SEMI-SUPERVISED CLUSTERING TECHNIQUES FOR CATEGORIZATION OF TEXT DOCUMENTS

Yan Yang

School of Electrical & Electronic Engineering

A thesis submitted to Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2013

Acknowledgments

I am indebted to many people for their help and support I received during my Ph.D study and research at Nanyang Technological University.

First and foremost, I would like to express my sincere thanks to my supervisor Dr. Chen Lihui for the invaluable guidance, frequent meetings and discussions, and the encouragement on my way of the research.

I would like to thank my senior graduate fellow Tjhi William Chandra and Nguyen Duc Thang for their friendship and help in my research through the brain storming and discussions.

I also would like to thank Hu Yao, Yunke, Tianchi and all the other FYP students who have helped me in the project.

Special thanks also to all the very helpful technicians: Ms. How in Media Technology Lab and Christina in Software Engineering Lab for being so helpful to create a very nice research environment in the lab and all kinds of technical help and advices.

Last and most importantly, I wish to thank my dear father, as the only relative left in the world to me, for their constant love and encouragement all the way during the time I was suffered.

Table of Contents

Summary

Nowadays, data mining becomes a very important research filed for knowledge discovery process. Among various data mining techniques, we focus on studying how a small amount of prior knowledge can be effectively incorporated into some popular clustering techniques, not only to improve the existing models, but also to develop novel semi-supervised clustering methods, especially for the categorization of high dimensional text documents. To be more specific, our objective is to investigate into some of the key performance characteristics of a good semi-supervised clustering method, and make them achievable in our purposed methods, such as: how to effectively incorporate the knowledge to guide the cluster search, accurately capture the underlying structure of the data, and achieve high capability of handling overlaps etc. In other words, our final goal is to develop some simple, fast and highly applicable clustering methods which aim to achieve good results with high quality and effectiveness with the help of the prior knowledge.

Fuzzy co-clustering (FCC) is a type of clustering approaches that has shown its capability for handling high dimensional textual data categorization by simultaneously grouping the documents and words into some co-clusters. Meanwhile, for a document which intuitively spans multiple topics, FCC is also able to capture the degree of memberships of that document to each topic. Under the FCC framework, we proposed three different semi-supervised approaches, namely Semi-Supervised Fuzzy Co-clustering with Labelling (SS-FCL), Semi-Supervised Fuzzy Co-clustering with Constraints (SS-FCC) and Dual Semi-Supervised Heuristic Fuzzy Co-clustering with Ruspini's condition (DSS-HFCR), respectively. Two types of prior knowledge in the forms of class labels and pair-wise constraints from the document domain are incorporated into SS-FCL and SS-FCC through different additional supervised constraint terms. Other than the categorization results of the documents, these two approaches also generate a group of work ranking clusters, which are useful in other data mining techniques, such as text summarization. Meanwhile, a heuristic dual-partitioning based approach called DSS-HFCR is also proposed in order to make full use of the available prior knowledge in terms of pair-wise constraints from both document and word domain. Moreover, DSS-HFCR can be directly downgraded to a simplified version if the prior knowledge is available from only a single domain Through extensive experimental

study on a number of benchmark textual datasets, we demonstrate how these approaches make good use of the knowledge to guide the cluster search during the clustering process for a better performance in terms of both accuracy and efficiency. Some useful guidelines for parameter selection are also discussed. A case study of sentiment data analysis by applying DSS-HFCR demonstrates the strength of our proposed methods in the specific application area.

FCC model is suitable for handling large sparse textual datasets, as it avoids applying an explicit similarity measure between two documents. However, similarity measure is still one of the most essential factors in many discriminative clustering approaches, and most of the these approaches still make use of only a single reference point (viewpoint) i.e. the origin for the similarity assessment. It is interesting and challenging to explore more effective similarity measures for high dimensional textual data, especially when some prior knowledge is available to the user.

For the second part of the thesis, inspired by a recently proposed multi-viewpoint based similarity measure (MVS) [1], we introduce another novel semi-supervised clustering framework, which is able to utilize multiple appropriate viewpoints for a more informative and effective similarity assessment by incorporating two types of knowledge. With the help of a small number of class labels or pair-wise constraints in the dataset, we formulate two MVS measures, and subsequently propose two new MVS-based clustering approaches: Label-based Clustering with Multi-Viewpoint based Similarity (namely LMVS) and Pair-wise Constraints-based Clustering with Multi-Viewpoint based Similarity (namely PMVS). Comparing with the existing semi-supervised clustering techniques, the key strength of LMVS and PMVS is a more effective similarity measure can be directly formulated in the MVS manner with the help of the knowledge, and immediately applied to clustering, rather than learned by an independent distance metric learning process before the real clustering process is carried out. Some validity tests are conducted to show the strength of the measures, and systematical theoretical analysis is also provided to explain how the prior knowledge is utilized for both similarity enhancement and search-guiding purpose during the clustering process. At the same time, some potential issues of MVSC reported in [1] can be successfully addressed. At last, extensive experimental study on a large number of benchmark textual datasets are presented to demonstrate the effectiveness and verify the merit of LMVS & PMVS Clustering, compared with other start-of-the-art semi-supervised clustering/learning approaches.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

In today's information era, there is a huge volume of data generated from almost every single field through sensors, computer and the Internet. All organizations have collected huge amount of data in their databases. Undoubtedly, knowledge discovery from such massive data becomes ultimately important for these organizations to discover useful knowledge as patterns or models from their data [2]. People always pay attention to how to get their desire patterns/information accurately and efficiently. Data mining is a step in the knowledge discovery process consisting of particular machine learning algorithms that, under some acceptable computational efficiency limitations, find patterns or models hidden among the volumes of data [2].

