73e-25	3.43e-19	0.0164
		NMI	0.356 $\pm$0.020	0.448 $\pm$0.019	0.475 $\pm$0.023	9.72e-62	7.77e-25	4.93e-09
		ARI	0.160 $\pm$0.024	0.210 $\pm$0.027	0.236 $\pm$0.030	1.38e-28	1.04e-19	7.54e-06

As it can be seen in table 6.6, for each data set the difference among the compared methods is statistically significant; all p-values are less than 1%. Furthermore, we exploit the statistics obtained by ANOVA in applying a post-hoc analysis of M3DC, RMC and RHCHME. The Scheffé's procedure is the most popular of the post-hoc procedures (see for instance [Scheffé, 1959]). The obtained results for studied data sets showed that M3DC almost always outperform significantly RMC and RHCHME, we illustrate this performance in Figure 6.4 and Table 6.7. Furthermore, Scheffé test (with $\alpha = 0.05$) confirm the performance differences between these compared methods. Most of the p-values are less than 5%. The same observations are verified from other data sets.

The M3DC algorithm which exploits only the informative part of the data and removes the noisy part, is clearly more efficient than RMC that combines eleven diverse manifolds generated, directly from the original data matrix, using three kinds of weighting schemes to construct the graph.



(a) Leukemia (b) Lung

(c) COIL20 (d) USPS

(e) CSTR (f) NG20

Figure 6.4 – Post-hoc analysis of M3DC, RMC and RHCHME Accuracy's using Scheffé test. These tests are performed on 50 random initialisations.

Table 6.7 – Post-hoc analysis of M3DC, RMC and RHCHME Accuracy's using Scheffé test (with $\alpha = 0.05$).

Data set		Methods	F-Value	P-value
Microarray	Leukemia	RMC M3DC	2.418	0.092
		RHCHME M3DC	32.143	2.88e-12
		RHCHME RMC	16.928	2.50e-07
	Lung	RMC M3DC	3.520	3.22e-02
		RHCHME M3DC	148.881	8.88e-36
		RHCHME RMC	106.617	3.89e-29
Image	Coil20	RMC M3DC	69.065	9.33e-22
		RHCHME M3DC	72.229	4.29e-23
		RHCHME RMC	0.132	8.77e-01
	USPS	RMC M3DC	62.249	3.30e-20
		RHCHME M3DC	224.372	5.70e-45
		RHCHME RMC	50.257	2.78e-7
Document-term	CSTR	RMC M3DC	60.422	8.85e-20
		RHCHME M3DC	81.424	2.21e-24
		RHCHME RMC	1.563	2.13e-01
	NG20	RMC M3DC	5.118	0.060
		RHCHME M3DC	68.150	1.49e-21
		RHCHME RMC	56.849	6.33e-19

6.6.1.4 Clustering evaluation using Silhouette Score

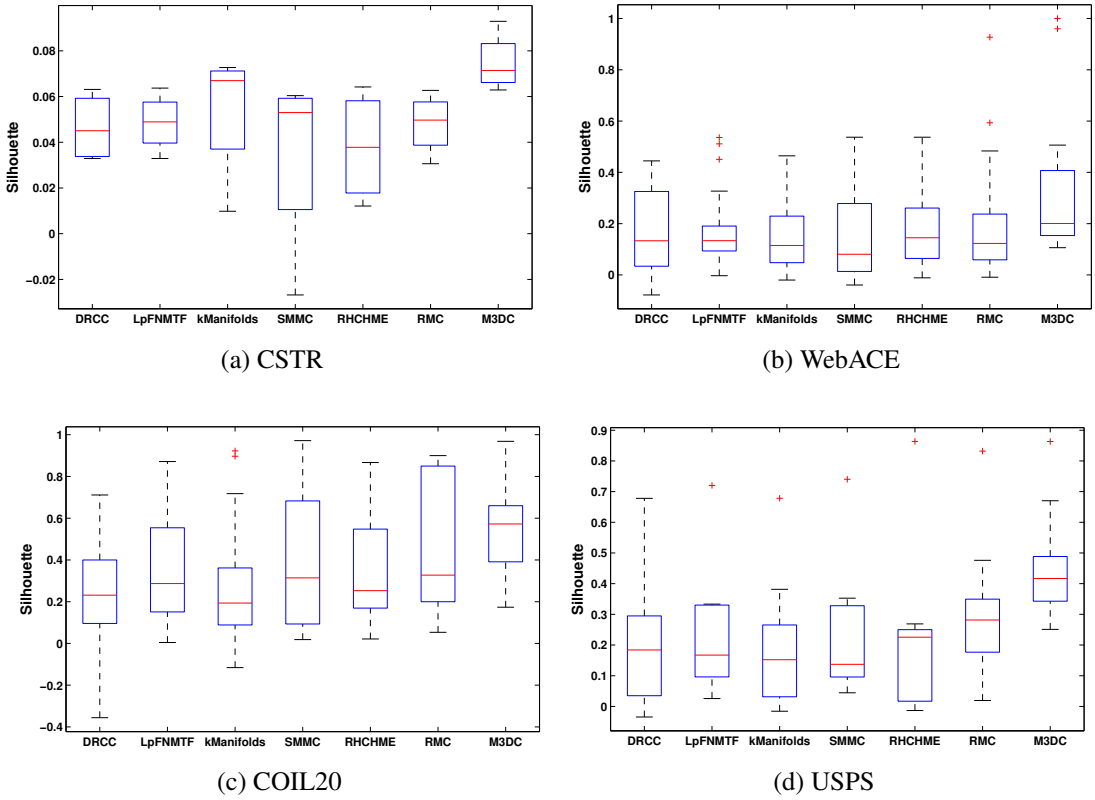
Silhouette index (noted SIL) is a very well-known clustering evaluation approach that introduces clustering quality scores for each individual point and calculates the final quality index as an average of the point-wise quality estimates [Rousseeuw, 1987]. Each point-wise estimate for a point $\mathbf{x}_{.p} \in \mathcal{P}_i$ is derived from two quantities: $a_{i,p}$ and $b_{i,p}$ which correspond to the average distance to other points within the same cluster and the minimal average distance to points from a different cluster, respectively. Formally,

$$a_{i,p} = \frac{1}{|\mathcal{P}_i| - 1} \sum_{\mathbf{x}_{.q} \in \mathcal{P}_i, q \neq p} \|\mathbf{x}_{.q} - \mathbf{x}_{.p}\| \quad \text{and} \quad b_{i,p} = \min_{j=1..k, j \neq i} \frac{1}{|\mathcal{P}_j|} \sum_{\mathbf{x}_{.q} \in \mathcal{P}_j} \|\mathbf{x}_{.q} - \mathbf{x}_{.p}\|$$

$$\text{For each data point } \mathbf{x}_{.p} : \quad SIL(\mathbf{x}_{.p}) = \frac{a_{i,p} - b_{i,p}}{\max(a_{i,p} - b_{i,p})}$$

$$\text{The Silhouette Score : } \quad SIL = \frac{1}{n} \sum_{p=1}^{n} SIL(\mathbf{x}_{.p})$$

In Figure 6.5 are reported the boxplots for four representative data sets to illustrate the good behavior, in terms of Silhouette Score (SIL), of M3DC comparing to different co-clustering methods. The same observations are verified from the remaining data sets.



(a) CSTR (b) WebACE

(c) COIL20 (d) USPS

Figure 6.5 – Performances of the compared co-clustering methods in terms of Silhouette score.

6.6.1.5 Impact of the multi-manifold coefficients γ's

In M3DC, the candidate manifolds are weighted according to how well they reflect their ability to the local geometrical structure of data. The multi-manifold coefficients γ_g^c computed by EMDA are additional indicators of the effectiveness of each method. For all data sets, the multi-manifold coefficients γ_g^c computed by EMDA and reported in Table 6.8.

One might ask what the most efficient projection methods in M3DC are. Through our experiments we noticed that LLE, ISOMAP and SNE contribute greatly for all the tested data sets. In order to visually illustrate the impact of γ's, in Fig.6.6, we report the single manifolds with the different methods of projection and multi-manifold combining them.

Note that the candidate manifolds are weighted according to them quality in reflecting the local geometrical structure of data. We observe that LLE, ISOMAP, SNE presenting the highest γ values is reflected in the Multi Manifolds while CDA with $\gamma_g^{CDA} = 0$ does not contribute in the construction of the intrinsic manifold.

Table 6.8 – Values of γ_g^c (%).

Data sets	Methods					
	CDA	LLE	LPP	MDS	SNE	ISO
Leukemia	8.79	24.61	0	10.40	26.20	30.00
Lung	14.47	43.57	0.67	0.04	12.13	29.12
Coil20	0	36.79	14.01	5.29	24.36	19.55
Coil100	8.24	4.01	47.15	2.83	21.67	16.10
ORL	0	20.34	14.60	9.13	22.76	33.17
Yale	2.18	23.86	13.31	24.26	14.96	21.42
USPS	0	18.04	5.46	8.80	41.37	27.33
CSTR	8.79	21.29	18.29	26.42	15.50	9.71
WebACE	0	8.44	23.47	38.59	19.83	9.67
RCV1	9.26	10.91	29.93	17.92	18.77	13.21
NG20	5.52	8.56	39.11	14.46	20.81	11.25

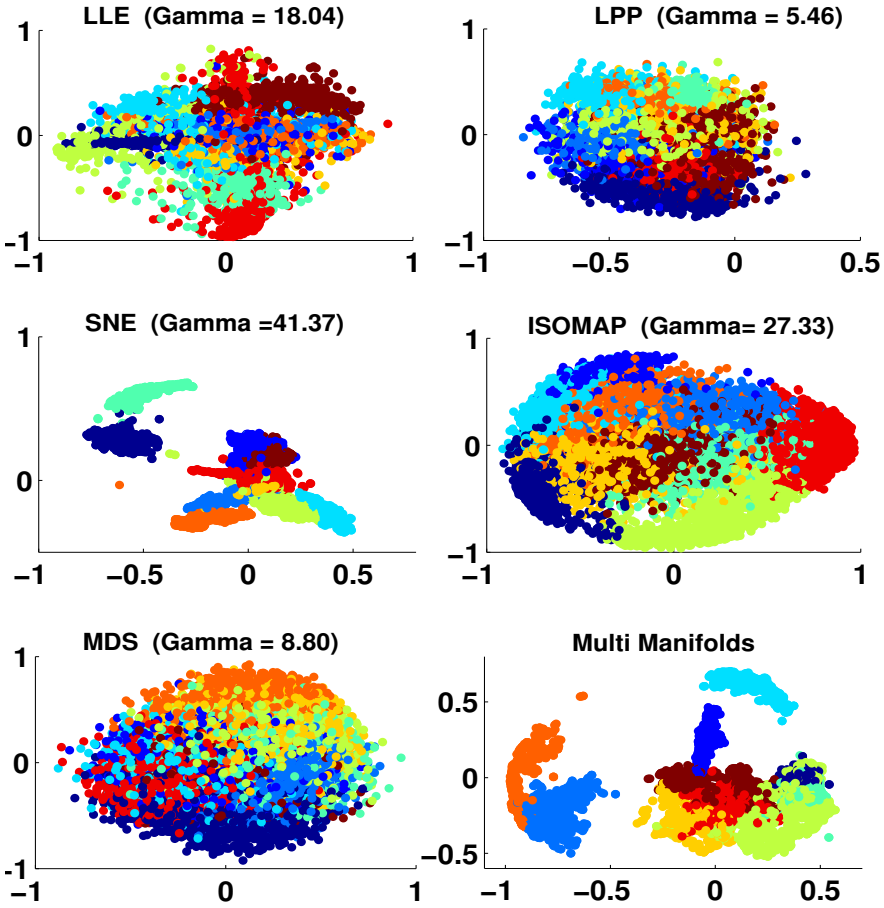


Figure 6.6 – USPS: Single and multi manifolds with the different methods, CDA is absent because $\gamma_g^{CDA} = 0$.

6.6.1.6 Assessing the number of feature clusters

For all the compared methods, i.e, DRCC, LPFNMTF, Kmanifolds, RHCHME, SMMC and RMC, it was suggested that the number of feature clusters is equal to the number of sample clusters (the true number of classes in data sets k). This is why, we set $\ell = k$ so as to ensure that the experiments of these methods against M3DC are compared in a fair fashion. However, in our approach, the candidate manifolds are generated by using some reduction dimension methods. Taking a small values of ℓ to determine the number of components (dimensions), may cause a loss of the information provided by the initial features. Contrariwise, a too large number increases the computational complexity. Then to assess the number of feature clusters, we varied ℓ between 2 and 100, and retained the values that optimizes the criterion as shown in Fig.6.7.

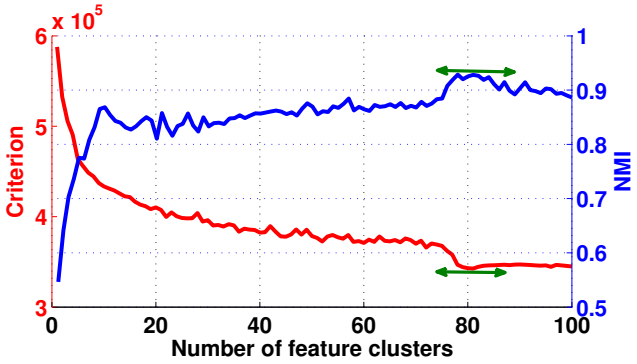
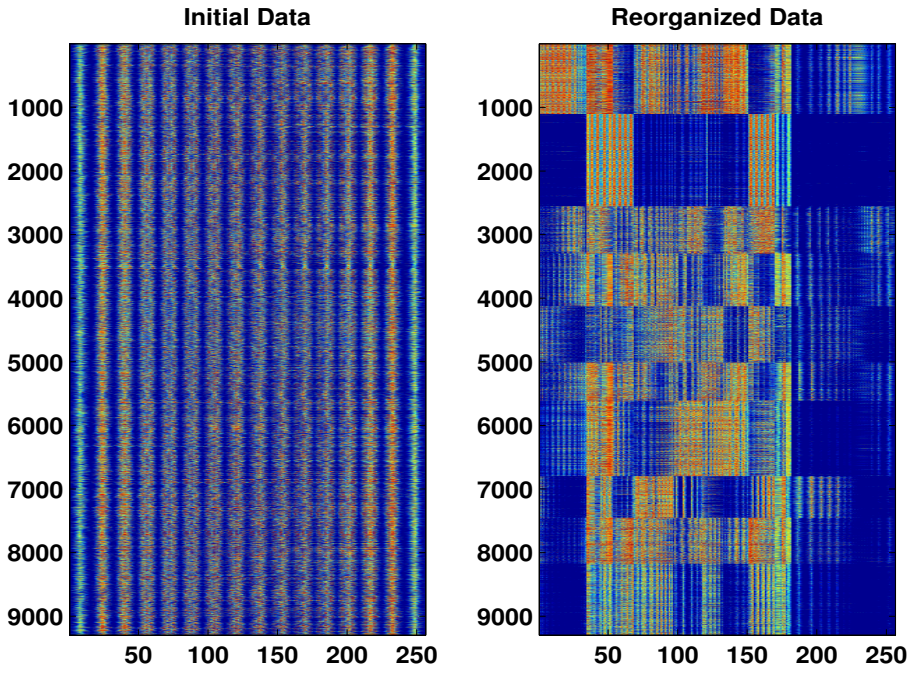


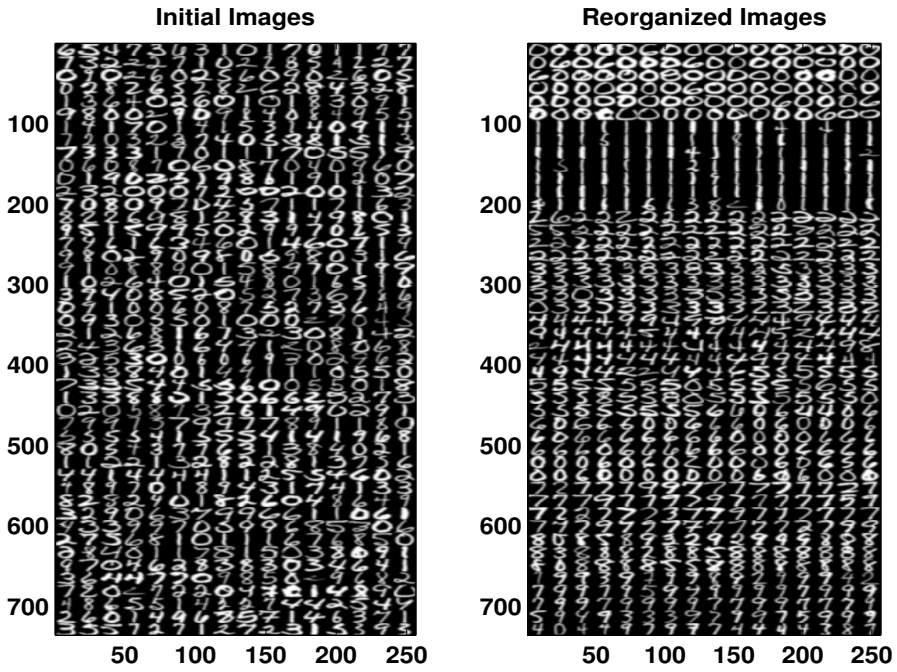
Figure 6.7 – Performances of M3DC on USPS data set. Criterion and NMI according to the number of feature clusters (ℓ).

We observe that the optimum coincides with the NMI optimum as illustrated in Fig.6.7. For all tested data sets, we have applied this successfully strategy to assess ℓ. However, we note that when we exceed this value the optimized criterion starts to increase while the NMI starts to decrease, thus showing that the quality of the co-clustering degrades and we start to observe the disadvantages of clustering (when $\ell = n$). In other words, the benefit of co-clustering on clustering starts to decrease.

Hereafter we evaluate the impact of ℓ. So far we have evaluated M3DC in terms of clustering but as we focus on image data, is M3DC really efficient in the restitution of images? In Fig.6.8 we report the results obtained using USPS data set with the appropriate number of feature clusters; $\ell = 82$ leads to Acc = 0.968, NMI = 0.921 and ARI = 0.939 instead of 0.929, 0.868 and 0.860 respectively for $\ell = 10$. The reorganisation of the original image data after co-clustering requires only the row clusters. This leads to reveal good results of M3DC as depicted in Fig.6.8.



(a) Data visualisation according row and column clusters



(b) Image visualisation according row clusters

Figure 6.8 – Performances of M3DC on USPS image data set.

6.6.2 Evaluation of M3DC on synthetic data sets

In order to evaluate our approach in term of co-clustering, we propose to evaluate the different algorithms on simulated data sets generated according a probabilistic model (see Appendix A).

Table 6.9 – Parameters of simulated data sets and error rates for samples, features and global.

Data	Dimension	Classes	Error Rate (%)			Proportions of sample clusters	Proportions of feature clusters
			$e(G, G')$	$e(F, F')$	$\delta = \delta(Y, Y')$		
Data1	**500x500**	**4x3**	8.4	2.6	10	$\boldsymbol{\pi} = [0.2, 0.3, 0.3, 0.2]$	$\boldsymbol{\rho} = [0.3, 0.4, 0.3]$
Data2			13.2	7.4	20		
Data3			25.4	6.2	30		
Data4			35.0	7.0	40		
Data5			43.0	13.0	50		
Data6			38.0	42.8	65		

To evaluate the three algorithms taking into account the degree of overlapping, the rate of sparsity and the proportions, we perform extensive experiments and we present error rates or accuracy arising from different simulated tables whose parameters are reported in table 6.9. The main points are the following.

- In Figure 6.9 are reported the performances of all algorithms according degrees of overlapping (10%, 20%, 30%,40%, 50% and 65%). M3DC is always better than LPFNMTF and DRCC whatever the degree of overlapping.

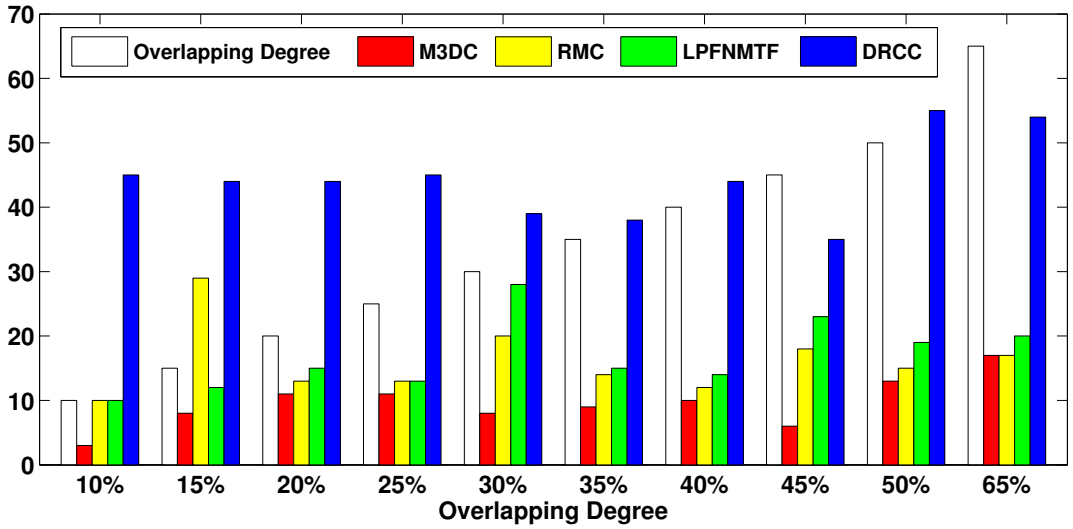
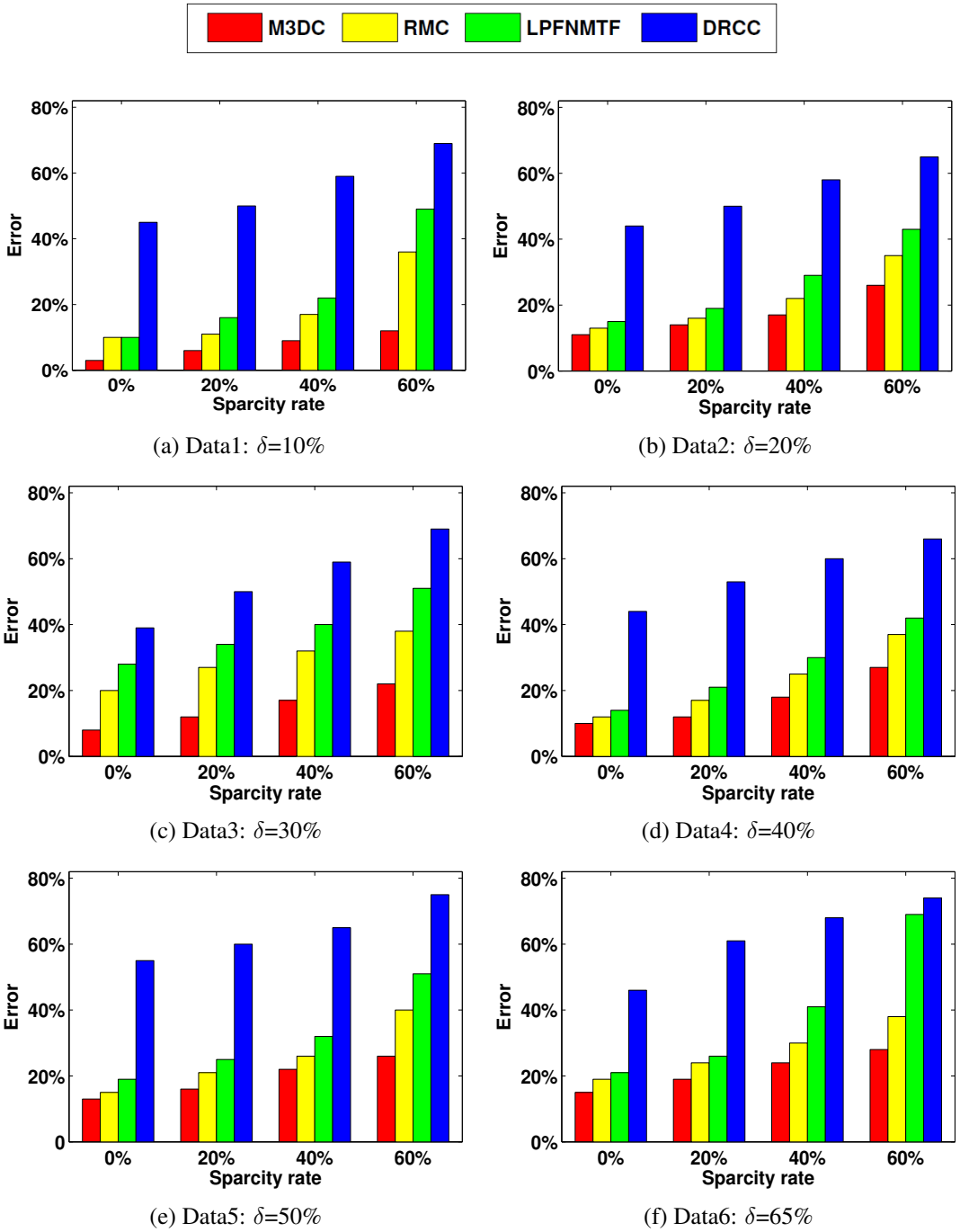


Figure 6.9 – Impact of overlapping

- From initial data sets with a degree of overlapping, we measure the impact of sparsity. In Figure 6.10 are reported the performances of all algorithms according degrees of overlapping

(10%, 20%, 30%, 40%, 50% or 65%) and rates of sparsity (0%, 20%, 40% or 60%). We observe the good behavior of M3DC in all situations.



(a) Data1: δ=10%

(b) Data2: δ=20%

(c) Data3: δ=30%

(d) Data4: δ=40%

(e) Data5: δ=50%

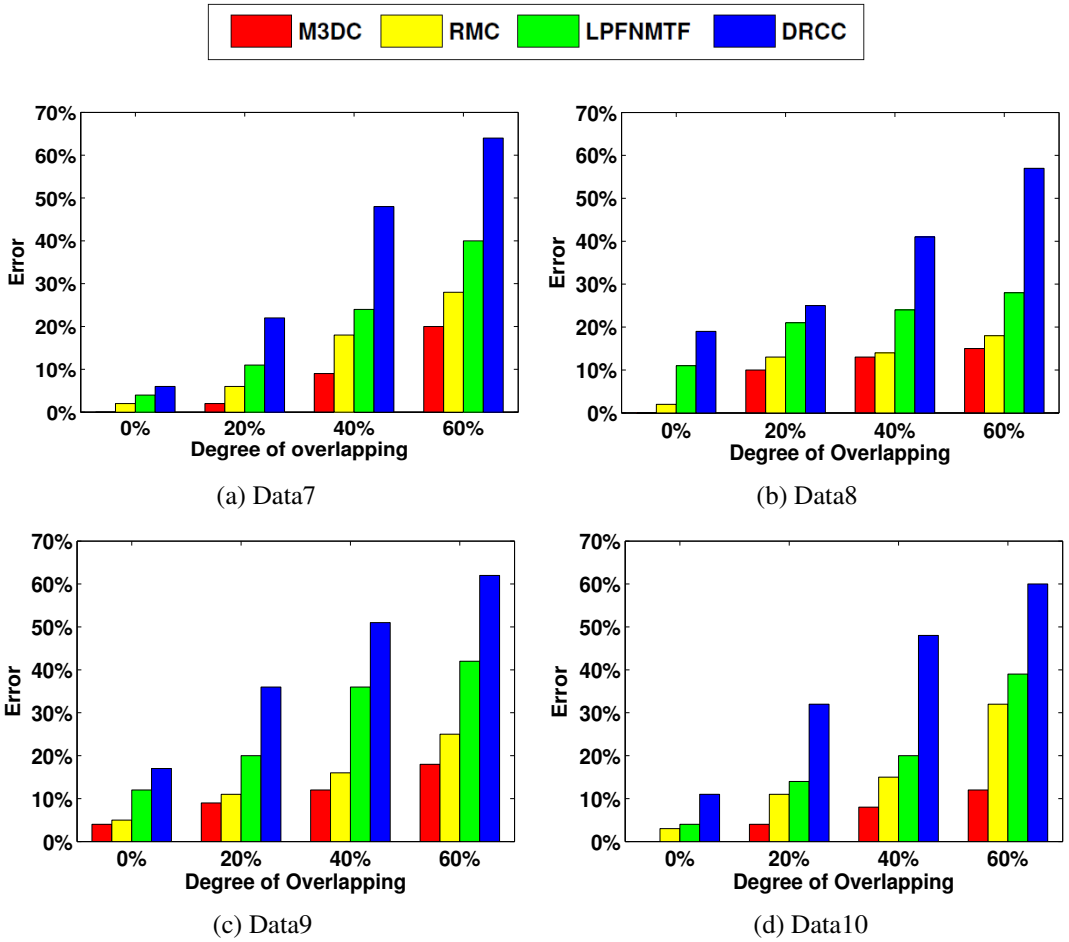
(f) Data6: δ=65%

Figure 6.10 – Impact of sparsity

- In order to evaluate the three algorithms in term of cluster proportions, we perform supplementary experiments by varying the proportions and the degree of overlapping. To this end, in Figure 6.11, we present the results obtained with Data7, Data8, Data9 and Data10 described in Appendix A.

  - Data7: $\boldsymbol{\pi} = [0.1, 0.4, 0.4, 0.1]$, $\boldsymbol{\rho} = [0.1, 0.8, 0.1]$;
  - Data8: $\boldsymbol{\pi} = [0.1, 0.1, 0.1, 0.7]$, $\boldsymbol{\rho} = [0.1, 0.8, 0.1]$;
  - Data9: $\boldsymbol{\pi} = [0.2, 0.3, 0.3, 0.2]$, $\boldsymbol{\rho} = [0.1, 0.8, 0.1]$;
  - Data10: $\boldsymbol{\pi} = [0.1, 0.1, 0.1, 0.7]$, $\boldsymbol{\rho} = [0.3, 0.4, 0.3]$.

  It appears clearly that M3DC is more robust than LPFNMTF and DRCC; even when the proportions are dramatically different it remains the most effective.



(a) Data7 (b) Data8

(c) Data9 (d) Data10

Figure 6.11 – Impact of cluster proportions

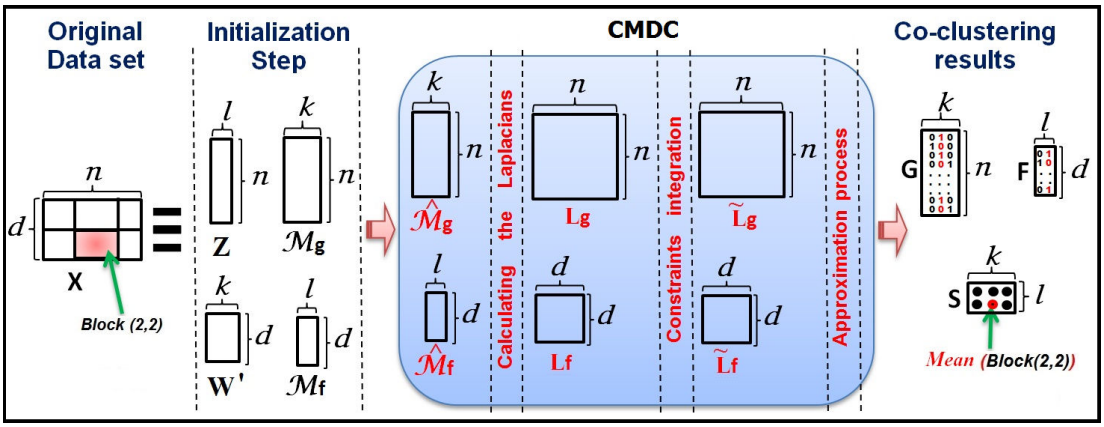
6.7 Conclusion

We proposed a novel algorithm which simultaneously considers the geometric structures of both the sample manifold and the feature manifold. Specifically, we employed multi-manifold learning to approximate the intrinsic manifold using a subset of candidates, which better reflects the local geometrical structure with a graph Laplacian. In order to use the respective strengths of different dimensionality reduction techniques, we selected six manifold-based dimensionality reduction methods that were designed for a variety of purposes and use different metrics for data similarity. Our candidate manifolds are obtained using these methods. The regularization terms are then incorporated into a matrix decomposition framework resulting in a unified objective function. In this way, matrix decomposition becomes an optimization problem. Note that we also investigated the crucial number of feature clusters and offered a simple manner to assess this number.

The numerical experiments show that M3DC gives very interesting results in terms of clustering and restitution of initial images. Furthermore it outperforms multi-manifold methods devoted to the same tasks. In our experiments, we used the popular reduction methods to generate multi-manifolds. It would be interesting to investigate other reduction methods.

Chapter 7

Semi-supervised Co-clustering



7.1 Introduction

Since the advent of Block Clustering [Hartigan, 1972], a number of co-clustering methods have been proposed in a variety of application domains, including image processing, text mining and bio-informatics. However, in many real world applications, the data set to be analyzed presents obstacles such as large dimensions, sparsity and heterogeneity. For this reason, in recent years there has been a surge of interest of constrained co-clustering methods that can cluster samples and feature sets at the same time, guided by certain supervisory information. Usually, this background knowledge can be represented as a set of pairwise constraints that can be generated from a subset of labeled data. Each constraint indicates that two samples (or two features) must belong to the same class (*Must-link*) or that they never be assigned together (*Cannot-link*).Furthermore, efforts have been made in recent years to extend existing co-clustering methods to constrained co-clustering [Chen et al., 2010; Pensa and Boulicaut, 2008; Song et al., 2010; Wang et al., 2008b]. Most of these methods encode *Must-link* (*ML*) and *Cannot-link* (*CL*) constraints by modifying the graph Laplacian, constraining the underlying Eigenspace, or by encoding them as part of a constrained optimization problem. Note that a measure of informativeness can help to select better constraint sets that improve the results of co-clustering. This chapter looks at low-rank factorization-based co-clustering for semi-supervised learning.

7.2 Constrained MDC algorithm (CMDC)

As shown in Section 6.3.1, MDC can be formulated as the following optimization problem:

$$\min_{G,F,S} \left\|X - FSG^\top\right\|^2 + \alpha \left\|GG^\top - L_g\right\|^2 + \beta \left\|FF^\top - L_f\right\|^2 \qquad (7.1)$$
$$G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times \ell}.$$

where α and β are regularization parameters to balance the reconstruction error of co-clustering in the first term, together with labeling smoothness in the sample space and feature space in the second and third terms.

7.2.1 Utility of constraints

Pairwise constraints are most often used in order to guide the learning process and to improve the performance of the model. However, Davidson et al. [Davidson et al., 2006] employed **informativeness** to measure the utility of constraint sets and to select the most beneficial ones.