We are living in the age of information, as the biggest source of data; the World Wide Web is a fertile area for data mining research. By just focusing on the heavy load of information shared through the Internet, we realize it is increasing extremely fast every second, meanwhile, the ability of the current corresponding data storage systems and the data mining techniques is not able to match that speed very closely. For example, according to the real time statistical information provided by Microsoft, there are billions of new emails sent every day, the total number of web pages is in the order of trillion and still increasing in order to billion due to the popularization of on-line social networking, such as facebook, twitters. Therefore, the demanding of doing web mining, which could be defined as the application of data mining on the Web, is also huge and critical to every Internet user. Unlike data mining of normal databases in which data are stored in uniformed structures, web data mining is performed on various types of data, such as content, hyperlinks, web logs click information and user profiles. Clustering is one of the most popular un-supervised data mining techniques to automatically find natural groupings of data based on certain pre-defined similarity measures. The objects in the same cluster are most similar to each other in content, while at the same time are also dissimilar to those in other clusters. It is widely applied in informational retrieval- e.g.: categorizing text documents into meaningful sub-groups based on the topics [3], medical diagnosis- e.g.: medical examination based on the cancer image [4],

image segmentation-e.g.: area-recognition on GPS landing data [5] , bioinformatics- e.g.: new protein structure discovery by clustering the microarray gene data [6], market analysis- clustering on customers according to their characteristics and historical transaction data for strategic decision making, etc.

These examples show the huge benefits that clustering can potentially provide. Be specific to the web data, it is told by that most of these sources are actually in the form of unstructured text. This brings us to a hot topic: text clustering. Document clustering or text clustering, is then a specific area within the data clustering field that we have introduced above, to provide the engine for us to categorize and organize the information in huge amount of web textual documents automatically and efficiently. This kind of data is large, overlapping, noisy, dynamic and often embedded in a very high dimensional space (indicating the number of words are huge), and therefore become very sparse as each document may only contain a very small portion of the total vocabulary. Other types of data, such as microarray genes, also express this property at different levels. The main theme of this thesis is about novel concepts and techniques that are applied for clustering this kind of sparse and high-dimensional data; other multimedia data, such as image, video, audio is not our focus [7].

Unlike supervised learning method, such as classification, clustering does not have and require categorical label to train. Therefore, sometimes it is really quite challenge for a completely un-supervised learning approach to make some complicated data well categorized for the user. On the other side, the cost of introducing the training process into clustering algorithm (so become a classification algorithm) could be very expensive since a large number of labels are required. Instead, to improve the performance, semi-supervised clustering, which directly incorporate a small amount of available prior knowledge (also named as side information) in various forms such as class labels, pair-wise constraints, into the clustering process with a suitable design, may be a good choice.

1.2 Motivation

Despite the extensive research work has been done on the data clustering field for last a few decades, the challenges on developing an effective clustering method for the high dimensional data still remains due to many unsolved issues involved in the real-world problems, especially when the method has to be totally un-supervised. These challenges are the main causes that motivate us to look into the problem persistently and carry on the research work in this field. The shortlist of some main challenges is presented as below:

- **Stability**: Quite a few clustering algorithms are sensitive to their initialization state. Their performance could become greatly different with different initialized values. However, in practical this is not desirable since we would always prefer the system

keeping stable. Therefore, question like: "Is there a method capable on reducing the sensitiveness to initialization of the existing algorithms? Or can we design an algorithm that performs consistently in many different applications?" attract the eyes of the researchers.