The informativeness is the amount of conflict between the constraints and the underlying objective function of an algorithm. It is based on measuring the number of constraints that the clustering algorithm cannot predict using its default bias [Davidson et al., 2006]. Given a set of constraints Ω, we generate the partition $\mathcal{P}_A$ by running an algorithm A on the data set without any constraints. We set $Unsat(\varsigma, \mathcal{P})$ to 1 if $\mathcal{P}$ does not satisfy ς, and 0 otherwise. The subset of informative constraints is therefore given by:

$$\text{Inf}_{\mathcal{P}_A}(\Omega) = \{\varsigma \in \Omega \,/\, Unsat(\varsigma, \mathcal{P}_A) = 1\}.$$

7.2.2 Integration of constraints

Many clustering and co-clustering methods have been proposed to encode *Must-link* (ML) and *Cannot-link* (CL) constraints by modifying the graph Laplacian directly [Kulis et al., 2005; Wang et al., 2009]. In our approach we apply a clustering algorithm (*k-means* or *Sk-means*) on both samples and features. $\mathcal{P}^0$ and $\mathcal{Q}^0$ are the obtained partitions. We then extract all the *ML* and *CL* constraints from labeled data. We denote the set of sample constraints $\Omega_g = ML_g \cup CL_g$ and the set of feature constraints $\Omega_f = ML_f \cup CL_f$. Finally, we select the informative constraints according to $\mathcal{P}^0$ and $\mathcal{Q}^0$. We denote the obtained informative constraint sets $\text{Inf}_{\mathcal{P}^0}(\Omega_g)$ and $\text{Inf}_{\mathcal{Q}^0}(\Omega_f)$.

Let x_r be a sample from $\mathcal{P}^0_p$ and $x_{r'}$ a sample from $\mathcal{P}^0_{p'}$:

- If there is an *ML* constraint between x_r and $x_{r'}$, the corresponding coefficient of x_r and $x_{r'}$ in $\hat{L}_g$ is replaced by the largest coefficient of the samples belonging to the same clusters as x_r or $x_{r'}$. (i.e. : $\mathcal{P}^0_p$ or $\mathcal{P}^0_{p'}$).

- If there is a *CL* constraint between x_r and $x_{r'}$, the corresponding coefficient of x_r and $x_{r'}$ in $\hat{L}_g$ is replaced by the smallest coefficient of the samples belonging to the same clusters as x_r or $x_{r'}$.

$$\tilde{L}_g(rr') = \begin{cases} \max_{x_i \in \mathcal{P}^0_r, x_{i'} \in \mathcal{P}^0_{r'}} (\hat{L}_g(ii')), & \text{if } (x_r, x_{r'}) \in \text{Inf}_{\mathcal{P}^0}(ML_g) \\ \min_{x_i \in \mathcal{P}^0_r, x_{i'} \in \mathcal{P}^0_{r'}} (\hat{L}_g(ii')), & \text{if } (x_r, x_{r'}) \in \text{Inf}_{\mathcal{P}^0}(CL_g) \end{cases}$$

Similarly, let y_s be a feature from $\mathcal{Q}^0_q$ and $y_{s'}$ a feature from $\mathcal{Q}^0_{q'}$. We apply the same transformation on $(\hat{L}_f)$ as follows:

$$\tilde{L}_f(ss') = \begin{cases} \max_{y_j \in \mathcal{Q}^0_s, y_{j'} \in \mathcal{Q}^0_{s'}} (\hat{L}_f(jj')), & \text{if } (y_s, y_{s'}) \in \text{Inf}_{\mathcal{Q}^0}(ML_f) \\ \min_{y_j \in \mathcal{Q}^0_s, y_{j'} \in \mathcal{Q}^0_{s'}} (\hat{L}_f(jj')), & \text{if } (y_s, y_{s'}) \in \text{Inf}_{\mathcal{Q}^0}(CL_f) \end{cases}$$

If we consider the modified graph Laplacians $\tilde{L}_g$ and $\tilde{L}_f$, the problem (6.12) becomes:

$$\min_{G,F,S} \left\|X - FSG^\top\right\|^2 + \alpha \left\|GG^\top - \tilde{L}_g\right\|^2 + \beta \left\|FF^\top - \tilde{L}_f\right\|^2 \quad (7.2)$$
$$G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times \ell}.$$

7.2.3 Optimization

As shown in Section 6.3.2, Eq. (7.2) can be simplified to

$$\min_{G,F,S} \left\|X - FSG^\top\right\|^2 + \alpha \left\|G - B_g Q_g\right\|^2 + \beta \left\|F - B_f Q_f\right\|^2 \quad (7.3)$$
$$G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times \ell}, Q_g^\top Q_g = I,\ Q_f^\top Q_f = I.$$

After some simple algebraic manipulations, the above equation can be rewritten as follows

$$\min_{G,F,S} \left\|X - FSG^\top\right\|^2 - 2\alpha Tr(G^\top B_g Q_g^\top) - 2\beta Tr(F^\top B_f Q_f^\top) \quad (7.4)$$
$$G \in \{0,1\}^{n\times k},\ F \in \{0,1\}^{d\times l},\ Q_g^\top Q_g = I,\ Q_f^\top Q_f = I.$$

The optimization problem (7.4) can be solved by using alternated iterative method.

- Computation of S: Fixing G and F, by setting the derivative of $\Psi(G,F,S)$ with respect to S as 0, we obtain:

$$S = (F^\top F)^{-1} F^\top X G (G^\top G)^{-1}. \quad (7.5)$$

- Computation of Q_g and Q_f: Fixing G, F and S, the computation of Q_g and Q_f can be performed by relying on the following theorem

Theorem 2. *Let $G_{n\times k}$ and $B_{n\times k}$ be two matrices. Consider the constrained minimization problem*

$$Q^* = \arg\min_Q \left\|G - BQ^\top\right\|^2 \text{ subject to. } Q^\top Q = I. \quad (7.6)$$

Let $U\Lambda V^\top$ be the (SVD) for $G^\top B$, then $Q^* = UV^\top$.

Proof. Expanding the matrix norm

$$\left\|G - BQ^\top\right\|^2 = Tr(G^\top G) - 2Tr(G^\top BQ^\top) + Tr(QB^\top BQ^\top). \quad (7.7)$$

Since $Tr(G^\top G) = n$ and $Q^\top Q = I$, the last term is equal to $Tr(B^\top B)$ and the optimization problem (7.6) is equivalent to

$$\arg\max_{Q} Tr(G^\top B Q^\top) \quad \text{subject to.} \quad Q^\top Q = I. \tag{7.8}$$

Let $G^\top B = U\Lambda V^\top$ be the (SVD) for $G^\top B$, the $Tr(G^\top B Q^\top)$ term becomes

$$\begin{aligned} \text{Tr}(U\Lambda V^\top Q^\top) &= \text{Tr}((U\Lambda^{0.5})(\Lambda^{0.5} V^\top Q^\top)) \\ &= \langle U\Lambda^{0.5}, \Lambda^{0.5} V^\top Q^\top \rangle. \end{aligned} \tag{7.9}$$

By the Cauchy-Schwartz inequality, we get

$$\begin{aligned} \langle U\Lambda^{0.5}, \Lambda^{0.5} V^\top Q^\top \rangle &\leq \|(U\Lambda^{0.5})\| \|(\Lambda^{0.5} V^\top Q^\top))\| \\ &= \|\Lambda^{0.5}\| \|\Lambda^{0.5}\| \\ &= \text{Tr}(\Lambda) \end{aligned}$$

due to the invariance of $\|\cdot\|$ under orthogonal transformations. Hence, the sum in (7.9) is maximized if $U^\top Q V = I$ and the solution Q^* to (7.9) is given by $Q^* = UV^\top$. □

As we have seen in (7.8), fixing G, F and S, the computation of Q_g and Q_f can be performed by:

$$\arg\max_{Q_g, Q_g^\top Q_g = I} Tr(G^\top B_g Q_g^\top) \qquad \text{and} \qquad \arg\max_{Q_f, Q_f^\top Q_f = I} Tr(F^\top B_f Q_f^\top).$$

By applying SVD to $B_g^\top G$ and due to Theorem 2 we obtain $Q_g = U_g V_g^\top$. Similarly, applying SVD on $B_f^\top F$ yields $Q_f = U_f V_f^\top$.

- Computation of G: We fix S, F and Q_g. Let $\widetilde{B}_g = B_g Q_g$, G can be updated by $g_{ip}^{(t+1)}$, defined as follows:

$$g_{ip}^{(t+1)} = \begin{cases} 1 & p = \texttt{arg}\min_{p'} ||(\mathbf{z}_{.i})^{(t)} - \mathbf{s}_{p'.}^{(t)}||_{D_g}^2 - 2\alpha(\widetilde{B}_g)_{ip'} \\ 0 & \text{otherwise.} \end{cases} \tag{7.10}$$

- Computation of F: We fix S, G and Q_f. Let $\widetilde{B}_f = B_f Q_f$, we similarly obtain F using $f_{jq'}^{(t+1)}$, defined as follows:

$$f_{jq}^{(t+1)} = \begin{cases} 1 & q = \texttt{arg}\min_{q'} ||(\mathbf{w}_{j.})^{(t+1)} - \mathbf{s}_{q'.}^{(t)})||_{D_f}^2 - 2\beta(\widetilde{B}_f)_{jq'} \\ 0 & \text{otherwise.} \end{cases} \tag{7.11}$$

Notice that the computation of G and F are performed on intermediate reduced matrices Z and W instead the original data matrix X.

7.2.4 CMDC algorithm

Our approach, which we have called Constrained Matrix Decomposition based Co-clustering (CMDC), is summarized in Algorithm 12.

Algorithm 12: CMDC algorithm.

Step 0:
* Initialize $G^{(0)}$ by applying a clustering algorithm on $X^\top$.
* Initialize $F^{(0)}$ by applying a clustering algorithm on X.

Step 1: Compute the normalized graph Laplacians L_g and L_f.

Step 2: Introduce the selected informative *ML* and *CL* constraints in $\tilde{L}_g$ and $\tilde{L}_f$.

Step 3: Computation of $S^{(0)}$ by using Eq.(7.5)

Step 4: Computation of $(G^{(t+1)}, F^{(t+1)}, S^{(t+1)})$ starting from $(G^{(t)}, F^{(t)}, S^{(t)})$;

repeat

- (a) - Calculate $(Z)^{(t)}$ by using Eq.(6.8)
- (b) - Update $G^{(t+1)}$ by using Eq.(7.10)
- (c) - Calculate $(W)^{(t+1)}$ by using Eq.(6.9)
- (d) - Update $F^{(t+1)}$ by using Eq.(7.11)
- (e) - Update $S^{(t+1)}$ by using Eq.(7.5)

until *convergence*;

It should be remarked that a variety of clustering algorithms may be used to initialize G and F. Here we retain *Kmeans* and *SKmeans*, and in Section 7.3 we evaluate their impact on the different algorithms that we are comparing.

7.3 Numerical experiments

In the following subsections we discuss some of the advantages of our contribution in relation to the graph-regularized-based methods : DRCC[Gu and Zhou, 2009], FNMTF and LPFNMTF

[Wang et al., 2011b]. We focus only on the quality of row clustering. We begin by giving a justification for the use of *Skmeans* in appropriate situations. We then look at the computation time of these techniques, and evaluate their performance in semi-supervised learning.

Data sets. Numerical experiments were performed using three types of benchmark data sets from the clustering and co-clustering literature [Ding et al., 2006b; Gu and Zhou, 2009; Wang et al., 2011b], namely image data, document-term data and microarray data. Table 7.1 summarizes the characteristics of these data sets. Note that, for all the used document-term data sets, we apply the TF-IDF transformation on all the document-term frequency matrices.

Table 7.1 – Data sets description.

Data sets	Type	n	d	k	Sparsity (%)
Coil20	Image	1440	1024	20	34.38
Coil100	Image	7200	1024	20	34.38
USPS	Image	9298	256	10	0
CSTR	Document-term	1428	1024	4	96.59
WebACE	Document-term	2340	1000	20	91.83
RCV1	Document-term	9625	29992	4	99.75
Leukemia	Bio	72	5551	2	0
Lung	Bio	203	2008	5	0

Performance metrics. To measure the clustering performance of the proposed algorithms we use three commonly adopted metrics. the accuracy, the Normalize Mutual Information [Strehl and Ghosh, 2002] and the Adjusted Rand Index [Hubert and Arabie, 1985].

Parameter settings. For the sake of fairness we adopt an experimental design similar to [Wang et al., 2011b]. We run each method mentioned above under different parameter settings 50 times, and the average result is computed. We report the best average result for each method. We set the number of sample clusters equal to the true number of classes for all the data sets.

- For DRCC, FNMTF and LPFNMTF; the best parameters are then used, as suggested in each of the reference articles (see for details [Gu and Zhou, 2009; Wang et al., 2011b]). Notice that it was suggested that the number of feature clusters is set to be the same as the number of sample clusters.
- For CMDC, we constructed the graph Laplacian matrix using the distance most suitable for the type of data, i.e. Euclidean distance for image and microarray data sets, and cosine distance for text data sets. Furthermore, the pairwise constraints are obtained each time from 5% of arbitrarily selected labeled data. Finally, the regularization parameters α and β are all searched from the grid $(0.01, 0.1, 1, 10, 100, 500, 1000)$ with $\alpha = \beta$ for both the sample and feature graphs.

- For CMDC, we used *SKmeans* to obtain the initial partition for text data sets. For the image data set and microarray data sets, we used *Kmeans*. Moreover, in order to assess the number of feature clusters, we varied ℓ between 2 and $10k$, and retained the one that optimizes the criterion.

7.3.1 Evaluation of CMDC on real data sets

7.3.1.1 Impact of informative constraints

First, in order to evaluate CMDC in a semi-supervised learning context we applied it on all the tested data sets, varying the percentage of labeled data (from 5% to 80%). We first randomly select the labeled data according to the corresponding percentage, from which we extract all informative *ML* and *CL* constraints. We then integrate the result constraint sets in the graph Laplacian matrices.

The results, reported in Figure 7.1, show clearly that introducing informative *ML* and *CL* constraints consistently improves the co-clustering performance of CMDC for all data.

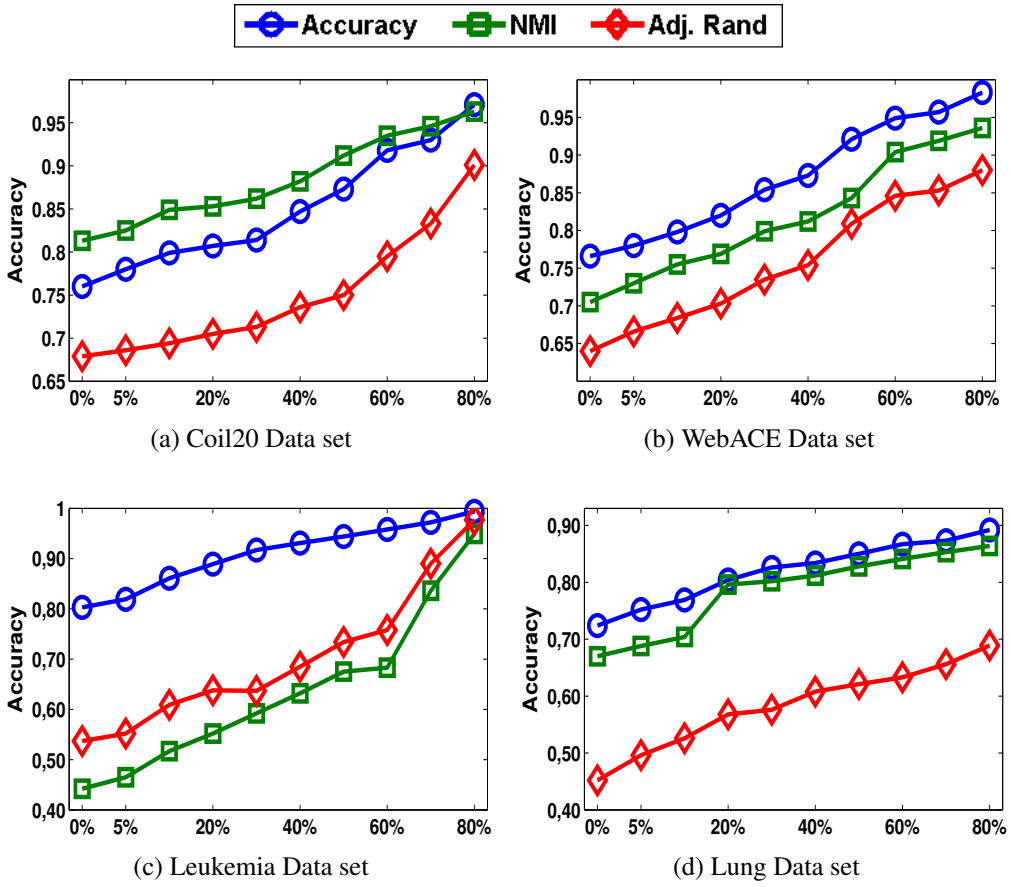
Secondly, in order to compare fairly CMDC, DRCC and LPFNMTF, we run each method under different parameter settings 50 times in both constrained (Const.) and unconstrained (UnConst.) cases. In the constrained case, we introduce the same constraint set in all algorithms and in the same way. We first randomly select 5% of labeled data, from which we extract all informative *ML* and *CL* constraints. We then integrate the result constraint sets in the graph Laplacian matrices. The obtained results are reported in Table 7.2.

Otherwise, To confirm the best behavior of CMDC versus DRCC and LPFNMTF and the impact of *ML* and *CL* constraints on CMDC, we conducted appropriated t-tests for both Constrained and Unconstrained cases, the *p-values* computed for all data sets are less than 1%. It appears clearly that introducing informative *ML* and *CL* constraints improves the clustering quality for all methods but CMDC already provides the best results.

7.3.1.2 Study on regularization parameters α and β

The choice of parameters α and β is not easy. However, through our experiments, we can give indications on the appropriate values to be taken for these two parameters. In Figure 7.2, are reported the performances of the three algorithms in terms of Acc and NMI according values of parameters α and β varying in the interval 0.001 to 1000, we took $\alpha = \beta$.

For all data sets and whatever values of α and β, CMDC outperforms LPFNMTF and DRCC in terms of Acc and NMI. Moreover, for sparse text data sets, the performance of CMDC, in contrary LPFNMTF and DRCC, grows with α and β, it is the best when α and



(a) Coil20 Data set (b) WebACE Data set

(c) Leukemia Data set (d) Lung Data set

Figure 7.1 – CMDC performance according to labeled data rate.

β are higher ($\alpha = \beta = 1000$). However for non-sparse data sets, i.e, image and microarray data sets, small values of α and β appear more interesting. In fact, the rate of sparsity seems an important element in the choice of α and β, we will explore this aspect through simulated data in next section.

7.3.2 Evaluation of CMDC on synthetic data sets

In order to evaluate our approach in term of co-clustering, we propose to evaluate the different algorithms on simulated data sets generated according a probabilistic model (see Appendix A).

To evaluate the three algorithms taking into account the degree of overlapping, the rate of

Table 7.2 – Impact of randomly 5% of labeled data on the compared algorithms.

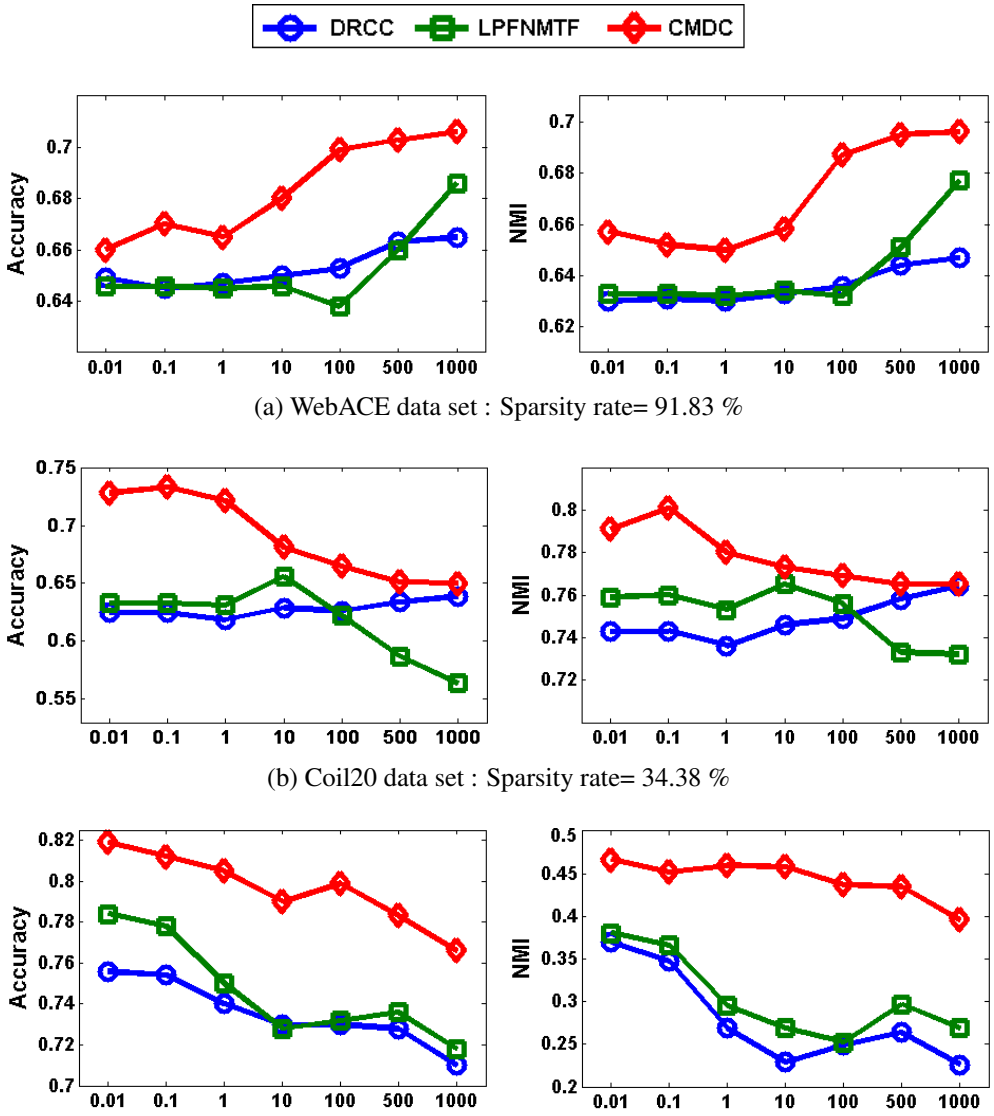
Data Sets	Metric	DRCC		LPFNMTF		CMDC	
		UnConst.	Const.	UnConst.	Const	UnConst.	Const.
Coil20	Acc	0.622	0.640	0.723	0.740	0.760	**0.783**
	NMI	0.747	0.768	0.796	0.808	0.813	**0.825**
	ARI	0.585	0.605	0.643	0.664	0.679	**0.686**
Coil100	Acc	0.486	0.510	0.534	0.560	0.555	**0.580**
	NMI	0.752	0.766	0.793	0.813	0.819	**0.837**
	ARI	0.442	0.463	0.495	0.524	0.514	**0.529**
USPS	Acc	0.655	0.688	0.700	0.716	0.755	**0.771**
	NMI	0.585	0.606	0.610	0.618	0.639	**0.656**
	ARI	0.519	0.548	0.548	0.560	0.580	**0.611**
CSTR	Acc	0.883	0.898	0.908	0.921	0.915	**0.928**
	NMI	0.729	0.732	0.784	0.789	0.794	**0.801**
	ARI	0.735	0.750	0.763	0.802	0.844	**0.846**
WebACE	Acc	0.694	0.678	0.731	0.742	0.766	**0.780**
	NMI	0.642	0.659	0.677	0.682	0.705	**0.730**
	ARI	0.577	0.584	0.609	0.611	0.640	**0.666**
RCV1	Acc	0.726	0.742	0.757	0.768	0.767	**0.783**
	NMI	0.468	0.486	0.504	0.522	0.539	**0.556**
	ARI	0.455	0.484	0.488	0.503	0.538	**0.560**
Leukemia	Acc	0.694	0.722	0.736	0.761	0.803	**0.819**
	NMI	0.224	0.230	0.275	0.385	0.442	**0.465**
	ARI	0.240	0.285	0.363	0.510	0.537	**0.552**
Lung	Acc	0.639	0.664	0.675	0.681	0.684	**0.716**
	NMI	0.453	0.509	0.537	0.576	0.670	**0.688**
	ARI	0.325	0.366	0.395	0.408	0.452	**0.496**

Table 7.3 – Parameters of simulated data sets and error rates for samples, features and global.

Data	Dimension	Classes	Error Rate (%)			Proportions of sample clusters	Proportions of feature clusters
			$e(G, G')$	$e(F, F')$	$\delta = \delta(Y, Y')$		
Data1	**500x500**	**4x3**	8.4	2.6	10	$\boldsymbol{\pi} = [0.2, 0.3, 0.3, 0.2]$	$\boldsymbol{\rho} = [0.3, 0.4, 0.3]$
Data2			13.2	7.4	20		
Data3			25.4	6.2	30		
Data4			35.0	7.0	40		
Data5			43.0	13.0	50		
Data6			38.0	42.8	65		

sparsity and the proportions, we perform extensive experiments and we present error rates or accuracy arising from different simulated tables whose parameters are reported in table 7.3. The main points are the following.

- Figure 7.3 shows that CMDC is always better than LPFNMTF and DRCC whatever the degree of overlapping ($\delta = 0\%$,10%, 20%, 30%,40%, 50% or 65%).
- From initial data sets having various degree of overlapping, we measure the impact of sparsity. In Figure 7.4 are reported the performances of all algorithms according degrees of mixing ($\delta = 0\%$,10%, 20%, 30%,40%, 50% or 65%) and rates of sparsity (0%, 20%, 40% and 60%). We observe the good behavior of CMDC in all situations.
- In order to evaluate the three algorithms in term of cluster proportions, we perform supple-



(a) WebACE data set : Sparsity rate= 91.83 %

(b) Coil20 data set : Sparsity rate= 34.38 %

(c) Leukemia data set : Sparsity rate= 0 %

Figure 7.2 – Co-clustering quality under different values of α and β.

mentary experiments by varying the proportions and the degree of overlapping. To this end, in Figure 7.5, we present the results obtained with Data7, Data8, Data9 and Data10 described in Appendix A. It appears clearly that CMDC is more robust than LPFNMTF and DRCC; even when the proportions are dramatically different it remains the most effective.

- Data7: $\boldsymbol{\pi} = [0.1, 0.4, 0.4, 0.1]$, $\boldsymbol{\rho} = [0.1, 0.8, 0.1]$;
- Data8: $\boldsymbol{\pi} = [0.1, 0.1, 0.1, 0.7]$, $\boldsymbol{\rho} = [0.1, 0.8, 0.1]$;

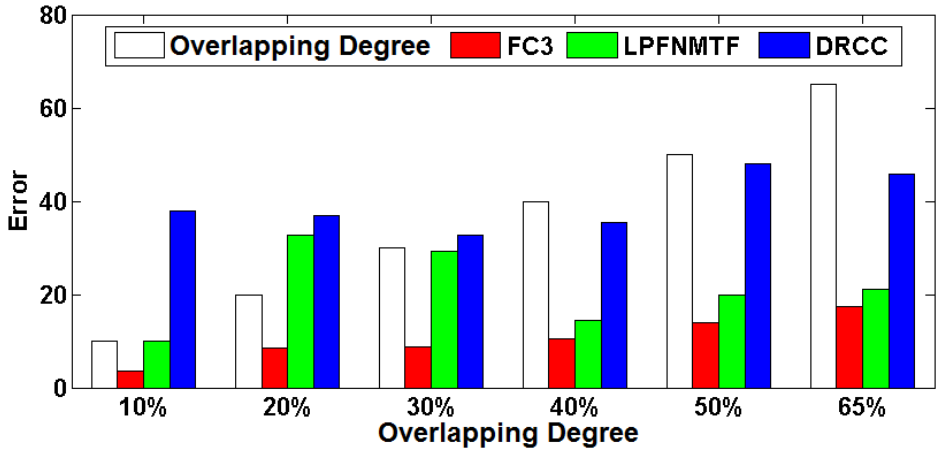


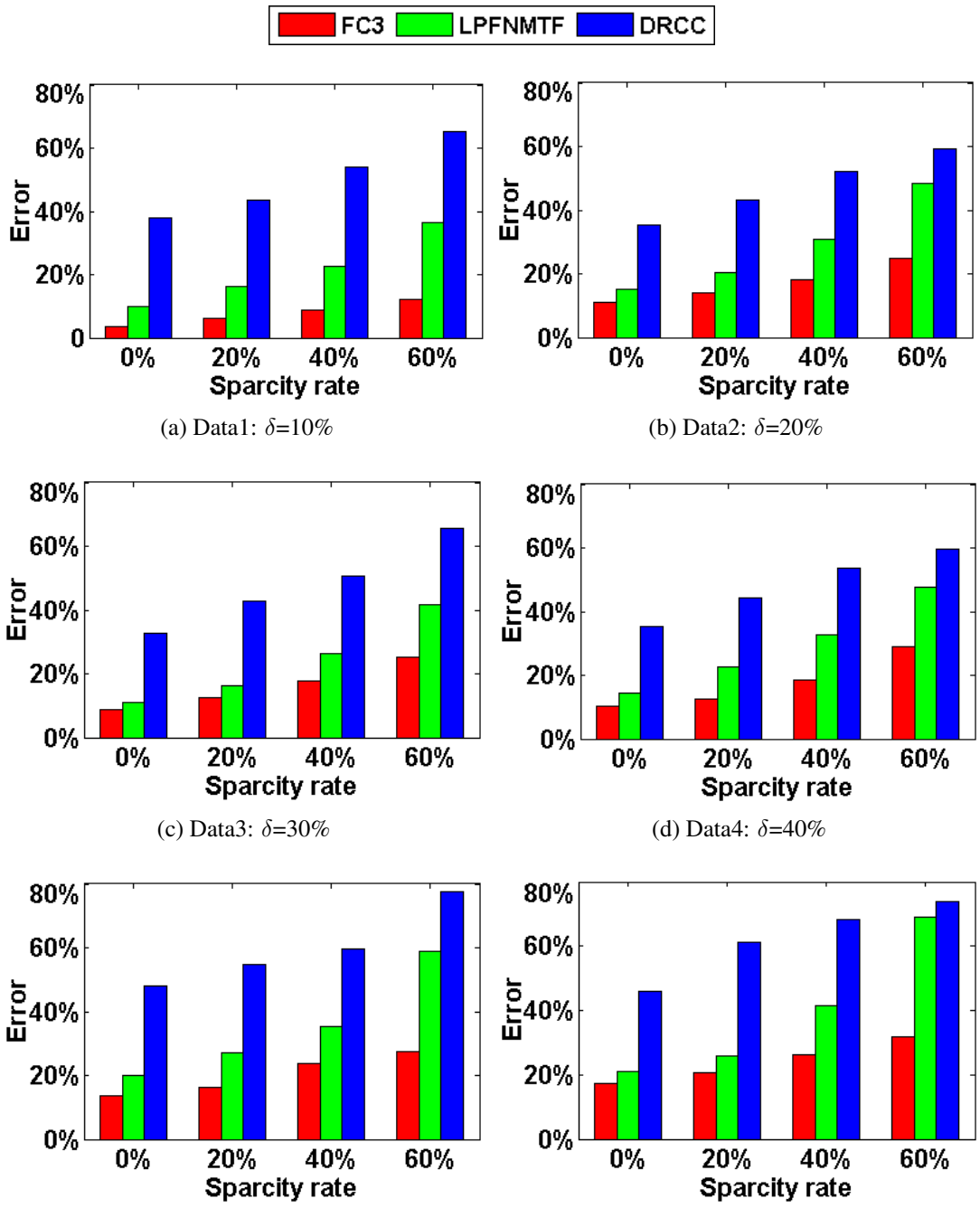
Figure 7.3 – Impact of overlapping

- Data9: $\boldsymbol{\pi} = [0.2, 0.3, 0.3, 0.2]$, $\boldsymbol{\rho} = [0.1, 0.8, 0.1]$;
- Data10: $\boldsymbol{\pi} = [0.1, 0.1, 0.1, 0.7]$, $\boldsymbol{\rho} = [0.3, 0.4, 0.3]$.

- Finally, the obtained results, showed in Figure 7.6, proves that for sparse data sets (Document-term data sets), the accuracy of CMDC grows with α and β, it is the best when α and β are higher. However, for data sets with low sparsity rate (image and microarray data sets), small values of α and β (< 0.1), seems more interesting as we have seen for real data sets.

7.4 Conclusion

In this chapter we describe a novel approach for constrained co-clustering with locality preserving that we called Matrix Decomposition based Co-clustering algorithm (CMDC). This approach is based on low-rank approximation of the binary cluster indicators and the original data. In the semi-supervised context, CMDC treats the co-clustering process by integrating some informative *ML* and *CL* constraints. The selected constraints are introduced in the graph Laplacian matrices in both sample and feature sides. In our experiments on real data sets, it is notable that CMDC outperforms other algorithms designed to solve the same task. It is not only more efficient, but it also requires less computation time. Furthermore, using some synthetic data sets, we investigated the robustness of CMDC in terms of clustering and co-clustering. The overall results showed that even with higher degree of overlapping, high rate of sparsity and proportions of clusters dramatically different, CMDC remains significantly efficient.