- **Scalability (Speed)**: As we mentioned at the beginning of the introduction, there would not have a fixed limit on the size of data which we go for analysis. In general, we prefer to choose a suitable clustering method to be as less computational demanding as possible, and meanwhile, to be scalable with the dimension of input data. In another aspect, since the computational workload always affects the speed, scalability and speed are then closed related factors.
- **Similarity Measure**: Similarity measure is usually a necessary term required to be embedded into a clustering method, no matter it carries an explicit or implicit form. It is because the very basic working principle of clustering is to divide a set of objects into groups of "similar" objects. Well known measurement such as Euclidean distance and cosine similarity are widely applied to many real-world applications. The latter one has been most commonly used in the case of high dimensional textual document clustering. However, we realize it could be much more challenge to get an appropriate measure to explore the true underlying structure of data, in a high dimensional space due to the curse of dimensionality. Therefore, in recent decades, more robust and satisfying measures, such as implicit measure in non-negative matrix factorization (NMF) and multi-viewpoint based similarity have also been proposed.
- **Overlap**: In many clustering algorithms, such as *kmeans* or hierarchical clustering, one object is only assigned to one cluster. However, in real cases, content of a document may across over more than one topic; in other word, one object might span multiple topics. For this reason, it is more natural for a clustering algorithm to allow each document to belong to multiple topics with various degrees to reflect the underlying inner structure of the dataset.
- **Simplicity& Availability**: Clustering algorithms are often used as one part of a bigger process in an entire system. It is very important for an algorithm to be easily adapted and capably integrate with various applications. The main reason that those very fundamental clustering algorithms such as *kmeans*, *k* nearest neighbor, are still considered as the top 10 most popular data mining algorithms in the world is because of their generalness and simplicity. On the other aspect, most of the existing algorithms are formulated in a way specific to some particular situations, and hence become difficult to be applicable in other situations.

Overall, we could say that the challenge encountered at developing a good data clustering algorithm is not limited to what we just mentioned above. In fact, there are still many others. However, on the opposite side, it implies that the potential of developing new data clustering techniques is still large. With our research, we hope to develop a few novel clustering approaches that are both effective in quality and efficient in computation, which in other words, is able to shorten the gap between the challenges from the theoretical level and the real usefulness for the practical applications of clustering.

1.3 Contributions

The goal of this study is to develop a few novel semi-supervised clustering approaches and improve the analysis of high dimensional textual data, by making good use of a small amount of prior knowledge which is available to user. On the way of targeting our objective, several contributions have been made and, they are declared briefly as follows:

- Provide a brief but broad review on various general types of text clustering techniques, followed by a more comprehensive and detailed review concentrated on the semi-supervised clustering field. The distinctive features and strength of these methods are discussed, as well as those important remaining issues which need to be aware.
- Propose three novel search-based semi-supervised approaches based on the fuzzy co-clustering framework. Two types of prior knowledge: class labels and pair-wise constraints from the document domain or word domain are directly incorporated into the clustering process and correctly guide the cluster search of the whole dataset softly. A group of optimal co-clusters can be obtained by simultaneously updating the document (object) partitioning membership and words (feature) partitioning / ranking through an iterative algorithm
- Develop a novel semi-supervised clustering framework by using the Multi-Viewpoint based Similarity (MVS) Measure. Two proposed MVS measures and the corresponding novel MVS-based clustering approaches are able to assess the similarity between a pair of documents from multiple appropriate viewpoints with the help of a small percentage of labeled documents or a few pair-wise constraints in the dataset, instead of only the origin in most of the traditional ways. Empirical study through validity tests on the MVS Matrices is conducted to show the effectiveness of the similarity measure; while theoretical analysis on the clustering criterion functions is then presented systematically to verify the new MVS-based clustering are able to make good use of the knowledge for both similarity enhancement and search-guiding purpose during the clustering process.
- Study of prior knowledge-based clustering application to web data is conducted. A case study of on-line sentiment data analysis by applying a proposed dual domain knowledge-

based fuzzy co-clustering algorithm demonstrates the strength of our proposed methods in the specific application area.

- Extensive experimental study is conducted to verify the effectiveness of each of the proposed clustering approaches on a number of benchmarked textual document collections. Better clustering performance in terms of accuracy and time efficiency and other unique strengths are reported and discussed, compared with a few state-of-the-art semi-supervised clustering/ learning approaches.

1.4 Thesis Outline

The rest of this thesis is organized as follows. The literature review of the data clustering, including high dimensional data presentation, similarity measure, representative clustering algorithms, and evaluation metrics is given in Chapter 2. Three novel semi-supervised approaches under the fuzzy co-clustering framework are presented in Chapter 3. Next, in Chapter 4, the semi-supervised MVS clustering framework is reported. The application of the research work is discussed in Chapter 5, showing how the proposed methods can be used in real-life problems. Finally, we conclude the thesis by Chapter 6.

Chapter 2

Research Background

2.1 Overview

Clustering is one of the most important machine learning techniques in various data analysis tasks. It has been widely applied to text mining, gene expression in bio-informatics, spatial database, medical image segmentation and Web application. Due to this, remarkable efforts have been made by researchers from different fields on the development of novel clustering methods. In this chapter, we review the important background knowledge in the field of data clustering. Instead of discussing general issues and providing an exhaustive survey of various clustering approaches, we concentrate more on issues and clustering approaches that are related to high dimensional data, such as textual documents or gene microarray data. The review in this chapter includes the following components: data (document) representation, similarity measure, evaluation metric, clustering algorithms, and some existing potential issues.

2.2 Data Representation

Data representation is the first important step for text-based information retrieval. It directly determines the transformed information and data structure which put into the clustering process later, from the original data collection. Since we decide to use textual documents as the example for high dimensional data illustration, we review and discuss some popular representation models in this section, and the document similarities which are possibly defined based on the presentation model will be covered later.

2.2.1 Vector Space Model

Vector Space Model (VSM) [8] is one of the simplest and most common representations for high dimensional data object such as textual documents, gene microarray data in clustering [8]. In this model, each document is represented as a vector in a space with each term as a dimension. The term could be defined as a single word, n-grams, noun phrase, etc. a weighting scheme is established to measure the importance of each term. Usually without specific declaration, each valid word appear in the whole document collection is served as a

term. Therefore, a term-document matrix $\boldsymbol{D}_{m*n}$ is used to represent a dataset that contains *n* objects (documents) and *m* unique feature (words). To decide the importance of each term, several term-weighting schemes have been explored, including binary term frequency, i.e. the related weight is valued according to whether if it appears in the document frequently and Term Frequency-Inverse Document Frequency (*tf-idf*) [9]. Moreover, to avoid the negative impact from a few words which may appear frequently in the most of the documents but in fact contain little discriminative information, inverse document frequency is multiplied with term frequency together to determine the final weight of each term. The *tf-idf* value for term *j* in document *i* is calculated as:

$$d_{ij} = tf_{ij} \cdot \log \frac{n}{idf_{ij}} \tag{2.1}$$

where tf_{ij} and idf_{ij} represent the frequency of the *j*th word appears in the *i*th document and the number of documents containing the *j*th word, respectively. The effect of using *tf-idf* is that a term *j* occurring frequently in a document *i* but rarely in the rest of the collection carries more discriminative power. Typically, even a moderate size of documents can lead to a large number of distinctive words (dictionary) that make up a very high dimensionality in the vector space model. However, one document typically only contains a tiny portion of words in the whole word collection and this makes the term-document matrix very sparse. This high dimensional sparse matrix can be very noisy and difficult for normal clustering algorithms.

Other than that, VSM treats words as independent entities, completely ignoring the structural information inside documents, such as syntax and meaningful relationship between words or between sentences. Recently, many efforts have been made to find a better way to represent text document. As mentioned, sparsity is a problem of VSM. A document vector has so many unrelated dimensions that may hide its actual meaning. To improve the information contained in the model, meanwhile to reduce the computation requirement, researchers have tried to determine a small number of features and their weights which are used to represent documents by feature reduction or extraction, or make use of semantic relatedness of words, or to find some sort of concepts, instead of words. These kinds of model will be described in the next two sections, respectively. However, such semantic relatedness or concepts is hard to obtain accurately. Despite its simplicity, VSM still offers the best performance until now. Its simplicity facilitates fast computation, at the same time provides sufficient numerical and statistical information. Hence, it is a common model used in most of the clustering algorithms nowadays.

2.2.2 Multi-Word Term (phrase) Representation

Multi-word term is a modified version of the VSM model. A document can be still represented as a vector, but the entities are now groups of words, or noun phrases, instead of single words [10]. In [11] , the author showed how to discover frequent term sets by association rule mining algorithm for text mining. In [12] , some linguistic-based methods are combined with the statistical-based ones to select a small number of words based on both frequency information and the syntactic position in the document. Then, the dimensionality of document vectors is significantly reduced compared to traditional word representation. This key-word selection processing enables a document to be represented with a less number of important words but with little information loss. As in natural language, words are often combined orderly into terms or phrases to express an idea, object or event, the purpose of this model is to keep some information of the dataset from the semantic side, other than the VSM can do. Therefore, additional steps such as natural language processing and lexicon analysis need be carried out. However, while the portion of semantic info might be increased, the quality of the statistical info can be inferior because group of words are then hard to repeat than word alone. Moreover, to identify semantic relationship of words accurately remains as another challenging task nowadays. Perhaps due to this reason that, although this representation sounds naturally more convincing, its experimental results are not always better than single-word VSM [13] .

2.2.3 Knowledge-based Representation with NLP

Recently, the research community related to text and language is increasing its interest and attention to knowledge-based model by incorporating the knowledge in *NLP* concept. In knowledge-based model, documents are no longer represented by their original words, but by explicit concepts or semantic roles which have already been pre-defined by a separate process, e.g. using *NLP* and domain knowledge, and stored in a pool of knowledge, or often referred to an ontology system. For example, a parser [14, 15] is applied to the documents to pre-define the semantic roles of the frames or arguments in sentence level. Concepts in an ontology system are categorized into specific domains, such as artificial intelligence, bio-informatics and so on. The document processing step then uses the help of this knowledge database to replace words in documents by their related concepts. Hence, documents are no longer represented as vectors of words, but vectors of concepts instead.