(a) Data1: δ=10%

(b) Data2: δ=20%

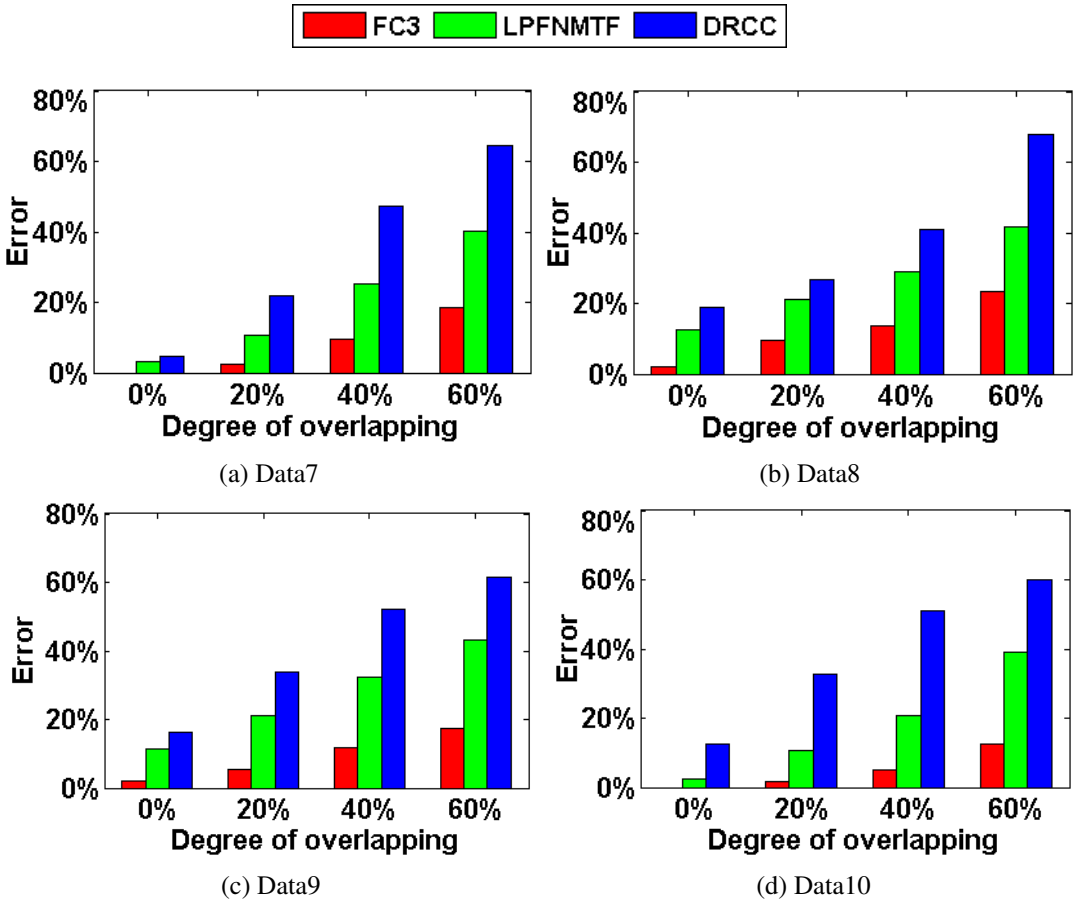
(c) Data3: δ=30%

(d) Data4: δ=40%

(e) Data5: δ=50%

(f) Data6: δ=65%

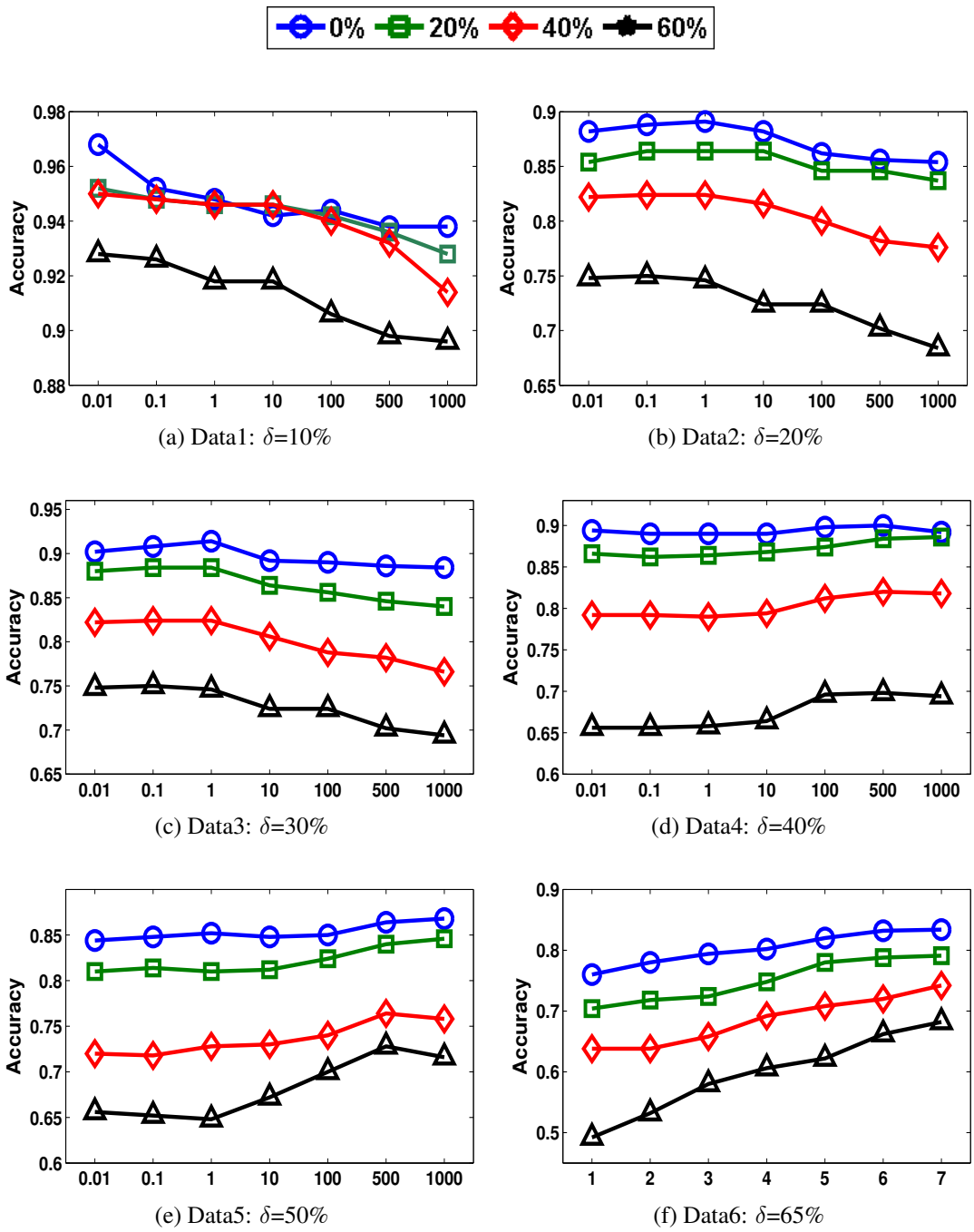
Figure 7.4 – Impact of sparsity



(a) Data7 (b) Data8

(c) Data9 (d) Data10

Figure 7.5 – Impact of cluster proportions



(a) Data1: δ=10%

(b) Data2: δ=20%

(c) Data3: δ=30%

(d) Data4: δ=40%

(e) Data5: δ=50%

(f) Data6: δ=65%

Figure 7.6 – Accuracy in function of α, β and sparsity rate

Chapter 8

Conclusions and perspectives

8.1 Conclusion

In this thesis we have investigated how to apply some unsupervised learning techniques, namely data dimensionality reduction, data clustering and data co-clustering, on the high-dimensional data sets. Unlike to the existing tandem clustering methods that combine a dimension reduction method (e.g PCA) and a clustering method (such as k-means) separately, we provided two convenient ways to integrate the data embedding and the data clustering steps into a single framework which performs the two tasks simultaneously. Furthermore, we have extended the proposed basic methods to tackle the co-clustering problem. Finally, we have proposed two methods to address the problem of co-clustering in two different contexts, the multi-manifold learning and the semi-supervised learning by integration prior knowledge in the model as pairwise *must-link* and *connot-link* constraints. Hereafter, we present in detail the main novel contributions of this thesis in chapters 3-7.

Chapter 3 [Allab et al., 2015b, 2016a]. In the proposed framework, referred to as Simultaneous SemiNMF and PCA for data Clustering (SemiNMF-PCA), we showed how PCA and SemiNMF can be integrated into a single framework to simultaneous data clustering and visualization. Specifically, we showed that the objective learning of Semi-NMF-PCA can be decomposed into two terms, the first one is the objective function of PCA and the second is the Semi-NMF criterion in a low dimensional space. This allows a better approximation of data reduction integrating a clustering solution. We further developed our method to incorporate manifold information and proposed the graph Regularized Fast Semi-NMF-PCA method.

Chapter 4 [Allab et al., 2016b]. We have proposed a framework, referred to as Power Spectral Data Embedding and Clustering (PSDEC), for spectral clustering combining low-dimensional embedding learning and clustering in a common procedure. Then the optimization of a single learning objective function is necessary to achieve spectral embedding and clustering tasks simultaneously. The spectral rotation technique is applied to get the continuous spectral vector which is closer to the cluster membership indicator than existing results. Several experiments revealed that PSDEC is less costly than traditional spectral clustering and better than existing methods commonly used for the same tasks.

Chapter 5 [Allab et al., 2016c]. In the proposed framework, referred to as Regularized SemiNMF-PCA for Co-Clustering (SemiNMF-PCA-Coclust), we showed how PCA and Semi-NMF can be integrated into a single framework of simultaneous data co-clustering and visualization. Specifically, we have extended SemiNMF-PCA algorithm presented in Chapter 3 to

perform SemiNMF via PCA for dimension reduction and data co-clustering. As in chapter 3, we further developed our method to incorporate manifold information of both data samples and data features and proposed the graph Regularized SemiNMF-PCA-Coclust method.

Chapter 6 [Allab et al., 2015a]. Motivated by the potential of dimensionality reduction methods, we proposed to tackle the aim of co-clustering via an ensemble learning. Specifically, we have considered the following well-known dimensionality reduction methods: Canonical Discriminant Analysis (CDA), Multi-Dimensional Scaling (MDS), Isometric Feature Mapping (ISOMAP), Locally Linear Embedding (LLE), Locally Preserving Projections (LPP) and Stochastic Neighbor Embedding (SNE). Next, we have proposed a novel Multi-Manifold Co-clustering algorithm referred as M3DC. It attempts to consider simultaneously the diversity of geometric structures in both the sample manifold and the feature manifold, with the aim of discarding the noisy part in each candidate manifold. In other words, instead of choosing a single manifold learning technique, M3DC considers the idea of applying a set of dimensionality reduction methods and extracting the associated manifolds. By considering both sample and feature manifolds, we aimed to develop an effective co-clustering algorithm.

Chapter 7 [submitted in KBS journal]. The aim of the proposed approach, referred to as Constrained Matrix Decomposition based Co-Clustering (CMDC), was to co-cluster efficiently data sets of different types by introducing the most beneficial prior knowledge on both the sample and feature spaces. Using Laplacian locality preserving, we projected the samples and features into lower-dimensional subspaces, preserving their local geometry. By replacing the original high dimensional data matrix by two low-dimensional intermediate matrices and two low-dimensional landmark-based representations, We have significantly reduced the complexity of the graph construction and the graph Laplacian eigendecomposition. This significantly reduces computational time. Finally, besides the similarity information encoded in the Laplacian graph in both sample and feature sides, CMDC allows to use label information to modify the two graph Laplacians according to the specified *ML* and *CL* constraints. Furthermore, using the measure of informativeness, CMDC selects the constraints that can correct the failures of most of the clustering and co-clustering methods. This is specifically relevant with the presence of some critical data located on the boundaries among the classes.

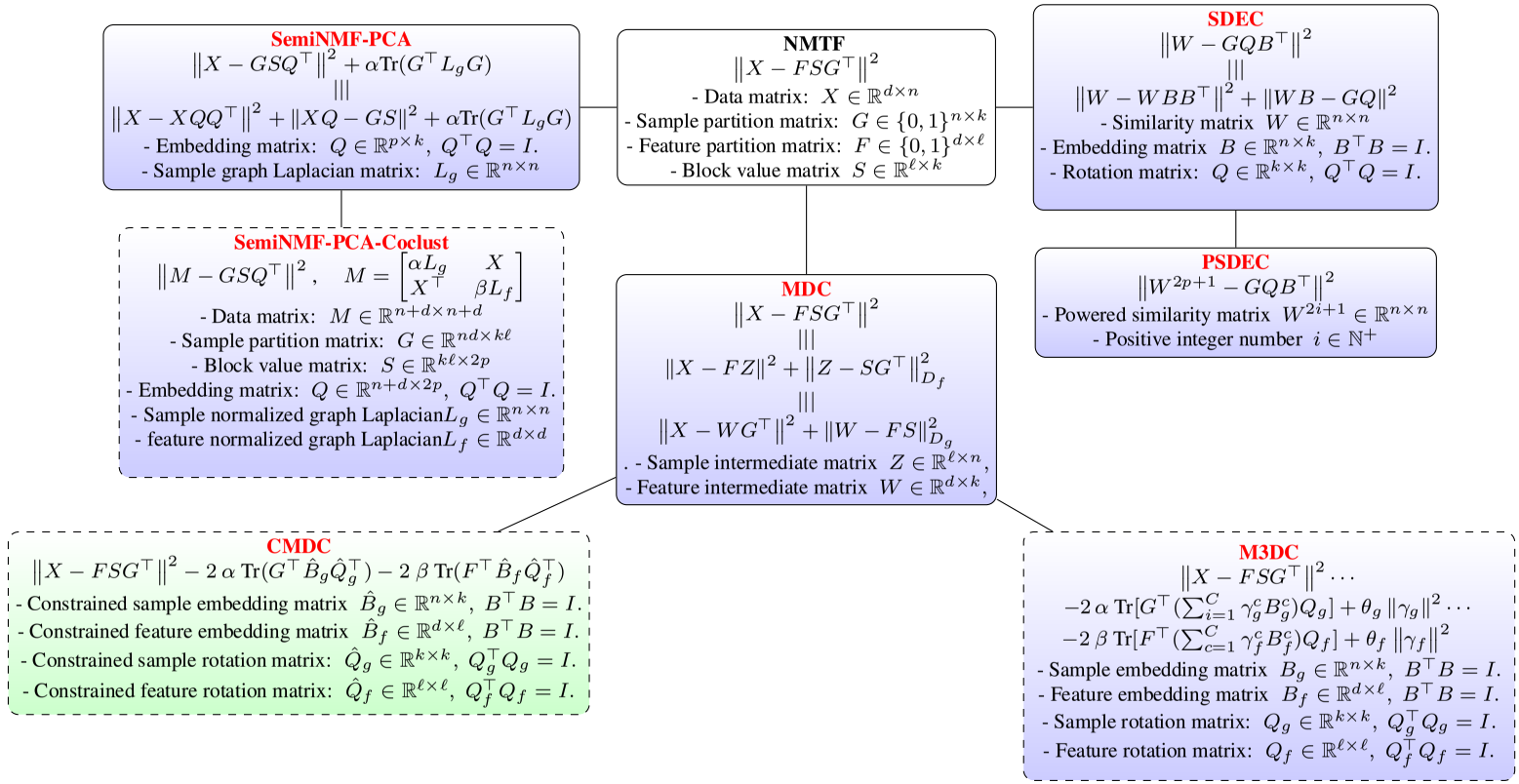


Figure 8.1 – Thesis Flowchart. All our proposed methods are based on Matrix Factorisation. We used solid lines for clustering methods and dashed lines for co-clustering methods. We used blue color for published methods while green color is devoted to unpublished methods. We assume that n is the number of samples, d is the number of features, k is the number of sample clusters, ℓ is the number of feature clusters, p is the number of components and Tr denotes the Trace matrix.

All objectives functions optimized in the thesis are reported in figure 8.1. Note that, they are based on the Frobenius norm. For future work, a promising direction would be to use the I-divergence (or generalised KL-divergence). Further, we can also investigate the use of other regularization terms. For instance, we can add some sparsity constraints by using coordinate descent, $\ell_{1,2}$ regularization or spectral regression. To study the impact of these parameters, a thorough study deserves to be performed.

In Chapter 3, we have proposed a solution to choose the number of components and retained the one that optimizes the criterion of our proposed method RF-Semi-NMF-PCA [Allab et al., 2015b, 2016a]. The obtained results in terms of clustering and visualization are very encouraging. However, the used strategy relies on a certain extent on the number of classes. It should be more beneficial to investigate simultaneously the choice of the number of classes and the number of components for all the proposed eigendecomposition based approaches. This objective will be our major ongoing research activity.

8.2 Personal Publications

I- Journals:

- **K. Allab**, L. Labiod, M. Nadif. **A Semi-NMF-PCA Unified Framework for Data Clustering**. In **TKDE**, *IEEE Transactions on Knowledge and Data Engineering*. 29(1), 2017. (Impact Factor 2016: 2.476).

II- International Conferences:

- **K. Allab**, L. Labiod, M. Nadif. **Simultaneous Semi-NMF and PCA for Clustering**. In **ICDM'15**. *IEEE International Conference on Data Mining*, pages 679-684, Atlantic City-NJ-USA. 2015. (Rank A+).
- **K. Allab**, L. Labiod, M. Nadif. **Multi-Manifold Matrix Tri-Factorization for Text Data Clustering**. In **ICONIP'15**. *Neural Information Processing*, Part I, Lecture Notes in Computer Science 9489, pages 705-715, Istanbul-Turkey. 2015. (Rank A).
- **K. Allab**, L. Labiod, M. Nadif. **Power Simultaneous Spectral Data Embedding and Clustering**. In **SDM'16**. *SIAM International Conference on Data Mining*, pages 270-278, Miami-FL-USA. 2016. (Rank A).
- **K. Allab**, L. Labiod, M. Nadif. **SemiNMF-PCA framework for Sparse Data Co-clustering**. In **CIKM'16**. *ACM International Conference on Information and Knowledge Management*, pages 347-356, Indianapolis-IN-USA. 2016. (Rank A).

III- National Conferences:

- **K. Allab**, L. Labiod, M. Nadif. **Classification croisée sous contraintes basée sur la Tri-Factorisation matricielle** (version longue). In **CAp'14**. *Conférence francophone dapprentissage*, Saint Etienne-France, 2014.
- **K. Allab**, L. Labiod, M. Nadif. **Classification croisée sous contraintes basée sur la Tri-Factorisation matricielle** (version courte). In **SFC'14**. *Rencontre de la Société Francophone de Classification*, Rabat-Maroc, 2014.
- **K. Allab**, L. Labiod, M. Nadif. **La Classification Croisée par Combinaison de Variétés**. In **SFC'15**. *Rencontre de la Société Francophone de Classification*, Nantes-France, 2015.