Wordnet [16] , a lexical database of English language, can be considered as an example of a simple ontology system. Nouns, verbs, adjectives and other types of words are grouped together into sets to describe their semantic and lexical relations in Wordnet. In [17], a structured document vector space with low dimensionality was created with the help of the Wordnet[1] ontology, hence allowing usual clustering algorithms to perform better as such

curse of dimensionality is reduced. One important property of ontology systems is the existence of relationship among entities. When documents are represented by simple word counts, distance measure (e.g. Euclidean) or cosine similarity can be used to determine relatedness among them. However, when they are represented by concepts, a specific new similarity measure need be explored in order to correctly reflect the relationship between these concepts. In other words the effectiveness of knowledge-based model is highly dependent on how you identify the semantic concepts and obtain the relationship among them.

2.2.4 Representation with Order Information

The phrase based model (PBM) [18, 19] uses phrases or sequences of words as units, with which word order information is reserved. The phrase in this context means a sequence of words, and doesn't imply any grammatical structure. Consider two phrases “the dog chased a cat" and “the cat chased a dog" [20] shown in Figure 2.1. Although their vector representations are identical, the meanings of them are obviously different. A phrase-based model takes word order information into account to improve the representation of a document, hence to improve the accuracy. Another motivation of a phrase-based model is that phrases can be used to describe or label a document cluster while separated terms (words) are used to interpret a cluster in the vector space model. It is generally agreed that phrases are more descriptive than individual words. Two examples of phrase-based models are reviewed below.

Suffix Tree Clustering (STC)

This method is proposed originally to clustering snippets returned by a search engine. In STC[21], valid phrases of each document are indexed into the suffix tree. After all documents are encoded into the suffix tree, each of base clusters can be identified with scores calculated as a function of documents it contains and the number of words making up its phrase. Finally, base-cluster merging is taken based on a high degree of overlap between their documents sets. The construction of suffix tree is linear time with the size of documents and each document is allowed to belong to multiple groups. However, later studies found that order information itself is not sufficient to produce good clusters because a clustering may not include all document in the corpus and it depends highly on document frequency which leads to larger cluster [22]. This raises the question that whether improvement can be seen if both word order information and frequency are used to measure the document similarity. In [22], better results are shown with both graph-based clustering and hierarchical clustering under a hybrid similarity measure calculated with a weighted combination of phrase matches and cosine similarity.

Document Index Graph (DIG)

DIG [23] is similar in spirit to the STC model in that it encodes word order information and defines similarity based on matches in word order. In DIG, each word is a vertex in a directed multi-graph. The directed edges of the graph represent the sequence of the words. The main difference between STC and DIG is that DIG stores words explicitly and maintains frequency information [23]. Another important point should be noted is that unlike STC, DIG is merely a document index model but not a clustering algorithm [20]. Based on the DIG model, a similarity matrix recording pairwise document similarities can be obtained by computing the overlapping sub-graphs, and used in any similarity-based clustering algorithms. An important point observed by both [23] and [22] is that only word order information is insufficient to measure document similarity, while word order information together with term frequency can provide a better document similarity measure to improve clustering accuracy. However, more computation time is needed to collect two kinds of information.

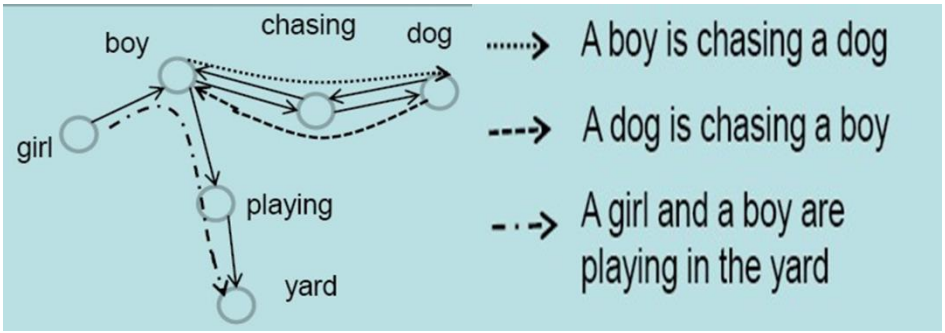

Figure 2.1: A simple sentence of word order kept

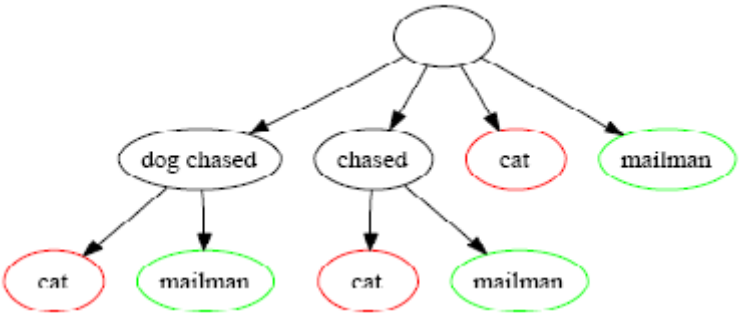

Figure 2.2: The suffix tree resulting from “dog chased cat” (red) and “dog chased mailman” (green)

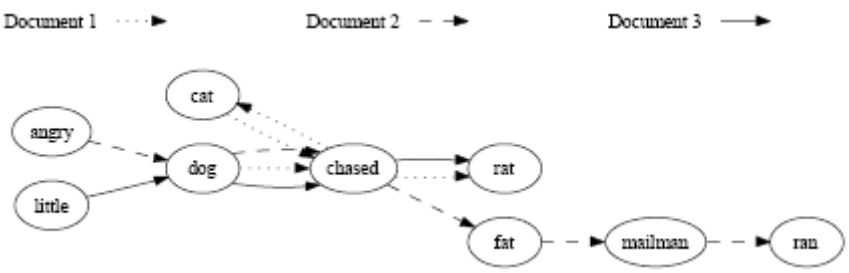

Figure 2.3: The example in Document Index Graph

2.3 Similarity Measure

The proximity among the objects including similarity and dissimilarity measure are one of the most important impact factors for the success of a clustering method. It is usually served as the basis of validity for building the criterion function in a clustering task. In general, there are plenty of (dis)similarity measures, and it is common that the same (dis)similarity measure with different algorithms or different (dis)similarity measures with the same algorithm, can lead to different results for the same dataset. Document similarity is usually carefully selected based on the document representation model, like vector space model (VSM) or phrase-based model (PBM). Sometimes it can be calculated directly without representing the documents into a specific model. Here we only review several widely used document similarity measures as well as some recently developed ones.

When VSM is applied for high dimensional data, the vector space usually has a very high dimensionality, for which some measures, like Euclidean distance are no more effective. One widely used similarity measure for documents is the cosine measure given in Eq.(2.2)

$$Sim\left(d_i, d_j\right) = \cos\left(d_i, d_j\right) = d_i^t d_j \quad , \tag{2.2}$$

where d_i, d_j are two normalized vectors representing two documents i and j, respectively.

There are also other measures such as Jaccard measure [24] may be used to measure document similarity based on "bag of words" model, i.e. two documents are regarded to be more similar if they share more common words. Eq.(2.3) combines the feature of cosine similarity and Euclidean distance, so the magnitude and the direction of the document vectors are both taken into accounted in.

$$S_{Jaccard} = \frac{d_i \cap d_j}{d_i \cup d_j} = \frac{\left(d_i \cdot d_j\right)}{\left\|d_i\right\|^2 + \left\|d_j\right\|^2 - \left(d_i \cdot d_j\right)} \quad , \tag{2.3}$$

Other popular measures like Pearson Correlation Coefficient appeared in [25], while cosine similarity keeps value non-negative, it allows the value of similarity falls in the range from +1 to -1. In [26], Pelillo argued that the symmetry and non-negativity assumption of the similarity measure in text clustering was actually a limitation. Moreover, Kullback-Leibler

divergence [27] was a good example of non-symmetric measure, which is widely applied measure for evaluating the difference between to probability distributions in information theory-based clustering.

The above models have an obvious limitation that is only a portion of the statistical information of the dataset is kept in the transformed format; other useful information relevant to the semantic meaning and the NLP concept such as sequence information is lost. As we mentioned in last section, the suffix tree model (STX) and directed index graph model (DIG) both reserve the word order information but not frequency information. Therefore, some recently proposed similarity measures for documents attempt to combine both the frequency and the order information [18].

Significant improvements in clustering results are shown in these studies when these hybrid similarities are used. For example, Based on STM, a hybrid similarity using linear combination of STM-based similarity φ_{ST} and *tf-idf* based cosine similarity φ_{cos} defined in Eq. (2.2) with coefficient λ is given in [19] as below:

$$\varphi_{HYB1} = \lambda \cdot \varphi_{ST} + (1-\lambda) \cdot \varphi_{\cos} , \tag{2.4}$$

$$\text{With } \varphi_{ST} = \frac{\left| E^i \cap E^j \right|}{\left| E^i \cup E^j \right|} , \tag{2.5}$$

where E_i and E_j are the edges in the suffix tree related to two selected documents respectively. This is one of the explicit ways to combine the frequency and order information together. Another way to incorporate the frequency into the STM-based similarity is given in . Each document is first encoded into STM so the words order information can be reserved, and then the resulting nodes in the suffix tree are mapped into VSM to further count to retrieve the frequency information. The nodes of the constructed suffix tree are then treated as terms in VSM, so documents are represented as shorter vectors with these terms as dimensions. The same *tf-idf* weighting scheme is again applied to weight the terms before cosine similarity is used to compute the proximities values. Similar hybrid similarity measure based on the DIG model has also been proposed in [23]. Several factors of the matched phrases are considered, including significance level, length, and frequency.

A consistent conclusion made in these studies is that better clustering performance can be achieved by using a more informative hybrid similarity which considers both term frequency and order, compared with using a similarity which only considers one type of information. “It has also been shown that better clustering accuracy can be achieved based on semantic meanings of documents with knowledge from external ontology, e.g., clustering result is reported to be significantly improved in [24] when concept information and category information is taken into account together with the content information to measure the document similarity.”