IV- Submitted papers:

- **K. Allab**, L. Labiod, M. Nadif. **Multi-Manifold Matrix Decomposition for Data Co-clustering**. *Revised version submitted in Pattern Recognition*, since 25/04/2016.
- **K. Allab**, L. Labiod, M. Nadif. **Simultaneous Spectral Data Embedding and Clustering**. *Submitted in TNNLS, IEEE Transactions on Neural Networks and Learning Systems*, since 27/06/2016.
- **K. Allab**, L. Labiod, M. Nadif. **A Regularized Constraint-based Algorithm for Data Co-clustering**. *Submitted in KBS, Knowledge-Based Systems*, since 20/07/2016.

Appendix A

Introduction. The work summarised in this thesis has implied conducting a wide array of experiments to test in practice intuitions and insights. This Appendix gathers some aspects common to these experiments, so as to avoid unnecessary repetitions along the thesis. An adequate choice of the data collections over which the experiments are going to be performed is of a capital importance to ensure the validity of the conclusions drawn from them. Not only the greatest similarity with the data that the clustering algorithms would be dealing with in a real-world situation should be sought, but also benchmark data sets must be always used, in so far as possible. In the work summarised in this thesis we have used benchmark and widely available data sets. The specific details of the data sets and splits used in each experiment will be discussed in the corresponding chapter of this thesis.

Textual document-term data sets. The experiments were performed using some benchmark document-term data sets from clustering and co-clustering literature. We used 9 real data sets, each with different sizes and balances[1]. Below, we present a description of each used data set.

- **CSTR:** [Li, 2005] Contains the abstracts of technical reports (TRs) published in the Department of Computer Science at University of Rochester from 1991 to 2007. There are 550 abstracts and they are divided into 4 research areas: Natural Language Processing, Robotics/Vision, Systems, and Theory. We also use the category information of terms obtained from ACM Keywords Taxonomy as prior knowledge.
- **WEBACE:** [Ding and Li, 2007] Contains news articles partitioned across 20 different topics obtained from the WEBACE project [Han et al., 1998].
- **CLASSIC3 and CLASSIC4**[2] Consists respectively of 3 different document collections: CISI, CRANFIELD, and MEDLINE for classic3 and 4 different document collections: CACM, CISI, CRANFIELD, and MEDLINE for classic4.
- **NG20:** [Zhong and Ghosh, 2005] is a collection of usenet articles divided into 20 different categories. We also include the **NG10** data set, a subset of **NG20** that contains the 10 topics.
- **RCV1:** [He, 2012] is a subset of a newswire stories corpus made available by Reuters containing 4 categories: C15, ECAT, GCAT, and MCAT.
- **SPORTS and REVIEWS:** [Zhong and Ghosh, 2005] Are two data sets from the CLUTO[3] toolkit that are collected from the San Jose Mercury newspapers articles. Reviews contains 5 document categories (food, movies, music, radio and restaurant) and sports contains 7 documents categories (baseball, basketball, bicycling, boxing, football, golfing and hockey).

[1] The balance coefficient is defined as the ratio of the number of documents in the smallest class to the number of documents in the largest class.

[2] http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets

[3] http://glaros.dtc.umn.edu/gkhome/views/cluto

Image data sets. To assess our approaches on other data with low sparsity, experiments were performed using some benchmark image data sets from the clustering literature.

- **Coil-100:** The Columbia object image library (COIL-100)[1] is a set of color images of 100 different objects taken from different angles (in steps of 5 degrees) at a resolution of 128×128 pixels.

Figure 8.2 – Coil-100 Data set

- **Coil-20:** The COIL-20 database [2] is an image library from Columbia which contains 20 objects. The images of each object were taken 5 degrees apart as the object is rotated on a turntable and each objects has 72 images. The size of each image is 32×32 pixels, with 256 grey levels per pixel.

Figure 8.3 – Coil-100 Data set

- **ORL:** ORL face database consists of a total of 400 face images, of a total of 40 subjects (10 samples per subject). The images were captured at different times and have different variations including expressions (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees. The original images were normalized (in scale and orientation) such that the two eyes were aligned at the same position. Then, the facial areas were cropped into the final images for matching. The size of each cropped image is 32×32 pixels, with 256 grey levels per pixel. Thus, each face image can be represented by a 1024-dimensional vector.

[1] available at http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php
[2] available at http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

Figure 8.4 – ORL Data set

- **Yale:** The Yale database consists of 165 face images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, with glasses, happy, left-light, no glasses, normal, right-light, sad, sleepy, surprised, and wink. We pre-processed these original images by aligning transformation and scaling transformation so that the two eyes were aligned at the same position. Then, the facial areas were cropped into the resulting images.

Figure 8.5 – Yale Data set

- **CMU PIE:** The CMU PIE face data set contains 41368 facial images of 68 people, The face images are captured under 43 different light and illumination conditions, and each person has 42 facial images (each person under 13 different poses and with 4 different expressions). Original images were normalized (in scale and orientation) so that the two eyes were aligned at the same position. then, the facial areas were cropped into the final experimental images. The size of each cropped image is 32×32 pixels, with 256 gray levels per pixel. Thus, each face image is represented by a 1024-dimensional vector in image space.

Figure 8.6 – CMU PIE Data set

- **USPS:** USPS digit database is one of the standard data sets for handwritten digit recognition. It contains 9298 normalized grey scale images of size 16×16, divided into a training set of 7291 images and a test set of 2007 images.

Figure 8.7 – USPS Data set

- **MNIST:** is a database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

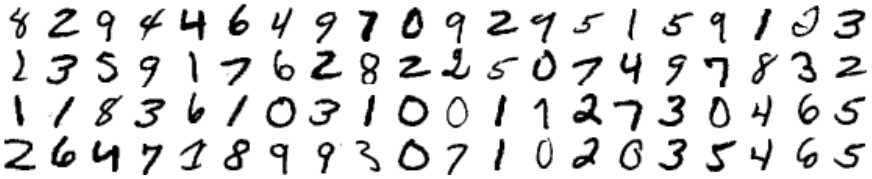


Figure 8.8 – MNIST Data set

Microarray data sets. Microarrays allow measuring the expression level of a large number of genes under different experimental samples or environmental conditions. The data generated from them are called gene expression data. The extraction of biological relevant knowledge from this data is not a trivial task.

- **Leukemia:** Consists of a set of 72 examples and two types of Leukemia, 47 ALL (Acute Lymphocytic Leukemia) and 25 AML (Acute Myelogenous Leukemia). The data contains initially 7929 features. In [Busygin et al., 2002] it was suggested deleting the control features "affymetrix" and the features having a value less than 20 (biologically, the low levels of expression are difficult to interpret), to obtain finally 1762 features.
- **Lung cancer:** This data set has been obtained from a total of 203 snap-frozen specimens composed of 186 lung tumors and 17 normal lung samples. Lung tumors include 139 adenocarcinomas, 21 squamous cell lung carcinomas, 20 pulmonary carcinoids, and 6 small-cell lung carcinomas (SCLC). mRNA expression levels of $12,600$ transcript sequences from samples are hybridized for biologically distinct subclasses of lung adenocarcinoma [Bhattacharjee et al., 2001].

- **Colon cancer:** Murali and Kasif used a colon cancer data set originated in to test XMOTIF. The matrix contains 40 colon tumor samples and 22 normal colon samples over about 6500 genes. Colon cancer data set is available at http:// www.weizmann.ac.il/physics.
- **Breast Cancer:** Contains 78 patient samples, 34 of which are from patients who had developed distance metastases within 5 years (labeled as relapse), the rest 44 samples are from patients who remained healthy from the disease after their initial diagnosis for interval of at least 5 years (labeled as nonrelapse). We used a version of data set contains $23,625$ genes and 32 relapse samples and 44 non-relapse samples.
- **Yeast:** We used the same gene expression data sets as used by Cheng and Church [Cheng and Church, 2000]. The yeast Saccharomyces cerevisiae cell cycle expression data set contains 2884 genes and 17 conditions. The gene expression values were mapped into the range 0 and 600 and missing values were represented by -1 in the yeast data set.

FCPS and Shape synthetic data sets. In order to illustrate the efficiency of some algorithms, we used some generated synthetic data sets selected from the Fundamental Clustering Problem Suite (FCPS)[1] and the Shape data sets[2]. FCPS and Shape data sets poses some hard clustering problems.

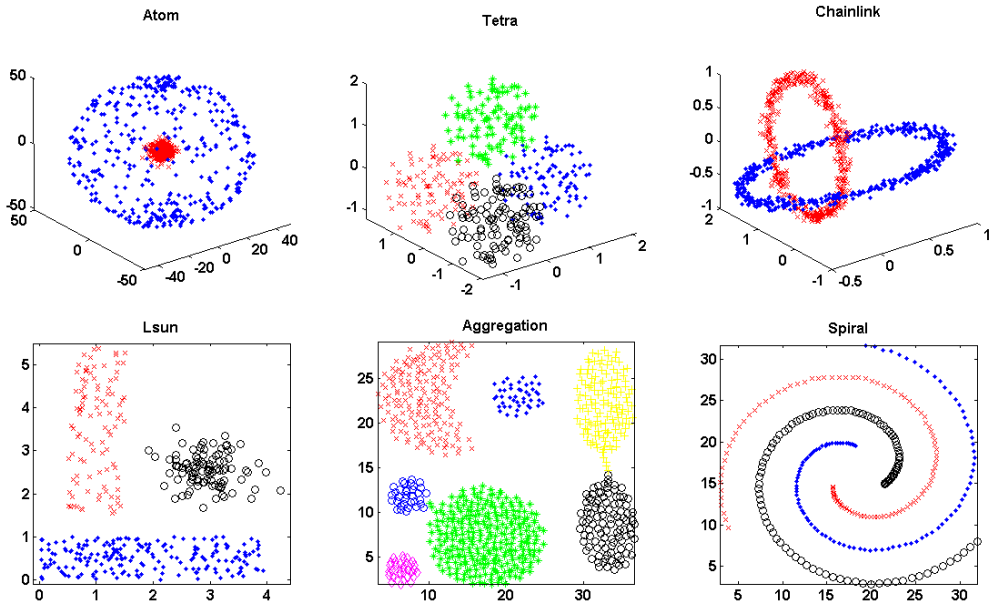


Figure 8.9 – FCPS and Shape synthetic data sets

[1] Can be downloaded from: http://www.uni-marburg.de/fb12/datenbionik/data
[2] Shape: http://cs.joensuu.fi/sipu/datasets/

SwissRoll data set. In order to illustrate the ability of some methods to preserve the initial topology and their capability to separate classes, we used a generated synthetic data set called SwissRoll (1600×3) with 4 classes (400 samples in each class). The original data was created by randomly sampling from a Gaussian Mixture Model with centers/means at $(7.5, 7.5), (7.5, 12.5), (12.5, 7.5)$ and $(12.5, 12.5)$. The covariance for each Gaussian is the 2×2 identity matrix. These points are generated in 2-dimensions, and then map them to 3-dimensions with the Swiss Roll mapping $(\mathbf{x}, \mathbf{y}) \longmapsto (\mathbf{x}\cos(\mathbf{x}), \mathbf{y}, \mathbf{y}\sin(\mathbf{y}))$. Figure 8.10 illustrates some manifold projections obtained using some reduction dimension methods, where the clusters are obtained thanks to k-means.

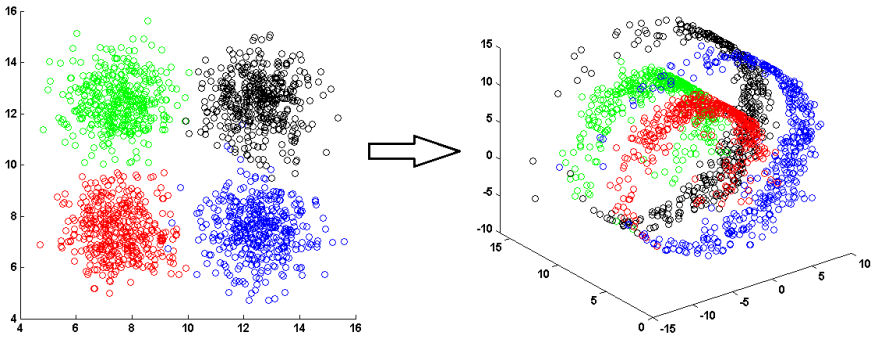


Figure 8.10 – SwissRoll Data set

Generated data sets for Co-clustering. Several papers have been devoted to co-clustering, however in the most cases the proposed algorithms are evaluated only in terms of clustering and not co-clustering. This is due to the non-availability of real data sets where row and column classes are generally simultaneously known. To this end, we propose to evaluate the three algorithms on simulated data sets generated according the *Latent block model* which is extensively studied in [Govaert and Nadif, 2003, 2005, 2008, 2014].

Latent block model. In the classical mixture model, the mixture density of the observed data X or likelihood can be expressed

$$f(X, \boldsymbol{\theta}) = \prod_i \sum_p \pi_p \varphi(\mathbf{x}_{.i}; \boldsymbol{\lambda}_p),$$

where $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\lambda})$ with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_p)$ denoting the proportions of clusters and $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p)$ denoting the parameters of densities φ.

The probability density function $f(X, \boldsymbol{\theta})$ can be written as $\sum_{G \in \mathcal{G}} p(G; \theta) f(X|G; \boldsymbol{\theta})$ [Govaert and Nadif, 2003], where $\mathcal{G}$ denotes the set of all possible assignments of samples into k

clusters,

$$p(G;\theta) = \prod_{i,p} \pi_k^{g_{ip}} \text{ and } f(X|G;\theta) = \prod_{i,p} \varphi(\mathbf{x}_{.i};\boldsymbol{\lambda}_k)^{g_{ip}}.$$

Table 8.1 – Parameters of simulated data sets and error rates for samples, features and global.

Data	Dimension	Classes	Error Rate (%)			Proportions of sample clusters	Proportions of feature clusters
			$e(G,G')$	$e(F,F')$	$\delta = \delta(Y,Y')$		
Data1	**500x500**	**4x3**	8.4	2.6	10	$\boldsymbol{\pi} = [0.2, 0.3, 0.3, 0.2]$	$\boldsymbol{\rho} = [0.3, 0.4, 0.3]$
Data2			13.2	7.4	20		
Data3			25.4	6.2	30		
Data4			35.0	7.0	40		
Data5			43.0	13.0	50		
Data6			38.0	42.8	65		

In co-clustering, the formulation of $f(X,\theta)$ can be extended to propose a latent block model defined by the following probability density function [Govaert and Nadif, 2003]:

$$\begin{aligned} f(X,\boldsymbol{\theta}) &= \sum_{(G,F)\in\mathcal{G}\times\mathcal{F}} p(G;\boldsymbol{\theta})p(F;\theta)f(X|G,F;\boldsymbol{\theta}) \\ &= \sum_{(G,F)\in\mathcal{G}\times\mathcal{F}} \prod_{i,p} \pi_p^{g_{ip}} \prod_{j,q} \rho_q^{f_{jq}} f(X|G,F;\boldsymbol{\theta}), \end{aligned}$$

where $\boldsymbol{\theta} = (\pi_1,\ldots,\pi_p,\rho_1,\ldots,\rho_q,\lambda_{11},\ldots,\lambda_{pq})$ and $\mathcal{G}$ and $\mathcal{F}$ denote the sets of all possibles assignments G of samples into k clusters and F of features into l clusters.

In this model we also assume local independence i.e., the $d \times n$ random variables X_{ij} are assumed to be independent once F and G are fixed; we have

$$f(X|G,F;\boldsymbol{\theta}) = \prod_{i,j,p,q} \varphi(X_{ij};\lambda_{pq})^{g_{ip}f_{jq}}$$

where $\varphi(.;\lambda_{pq})$ is a probability density function defined.

Assuming that for each block $k\ell$ the values X_{ij} are distributed according to a Gaussian distribution (μ_{pq},σ_{pq}^2) with $\mu_{pq} \in \mathbb{R}$ and $\sigma_{pq}^2 \in \mathbb{R}^+$, the density φ with $\lambda_{pq} = (\mu_{pq},\sigma_{pq}^2)$ is the following

$$\varphi(X_{ij},\mu_{pq},\sigma_{pq}^2) = \frac{1}{\sqrt{2\pi\sigma_{pq}^2}} \exp - \left\{ \frac{1}{2\sigma_{pq}^2}(X_{ij}-\mu_{pq})^2 \right\}.$$

Parsimonious model can be defined by imposing constraints on the proportions or variances. In [Nadif and Govaert, 2010], considering the co-clustering under the classification mixture approach, the authors shown that the criterion optimized by Croeuc is associated to a parsimonious Gaussian latent block model where the proportions of sample clusters and fea-

ture clusters are assumed equal respectively. Next, we rely on the Gaussian latent block model to generate the data according different parameters while controlling the degree of mixing. Different parameters are used and some situations are reported in table 8.1.

Error rate and degree of overlapping One characteristic of a mixture model is the degree of mixing or overlapping among the components. In the classical situation, this concept of cluster separation can be visualized, for instance by using *principal component analysis* (PCA), but this concept of cluster separation is difficult to be applied to the latent block models. Another solution is to compute the true error rate associated to the model, which is defined as the expectation of the misclassification probability $\mathbb{E}(\delta((G,F),d(X))$ where G, F and X are the random variables associated to the latent block model, d is the optimal Bayes rule $d(X) = (G',F') = \arg\max_{G,F} p(G,F|X)$ associated to this model and δ is the error rate.