Other than text-based document representation and similarity, it is also important to consider the link information for measuring the similarity of Web documents. In [28], the results returned by a search engine are represented with features derived from shared links. In [29], a Webpage is described with keywords extracted from its pages that point to that page based the incoming links. The similarities of pairs of pages are then calculated with WordNet-based similarity measure. In [30], the similarity between two web documents is derived based on three types of information, i.e., the textual information, hyperlink structure and co-citation information.

2.4 Existing Clustering Approaches for High Dimensional Textual Data

After the data is represented into a specific mathematic form, a clustering algorithm is applied as the engine to generate the final clusters. In this section, we give a very board literature review on different existing clustering approaches for high dimensional data and briefly discuss on the strength, weakness and applicable area in real-world one by one. Some survey papers in the area of fuzzy clustering, spectral clustering, sub-space clustering model based clustering can be found in [20, 31-33] .

2.4.1 Hierarchical Clustering

A hierarchical method usually creates a hierarchical decomposed structure for a given dataset called dendrogram [34]. From the dendrogram, partitioning with different granularity can be selected by cutting it from different levels, the lower level the larger number of clusters with smaller sizes. According to the way of building the dendrogram, hierarchical clustering can be further grouped into bottom-up [34] approach and top-down divisive approach [35]. Agglomerative hierarchical clustering starts with every single object as a cluster and then gradually merges clusters until all objects are in one cluster; while divisive hierarchical clustering processes in an opposite way which starts with one cluster containing all objects in the dataset and then recursively split it into smaller ones until all of the clusters only contain one object. “Linkage Metric” is the tool to assess the proximity between individual objects, and then the proximity measure of two subsets of the objects can be generalized from that to decide where the merging or splitting takes place in the dataset. Some of the popular inter-cluster linkage metrics include single link, complete link and average link [36]. It is noted that the linkage metric significantly affects the performance of hierarchical clustering methods since it does reflect the definition of closeness of two clusters.

As an important data mining technique, hierarchical clustering has been widely used for organizing the textual documents into a hierarchal structure in many real applications for many years. One of the main advantages is that it does not require the data to be represented

as vectors since only pair-wise similarities are measured. However, some of the disadvantages can be also summarized: firstly the time complexity is typically high at $(n^2 log^n)$, which leads to high computational cost for large datasets. Secondly, the termination criterion can be vague. Thirdly, most hierarchical clustering methods do not revisit clusters once they are constructed. The experimental results given in some of the studies [36] show that, compared with hierarchical clustering, partitioning clustering is usually more favorable for large high dimensional data (document) clustering due to high accuracy and low computational cost.

2.4.2 Partitioning Clustering

Unlike hierarchical clustering which produces a series of nested clusters, partitioning clustering generates a flat partitioning of the dataset. *kmeans* [37] , one of top 10 algorithms not only in clustering but the whole data mining field is indeed the most appropriate example for this section. Given a dataset *S*, which consists of a set of *n* objects, a crisp *kmeans* clustering is to partition the whole set these *n* objects into *k* disjoint sub-sets, where each subset correspond to a cluster, each object is only grouped into one cluster, and each cluster contains at least one object. i.e., $S = S_1 \cup S_2, ... \cup S_k, \forall r, S_r \neq \varnothing, and \forall r \neq f$, and $n_r = |S_r|$.

It is noted that a good clustering result makes the objects in the same clusters are similar to each other, while the objects in two different clusters are very dissimilar. We have already discussed the importance of choosing a suitable similarity measure in Section 2.3. The idea of searching for the optimal partitioning is achieved by minimizing the sum of squared error (SSE) objective function, between each object and its corresponding center vector, based on a pre-defined similarity measure.

When Euclidean-distance is used, the function is expressed as below:

$$J_{kmeans} = \sum_{r=1}^{k} \sum_{d_i \in S_r} \left\| d_i - c_r \right\|^2 \quad , \tag{2.6}$$

where c_r is the centroid of cluster r calculated as the mean of all the objects in that cluster, i.e:

$$c_r = \frac{1}{n_r} \sum_{d_i \in S_r} d_i \ . \tag{2.7}$$

As discussed in Section 2.3, similarity measure is another important factor in clustering. Due to the "Curse of Dimensionality"[38] , Euclidean- norm used in Eq (2.6) may cause *kmeans* performing poorly on text mining as the textual data is usually represented in in the high dimensional and sparse space. Various extended versions of *kmeans* had been developed to overcome this problem. Spherical *kmeans* algorithm was proposed in [38] by using cosine similarity instead of Euclidean distance to handle high dimensional clustering. That is because in a very high dimensional space, direction is more important than the magnitude (distance).

Therefore, instead of minimizing Eq.(2.6), we then try to maximize the objective function shown as below:

$$J_{kmeans} = \sum_{r=1}^{k} \sum_{d_i \in S_r} d_i \cdot c_r \tag{2.8}$$

It is noted all the vectors d_i and c_r which named by the authors, are normalized to unit length when cosine similarity is applied.