This expectation is generally difficult to be computed theoretically, and Monte Carlo simulations are used to estimated it by the proportion of misclassified, for instance in the classical clustering situation, between the partition simulated with those we obtained by applying a classification step from the true parameters $\hat{\theta}$. This parameter being fixed, the problem is to determine the "best" partitions G and F, that is the pair of partitions G, F maximizing respectively the posterior probability (for details, see [Govaert and Nadif, 2014]):

$$\begin{cases} g_{ip} = \arg\max_{p'} \ \pi_p \prod_{j,q} \varphi(X_{ij}, \mu_{p'q}, \sigma^2_{p'q})^{f_{iq}} \\ f_{jq} = \arg\max_{q'} \ \rho_q \prod_{i,p} \varphi(X_{ij}, \mu_{pq'}, \sigma^2_{pq'})^{g_{ip}}. \end{cases}$$

The proportion of misclassified can be defined as follows: if C is the confusion matrix between the two partitions, relabel the components of the partition G' such that the trace of matrix C is maximal, then compute $e(G,G') = 1 - \frac{1}{n}\sum_{i,p} g_{ip}g'_{ip}$ which is (1-Acc). This definition can be extended to the comparison of two pairs of partitions $Y = (G,F)$ and $Y' = (G',F')$ as follows:

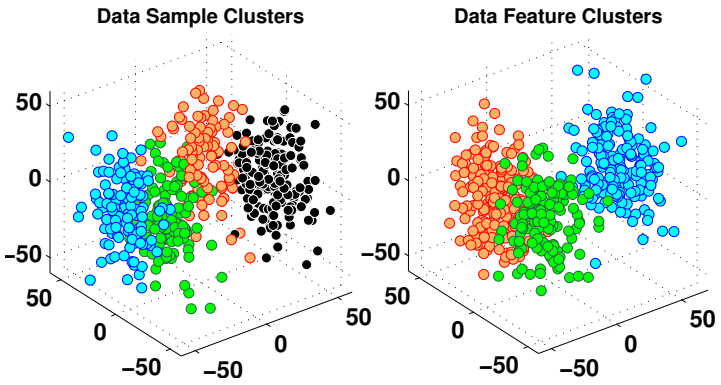
$$\delta(Y,Y') = \delta((G,F),(G',F')) = 1 - \frac{1}{nd}\sum_{i,j,p,q} y_{ijk\ell}y'_{ijpq},$$

where $y_{ijpq} = g_{ip}f_{jq}$ and $y'_{ijpq} = g'_{ip}f'_{jq}$ and, it can be shown that

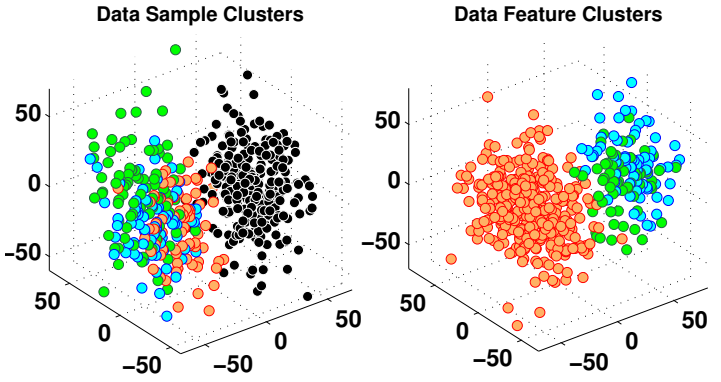
$$\delta(Y,Y') = e(G,G') + e(F,F') - e(G,G') \times e(F,F').$$

Then we can simulate with the expected error. We perform this process several times. First,

we have simulated data sets of size 500×500 with different degree of overlapping. Figure 8.11 illustrates two data sets with degree of overlapping equal to 10% and 65% .



(a) Data with degree of overlapping = 10 %

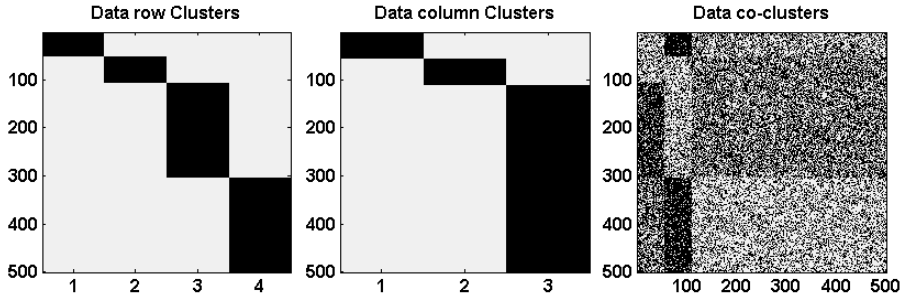


(b) Data with degree of overlapping = 65 %

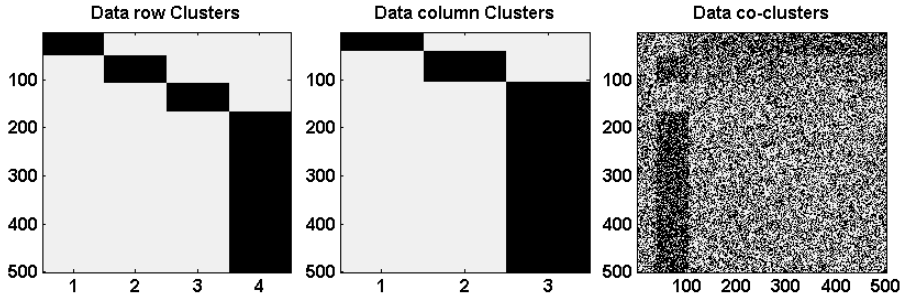
Figure 8.11 – Visualisation of simulated data sets by PCA

Furthermore, in order to evaluate some algorithms in term of cluster proportions, we four simulate data sets of size 500×500 by varying the proportions and the degree of overlapping. Figure 8.12 illustrate the obtained data sets. To this end, the generated data sets are obtained with:

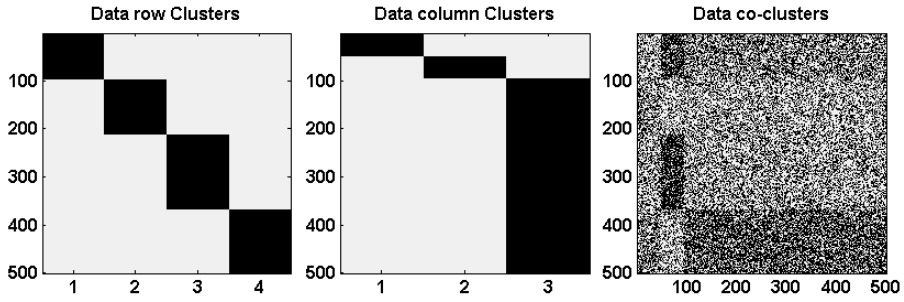
- Data7: $\boldsymbol{\pi} = [0.1, 0.4, 0.4, 0.1], \boldsymbol{\rho} = [0.1, 0.8, 0.1]$;
- Data8: $\boldsymbol{\pi} = [0.1, 0.1, 0.1, 0.7], \boldsymbol{\rho} = [0.1, 0.8, 0.1]$;
- Data9: $\boldsymbol{\pi} = [0.2, 0.3, 0.3, 0.2], \boldsymbol{\rho} = [0.1, 0.8, 0.1]$;
- Data10: $\boldsymbol{\pi} = [0.1, 0.1, 0.1, 0.7], \boldsymbol{\rho} = [0.3, 0.4, 0.3]$.



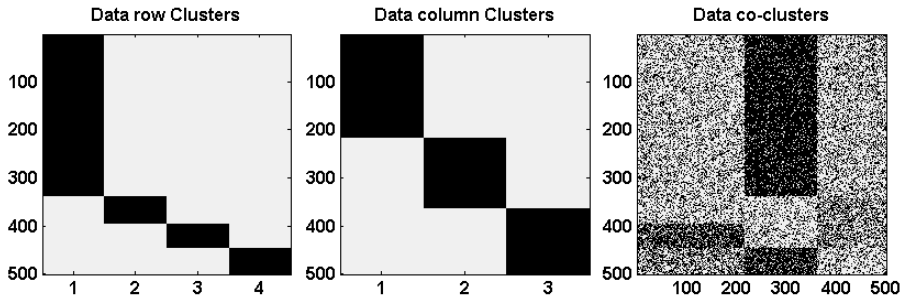
(a) Data7: $\boldsymbol{\pi} = [0.1, 0.4, 0.4, 0.1]$, $\boldsymbol{\rho} = [0.1, 0.8, 0.1]$



(b) Data8: $\boldsymbol{\pi} = [0.1, 0.1, 0.1, 0.7]$, $\boldsymbol{\rho} = [0.1, 0.8, 0.1]$



(c) Data9: $\boldsymbol{\pi} = [0.2, 0.3, 0.3, 0.2]$, $\boldsymbol{\rho} = [0.1, 0.8, 0.1]$



(d) Data10: $\boldsymbol{\pi} = [0.1, 0.1, 0.1, 0.7]$, $\boldsymbol{\rho} = [0.3, 0.4, 0.3]$

Figure 8.12 – Simulated data sets with various cluster proportions

References

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014. 19

Allab, K., Labiod, L., and Nadif, M. (2015a). Multi-Manifold Matrix Tri-Factorization for Text Data Clustering. In *Neural Information Processing*, volume 1, pages 705–715. 7, 141

Allab, K., Labiod, L., and Nadif, M. (2015b). Simultaneous Semi-NMF and PCA for Clustering. In *IEEE ICDM'15*, pages 679–684. 6, 68, 140, 143

Allab, K., Labiod, L., and Nadif, M. (2016a). A Semi-NMF-PCA Unified Framework for Data Clustering. *To appear in IEEE Transactions on Knowledge and Data Engineering*. 6, 140, 143

Allab, K., Labiod, L., and Nadif, M. (2016b). Power Simultaneous Spectral Data Embedding and Clustering. In *SIAM SDM'16*, pages 270–278. 6, 140

Allab, K., Labiod, L., and Nadif, M. (2016c). SemiNMF-PCA framework for Sparse Data Co-clustering. In *To appear in ACM CIKM'16*. 7, 140

Anagnostopoulos, A., Dasgupt, A., and Kumar, R. (2008). Approximation algorithms for co-clustering. In *PODS'08*, pages 201–210. 2

Anderberg, M. R. (1973). *Cluster Analysis for Applications*. Academic Press: New York. 11

Arabie, P. and Hubert, L. (1994). Cluster analysis in marketing research. *In* Bagozzi, R.P. (Eds.), Advanced Methods of Marketing Research. *Oxford, Blackwell*, pages 160–189. 4, 28

Azran, A. and Ghahramani, Z. (2006). Spectral methods for automatic multiscale data clustering. In *IEEE CVPR'06*, pages 190–197. 64, 65

Bach, F. R. and Jordan, M. I. (2006). Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, 7:1963–2001. 56

Baier, D., Gaul, W., and Schader, M. (1997). Two-mode overlapping clustering with applications to simultaneous benefit segmentation and market structuring. In Klar, R. and Opitz, O., editors, *Classification and knowledge organization*, Heidelberg. Springer. 95

Banerjee, A., Dhillon, I. S., Ghosh, J., Merugu, S., and Modha, D. (2007). A generalized maximum entropy approach to bregman co-clustering and matrix approximation. *Journal of Machine Learning Research*, 8:1919–1986. 18

Barbakh, W. and Fyfe, C. (2008). Clustering and visualization with alternative similarity functions. In *The WSEAS AIKED08*, pages 238–244. 16

Beck, A. and Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31 (3):167–175. 106

Belkin, M. and Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS'01*. 36, 81

Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., and Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(1):155–173. 22

Bezdek, J. (1981). Pattern recognition with fuzzy objective function algorithms. *Plenum Press, New York, NY, USA*. 13

Bezdek, J. and Pal, N. (1992). *Fuzzy Models for Pattern Recognition*. IEEE press, New York, NY, USA. 13

Bhattacharjee, A., Richards, W. G., Staunton, J., C. Li, S. M., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E. J., Lander, E. S., Wong, W., Johnson, B. E., Golub, T. R., Sugarbaker, D. J., and Meyerson, M. (2001). Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *National Academy of Sciences of the United States of America*, 98(24):13790–13795. 149

Boutsidis, C., Kambadur, P., and Gittens, A. (2015). Spectral clustering via the power method - provably. In *ICML'15*, pages 40–48. 64, 69

Busygin, S., Jacobsen, G., and Kramer, E. (2002). Double conjugated clustering applied to leukemia microarray data. In *SIAM ICDM, Workshop on clustering high dimensional data*. 149

Cai, D., He, X., Han, J., and Zhang, H.-J. (2006). Orthogonal laplacianfaces for face recognition. *IEEE Transactions on Image Processing*, 15(11):3608–3614. 40, 70

Cai, R., Lu, L., and Cai, L. (2005). Unsupervised auditory scene categorization via key audio effects and information-theoretic co-clustering. *IEEE ICASSP'05*, pages 1073–1076. 18

Chan, P. K., Schlag, M., and Zien, J. Y. (1994). Spectral k-way ratio-cut partitioning and clustering. *IEEE Transactions on CAD*, 13(9):1088–1096. 36, 56

Charrad, M., Lechevallier, Y., Ahmed, M. B., and Saporta, G. (2009). Block clustering for web pages categorization. In *Intelligent Data Engineering and Automated Learning-IDEAL 2009*, pages 260–267. Springer. 18

Chen, W.-Y., Song, Y., Bai, H., Lin, C.-J., and Chang, E. (2011). Parallel spectral clustering in distributed systems. *IEEE TPAMI*, 33(3):568–586. 16

Chen, Y., Wang, L., and Dong, M. (2010). Non-negative matrix factorization for semisupervised heterogeneous data coclustering. *In TKDE*, pages 1459–1474. 5, 26, 124

Chen, Y., Wang, L., Dong, M., and Hua, J. (2009). Exemplar-based visualization of large document corpus (infovis2009-1115). *IEEE Transactions on Visualization and Computer Graphics*, 15:1161–1168. 88

Cheng, Y. and Church, G. (2000). Biclustering of expression data. In *AAAI'00*, pages 93–103. 2, 19, 150

Cho, H., Dhillon, I. S., Guan, Y., and Sra, S. (2004). Minimum sum-squared residue co-clustering of gene expression data. In *SIAM SDM'04*, pages 114–125. 95

Collins, M., Dasgupta, S., and Schapire, R. (2001). A generalization of principal component analysis to the exponential family. *NIPS'01*. 30

Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. Wiley, New York. 12

Davidson, I., Wagstaff, K. L., and Basu, S. (2006). Measuring constraint-set utility for partitional clustering algorithms. In *ECPP-KDD'06*. 124, 125

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society.*, 39 (01):1–38. 15

Dhillon, I., Mallela, S., and Modha, D. (2003a). Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98. ACM. 2

Dhillon, I. S. (2001a). Co-clustering documents and words using bipartite spectral graph partitioning. In *ACM SIGKDD'01*, pages 269–274. 2, 40, 78, 80, 83, 84

Dhillon, I. S. (2001b). Co-clustering documents and words using bipartite spectral graph partitioning. In *ACM SIGKDD'01*, pages 269–274. 18, 20

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means, spectral clustering and normalized cuts. In *ACM SIGKDD'04*, pages 551–556. 56

Dhillon, I. S., Mallela, S., and Kumar, R. (2002). Enhanced word clustering for hierarchical text classification. *ACM SIGKDD'02*, pages 191–200. 18

Dhillon, I. S., Mallela, S., and Kumar, R. (2003b). A divisive infomation-theoretic feature clustering algorithm for text classification. *Machine Learning Research*, 3:1265–1287. 18, 79, 83, 84

Dhillon, I. S., Mallela, S., and Modha, D. S. (2003c). Information-theoretic co-clustering. *ACM SIGKDD'03*, pages 89–98. 2

Diday, E. (1971). Une nouvelle méthode en classification automatique et reconnaissance des formes la méthode des nuées dynamiques. *Revue de statistique appliquée*, 19(2):19–33. 19

Ding, C. and He, X. (2004). K-means clustering via principal component analysis. *ICML'04*. 30, 36

Ding, C., He, X., and Simon, H. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. In *SIAM SDM'05*. 5, 24, 25, 35, 40, 56, 60, 70

Ding, C., He, X., Zha, H., Gu, M., and Simon, H. (2001). A min max cut algorithm for graph partitioning and data clustering. In *IEEE ICDM'01*, pages 107–114. 24, 56

Ding, C. and Li, T. (2007). Adaptive dimension reduction using discriminant analysis and k-means clustering. In *ICML'07*, pages 521–528. 146

Ding, C., Li, T., and Jordan, M. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE TPAMI*, 32(1):45–55. 29, 35, 39, 40

Ding, C., Li, T., and Peng, W. (2006a). Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method. In *Proceedings of the national conference on artificial intelligence*, volume 21 (1), page 342. 24

Ding, C., Li, T., Peng, W., and Park, H. (2006b). Orthogonal nonnegative matrix trifactorizations for clustering. In *ACM SIGKDD'06*, pages 126–135. 2, 5, 24, 26, 34, 35, 83, 109, 129

Engel, D., Huttenberger, L., and Hamann, B. (2012). A survey of dimension reduction methods for high-dimensional data analysis and visualization. *IRTG 1131 Workshop, Germany.*, 27:135–149. 101, 103

Everitt, B. S. and Dunn, G. (2001). *Applied Multivariate Data Analysis*. Arnold, London. 11

Fan, M., Qiao, H., Zhang, B., and Zhang, X. (2012). Isometric multi-manifold learning for feature extraction. In *IEEE ICDM'12*, pages 241–250. 3, 102

Fielder, M. (1975). A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(100):619–633. 16

Freitag, D. (2004). Trained named entity recognition using distributional clusters. *EMNLP'04*, pages 262–269. 18

Friedman, J. (1994). An overview of predictive learning and function approximation. *Statistics to Neural Networks*, Springer Berlin Heidelberg:1–61. 4

Gao, B., Liu, T., Zheng, X., Cheng, Q., and Ma, W. (2005a). Consistent bipartite graph copartitioning for star-structured high-order heterogeneous data co-clustering. *SIGKDD'05*, pages 41–50. 18

Gao, B., Liu, T.-Y., Feng, G., Qin, T., Cheng, Q.-S., and Ma, W.-Y. (2005b). Hierarchical taxonomy preparation for text categorization using consistent bipartite spectral graph copartitioning. *IEEE TKDE*, 17(9):1263–1273. 20, 23

Gaussier, E. and Goutte, C. (2005). Relation between plsa and nmf and implications. In *ACM SIGIR'05*, pages 601–602. 24

Gillis, N. (2014). The why and how of nonnegative matrix factorization. *Regularization, Optimization, Kernels, and Support Vector Machines*, pages 257–291. 22

Gittins, R. (1985). *Canonical Analysis A Review with Applications in Ecology*. Springer-Verlag. 100, 101, 103

Goldberg, A. B., Zhu, X., Singh, A., Xu, Z., and Nowak, R. (2009). Multi-manifold semi-supervised learning. In *ICAIS'09*, pages 169–176. 3, 102

Golub, G. H. and van Loan, C. F. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA. 57, 64, 68

Gordon, A. D. (1999). *Classification (2nd ed.)*. Chapman and Hall/CRC, Boca Raton, FL. 11

Govaert, G. (1983). *Classification croisée*. PhD thesis, Université Paris 6, France. 19, 83, 84

Govaert, G. (1995). Simultaneous clustering of rows and columns. *Control and Cybernetics*, 24:437–458. 78

Govaert, G. and Nadif, M. (2003). Clustering with block mixture models. *Pattern Recognition*, 36(2):463–473. 2, 20, 151, 152

Govaert, G. and Nadif, M. (2005). An em algorithm for the block mixture model. *IEEE TPAMI*, 27:643–647. 2, 151

Govaert, G. and Nadif, M. (2008). Block clustering with bernoulli mixture models: Comparison of different approaches. *Comput. Stat. and Data Anal.*, 52:3233–3245. 20, 151

Govaert, G. and Nadif, M. (2010). Latent block model for contingency table. *Communications in Statistics - Theory and Methods*, 39(03):416–425. 20

Govaert, G. and Nadif, M. (2014). *Co-Clustering: Models, Algorithms and Applications*. Wiley. 2, 20, 94, 151, 153

Gu, Q. and Zhou, J. (2009). Co-clustering on manifolds. In *ACM SIGKDD'09*. 3, 26, 35, 79, 80, 83, 84, 97, 98, 108, 109, 128, 129

Guan, J., Qie, G., and Xue, X. Y. (2005). Spectral images and features coclustering with application to content-based image retrieval. *IEEE International Workshop on Multimedia Signal Processing (MMSP'05)*. 18

Hagen, L. W. and Kahng, A. B. (1992). New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on CAD*, 11(9):1074–1085. 24, 56

Han, E.-H., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., and Moore, J. (1998). Webace: a web agent for document categorization and exploration. In *International Conference on Autonomous Agents*, pages 408–415. ACM. 146

Hansen, P. and Jaumard, B. (1997). Cluster analysis and mathematical programming. *Mathematical Programming*, pages 191–215. 14

Hartigan, J. (1972). Direct clustering of data matrix. *American Stat. Assoc.*, 67(337):123–129. 19, 78, 124

Hartigan, J. (1975). *Clustering Algorithms*. Wiley and Sons, 99th edition. 10, 19

Hatzivassiloglou, V. and McKeown, K. (1993). Towards the automatic identification of adjectival scales: Clustering adjectives according to meaning. In *Annual Meeting of the Association for Computational Linguistics*, pages 172–182. 12

He, D. C. X. (2012). Manifold adaptive experimental design for text categorization. *IEEE TKDE*, 24(4):707–719. 146

Hinneburg, A. and Keim, D. (1998). An efficient approach to clustering in large multimedia databases with noise. In *AAAI'98*, pages 58–65. 15

Hoffman, T. and Puzicha, J. (1999). Latent class models for collaborative filtering. *IJCAI'99*, pages 688–693. 19

Hoppner, F., Klawonn, F., Kruse, R., , and Runkler, T. (1999). Fuzzy cluster analysis. *J. Wiley and Sons, Chichester, England*. 13

Hoyer, P. (2002). Non-negative sparse coding. *IEEE Workshop on Neural Networks for Signal Processing*, pages 557–565. 22

Huang, J., Nie, F., Huang, H., and Ding, C. (2014). Robust manifold nonnegative matrix factorization. *ACM TKDD*, 8(3):1–21. 24

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, pages 193–218. 17, 39, 70, 84, 108, 129

Jain, A. and Dubes, R. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 11

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys (CSUR)*, 31(3):264–323. 14

Jolliffe, I. (2002). Principal component analysis. *Springer, 2nd edition*. 30

Jun, H. and Richi, N. (2015). Robust clustering of multi-type relational data via a heterogeneous manifold ensemble. *ICDE'15*. 102, 108

Karaboga, D. and Ozturk, C. (2010). Fuzzy clustering with artificial bee colony algorithm. *Scientific Research and Essays*, 5(14):1899–1902. 13

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., and Ueda, N. (2006). Learning systems of concepts with an infinite relational model. *National Conference on Artificial Intelligence*, pages 381–388. 19

Kendall, M. and Gibbons, J. (1990). *Rank correlation methods (5th edition)*. New York : Oxford University Press. 12

Kim, H. and Park, H. (2007). Sparse non-negative matrix factorizations via alternating nonnegativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502. 23, 24

Klugar, Y., Basri, R., Chang, J., and Gerstein, M. (2003). Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 13:703–716. 19

Kokiopoulou, E., Chen, J., and Saad, Y. (2011). Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications*, 18(3):565–602. 104

Kong, D., Ding, C., and Huang, H. (2011). Robust nonnegative matrix factorization using l21-norm. In *ACM CIKM'11*, page 673. ACM Press. 24

Kuang, D., Park, H., and Ding, C. H. Q. (2012). Symmetric nonnegative matrix factorization for graph clustering. In *SIAM SDM'12*, pages 106–117. 68

Kulis, B., Basu, S., Dhillon, S., and Mooney, R. (2005). Semi-supervised graph clustering: a kernel approach. In *ICML'05*, pages 457–464. 125

Labiod, L. and Nadif, M. (2011). Co-clustering under nonnegative matrix tri-factorization. In *ICONIP'11*. 79, 83, 84

Lazzaroni, L. and Owen, A. (2002). Plaid models for gene expression data. *Statistica Sinica*, 12:61–86. 19

Lee, D. and Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791. 2, 20

Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. *NIPS'01*, pages 556–562. 20

Lee, L. (2001). On the effectiveness of the skew divergence for statistical language analysis. In *AISTATS*. Citeseer. 12

Li, H. and Abe, N. (1998). Word clustering and disambiguation based on co-occurrence data. *Annual Meeting of the Association for Computational Linguistics and International Conference on Computational Linguistics (COLING-ACL)*, pages 188–197. 18

Li, P., Bu, J., Chen, C., and He, Z. (2012). Relational co-clustering via manifold ensemble learning. In *CIKM '12*, pages 1687–1691. 102, 108

Li, S., Hou, X., Zhang, H., and Cheng, Q. (2001). Learning spatially localized, parts-based representation. *IEEE CVPR'01*, 1:207–212. 22

Li, T. (2005). A general model for clustering binary data. In *KDD'05*, pages 188–197. 78, 146

Li, T. and Ding, C. (2006). The relationships among various nonnegative matrix factorization methods for clustering. *IEEE ICDM'06*, pages 362–371. 5, 24, 25

Li, T., Ding, C., Zhang, Y., and Shao, B. (2008). Knowledge transformation from word space to document space. *ACM SIGIR'08*, pages 187–194. 26

Lin, C. J. (2007). Projected gradient methods for nonnegative matrix factorization. *Journal of Neural Computation*, 19(10):2756–2779. 63, 66

Lin, F. and Cohen, W. (2010). Power iteration clustering. In *ICML'10*, pages 655–662. 64

Liu, W., Zheng, N., and Lu, X. (2003). Non-negative matrix factorization for visual coding. *IEEE ICASSP'03*, 3:293–299. 22

Long, B., Wu, X., Zhang, Z., and Yu, P. S. (2006). Spectral clustering for multi-type relational data. *ICML'06*, pages 585–592. 20

Long, B., Zhang, Z., Y., and S., P. (2007). A probabilistic framework for relational clustering. *ACM SIGKDD'07*, pages 470–479. 19

Long, B., Zhang, Z., and Yu, P. S. (2005). Unsupervised learning on k-partite graphs. In *ACM SIGKDD'05*, pages 317–326. 2, 5

Lovász, L. and Plummer, M. (2009). *Matching Theory*. AMS Chelsea Publishing Series. American Mathematical Soc. 17

Lu, J., Tan, Y.-P., and Wang, G. (2013). Discriminative multi manifold analysis for face recognition from a single training sample per person. *IEEE TPAMI*, 35:39–51. 3, 102

Luo, D., Huang, H., Ding, C., and Nie, F. (2010). On the eigenvectors of p-laplacian. *Journal of Machine Learning*, 81(1):37–51. 56

Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416. 16, 56, 58

Madeira, S. and Oliveira, A. (2004). Biclustering algorithms for bilogical data analysis: a survey. *IEEE Transactions on Computational Biology and Bioinformatics*, 1:24–45. 18

McLachlan, G. and Peel, D. (2004). *Finite mixture models*. John Wiley & Sons. 15

McQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *the Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. 14

Meila, M. and Shi, J. (2001). Learning segmentation by random walks. In *NIPS'01*, pages 873–879. 59, 60, 65

Nadif, M. and Govaert, G. (2010). Model-based co-clustering for continuous data. In *ICMLA'10*, pages 175–180. 152

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *NIPS'01*, pages 849–856. 16, 56, 59

Nie, F., Ding, C., Luo, D., and Huang, H. (2010). Improved minmax cut graph clustering with nonnegative relaxation. In *ECML/PKDD'10*, volume 6322, pages 451–466. 56

Pauca, V., Piper, J., and Plemmons, R. (2006). Nonnegative matrix factorization for spectral data analysis. *Linear Algebra and its Applications*, 416(1):29–47. 23

Pauca, V., Shahnaz, F., Berry, M., and Plemmons, R. (2004). Text mining using nonnegative matrix factorizations. *SIAM SDM'04*, pages 452–456. 23

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. 41, 84

Pensa, R., Boulicaut, J.-F., Cordero, F., and Atzori, M. (2010). Co-clustering numerical data under user-defined constraints. *Statistical Analysis and Data Mining*, 3(1):38–55. 19

Pensa, R. G. and Boulicaut, J.-F. (2008). Constrained co-clustering of gene expression data. In *'SIAM SDM'08*, pages 25–36. 5, 19, 124

Qie, G. (2004). Image and feature co-clustering. *ICPR'04*, pages 991–994. 18

Rocci, R. and Vichi, M. (2008). Two-mode multi-partitioning. *Computational Statistics and Data Analysis*, 52(4):1984–2003. 94, 95

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65. 18, 114

Santos, J., de Sa, J. M., and Alexandre, L. (2008). Legclust: A clustering algorithm based on layered entropic subgraphs. *IEEE TPAMI*, 30(1):62–75. 16

Scheffé, H. (1959). *The Analysis of Variance*. New York: Wiley. 49, 113

Schonemann, P. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10. 33, 62, 69

Shahnaz, F., Berry, M., Pauca, P., and Plemmons, R. (2006). Document clustering using nonnegative matrix factorization. *Information Processing and Management*, 42(2):373–386. 23

Shan, H. and Banerjee, A. (2008). Bayesian co-clustering. In *ICDM'08*, pages 530–539. 19

Shang, F., Jiao, L. C., and Wang, F. (2012). Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognition*, 45(6):2237–2250. 3

Sheikholesami, G., Chatterjee, S., and Zhang, A. (1998). Wavecluster: A multiresolution clustering approach for very large spatial databases. *ICVLD'98*, pages 428–439. 15

Shen, B. and Si, L. (2010). Nonnegative matrix factorization clustering on multiple manifolds. *AAAI'10*, pages 575–580. 24

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE TPAMI*, 22 (8):888–905. 16, 24, 56

Sibson, R. (1973). Slink: an optimally efficient algorithm for the single-link cluster method. *Comput. Journal*, 16:30–34. 14

Sindhwani, V., Hu, J., and Mojsilovic, A. (2008). Regularized co-clustering with dual supervision. *NIPS'08*, pages 1505–1512. 26

Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438. 14

Sokal, R. R. and Sneath, P. H. A. (1963). *Principles of numerical taxonomy*. Freeman and Co., San Francisco. 11

Song, Y., Pan, S., Liu, S., Wei, F., Zhou, M., and Qian, W. (2010). Constrained co-clustering for textual documents. In *AAAI'10*. 5, 124

Sorensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biologiske Skrifter*, 5:1–34. 14

Souvenir, R. and Pless, R. (2005). Manifold clustering. *IEEE ICCV'05*, pages 648–653. 102

Strehl, A. and Ghosh, J. (2002). Cluster ensembles:a knowledge reuse framework for combining multiple partitions. *Machine Learning Research*, pages 583–617. 17, 39, 70, 84, 108, 129

Takamura, H. and Matsumoto, Y. (2003). Co-clustering for text categorization. *Information Processing Society of Japan Journal*. 18

Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18:36–44. 19

Timmerman, M., Ceulemans, E., Kiers, H., and Vichi, M. (2010). Factorial and reduced k-means reconsidered. *Computational Statistics & Data Analysis*, 54(7):1858 – 1871. 4

van der Maaten, L., Postma, E. O., and van den Herik, H. J. (2008). Dimensionality reduction: A comparative review. 101, 103

Verma, D. and Meila (2003). A comparison of spectral clustering algorithms. *Technical Report UW-CSE-03-05-01, Washington University*. 16

Vichi, M. and Kiers, H. (2001). Factorial k-means analysis for two-way data. *Computational Statistics & Data Analysis*, 37(1):49–64. 4, 28

Vichi, M. and Saporta, G. (2009). Clustering and disjoint principal component analysis. *Computational Statistics & Data Analysis*, 53(8):3194 – 3208. 4

Wang, D., Li, T., Zhu, S., and Ding, C. (2008a). Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *SIGIR'08*, page 307. 23

Wang, F., Ding, C., and Li, T. (2009). Integrated kl (kmeans-laplacian) clustering: A new clustering approach by combining attribute data and pairwise relations. In *SIAM SDM'09*, pages 38–48. 125

Wang, F., Li, T., and Zhang, C. (2008b). Semi-supervised clustering via matrix factorization. In *SIAM SDM*, pages 1–12. 5, 26, 124

Wang, H., Huang, H., and Ding, C. (2011a). Simultaneous clustering of multi-type relational data via symmetric nonnegative matrix tri-factorization. In *CIKM'11*, pages 279–284. 5, 79

Wang, H., Nie, F., Huang, H., and Makedon, F. (2011b). Fast nonnegative matrix tri-factorization for large-scale data co-clustering. In *IJCAI'11*. 3, 26, 80, 83, 84, 97, 98, 108, 109, 129

Wang, Y., Jiang, Y., Wu, Y., and Zhou, Z. (2010). Multi-manifold clustering. In *PRICAI'2010*, pages 280–291. 108

Wang, Y., Jiang, Y., Wu, Y., and Zhou, Z. (2011c). Spectral clustering on multiple manifolds. *IEEE TNNLS*, 22(7):1149–1161. 102, 108

Wang, Y. and Zhang, Y.-J. (2013). Nonnegative matrix factorization: A comprehensive review. *IEEE TKDE*, 25(6):1336–1353. 3, 22

Xu, W., Liu, X., and Gong, Y. (2003). Document clustering based on non-negative matrix factorization. *ACM SIGIR'03*, pages 267–273. 22

Yang, J., Wang, W., Wang, H., and Yu, P. (2003). Enhanced biclustering on expression data. *BIBE'03*, pages 321–327. 19

Yang, W., Sun, C., and Zhang, L. (2011). A multi-manifold discriminant analysis method for image feature extraction. *Pattern Recognition*, 44(8):1649–1657. 3, 102

Yoo, J. and Choi, S. (2010). Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds. *Information processing & management*, 46(5):559–570. 5

Zass, R. and Shashua, A. (2005). A unifying approach to hard and probabilistic clustering. In *IEEE ICCV'05*, pages 294–301. 5

Zhang, Z. and Zha, Z. (2004). Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal of Scientific Computing*, 26:313–338. 36, 81

Zhirong, Z. Y. and Laaksonen, J. (2007). Projective nonnegative matrix factorization with applications to facial image processing. *Journal of Pattern Recognition and Artificial Intelligence*, 21(8):1353–1362. 32, 40, 70

Zhong, S. and Ghosh, J. (2005). Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3):374–384. 146