The *kmeans* algorithm can be described as following:

1. *Initialization*: All the document vectors are normalized to unit length, and randomly partitioned into *k* groups. Given a group of vectors, it can be proven that the group's mean itself has the maximum sum of cosine similarities to all the elements in the group. Hence, centroid vector of cluster *r* is determined by Eq.(2.7).
2. *Re-assignment*: Calculate the distances(similarities) of each object vector to all the centroid vectors, then re-locate it to the cluster with the closest centroid, i.e.:

   $$d_i \in S_r \Leftrightarrow d_i^T c_r \geq d_i^T c_t, \forall t \neq r, 1 \leq t \leq k$$

3. *Re-defining centroid vector*: the *k* centroid vectors will be re-calculated according to Eq.(2.7) based on the new partitions in step 2
4. Step 2 and 3 are repeated iteratively until no further change on the partitions can be made.

Another important variant of *kmeans* is the bisecting *kmeans* [39], which has also shown the strength on textual document clustering. It is similar to divisive hierarchical clustering, starting from the entire dataset in one cluster, and then subsequently dividing into the number of *k* clusters. In kernel kmeans [40], clusters with nonlinearly separated boundaries can be detected. The objects are firstly mapped into a high or infinite dimensional space, and the clustering process is carried out in that mapped space. In on-line or incremental *kmeans* [41], some modified objective functions need to be developed together with a modified algorithm in order to allow the objects come one by one, and the update of a centroid vector only take place when that particular cluster got new member to assign in. More recently, Feature Weighting *kmeans* [42], which can be categorized into the subspace clustering for high dimensional data is proposed. It allows different clusters are formed under different sub-groups of dimensional space. In other words, an approximate dimension reduction process is conducted for *kmeans* to deal with the "curse of the dimension". A good survey on sub-space clustering for high-dimensional data is reported in [32].

Besides those great advantages to make *kmeans* remain one of the top 10 algorithms, some drawbacks of *kmeans* such as the sensitivity to initialization and local minima. Convergence have also been discussed in [38].

2.4.3 Fuzzy (Soft) Clustering

The cluster assignment in a hard or crisp clustering approach like *kmeans* only allows taking binary values to represent the belonging of each object, i.e. one object can be assigned in only one cluster. Principles from the fuzzy set theory are applied in fuzzy clustering to to allow each object to belong more than one cluster at the same time with different degrees. This degree of belonging is called fuzzy membership, and denoted as u_{ri} in our thesis. It takes a continuous value between 0 and 1, and requires the sum of membership of each object to all the clusters is strictly equal to 1. This enables fuzzy clustering to capture clustering structure realistically, as most real-world categorizations naturally contain some overlapping content (e.g. physics are cannot exist without math). For this reason, fuzzy clustering is often considered to be more informative to reflect the nature of datasets than its crisp counterpart.

It is noted that the term fuzzy clustering is sometimes mixed with the term "soft clustering" [43-47] in the data mining field. The research efforts made on developing soft clustering algorithms is very rich in literature. A few good survey papers have been published on how fuzzy clustering approaches works on data mining, pattern recognition, informational retrieval and machine learning fields [48, 49]. Some simple correlation analysis for different fuzzy (soft) clustering approaches can be found in [50].

Fuzzy clustering algorithm can also work based on many different effective distance functions, such as Euclidean distance, cosine similarity, Bregman distance [51]. For document clustering, existing works have demonstrated that fuzzy clustering approaches generally outperform the binary counterparts based on original textual data [52] or relational data [53].

Now we would like to introduce two fundamental fuzzy clustering algorithms in details.

Fuzzy c-means (FCM) [54] is the most well-known fuzzy clustering algorithm. The goal of FCM is to partition objects into a set of clusters based on their distances to the cluster centroids. At the same time FCM also tries to capture the overlaps among the different clusters by representing the object-to-cluster assignments by fuzzy memberships which are produced by a fuzzifier α. The higher the fuzzy membership, the more the object belongs to the cluster. The objective function to be minimized and two updating rules are shown as following:

$$J_{FCM} = \sum_{i=1}^{n}\sum_{r=1}^{k} u_{ri}^{\alpha} Dist\left(d_i, c_r\right) , \tag{2.9}$$

$$u_{ri} = \left(\sum_{l=1}^{k} \left(\frac{Dist_{ri}}{Dist_{li}} \right)^{1/(\alpha-1)} \right)^{-1} , \tag{2.10}$$

$$c_r = \frac{\sum_{i=1}^{n} u_{ri}^{\alpha} d_i}{\sum_{i=1}^{n} u_{ik}^{\alpha}} \quad . \tag{2.11}$$

Let $\boldsymbol{U}=[u_{ri}]$ be a $k \times n$ fuzzy membership matrix, and $\boldsymbol{C}=[c_1,c_2,\ldots c_k]$ be the cluster centroids matrix. Staring from a set of random initialized centroids within $\boldsymbol{C}$, $\boldsymbol{U}$ and $\boldsymbol{C}$ are iteratively updated by Eq.(2.10) and (2.11). This iteration will repeat until $\max_{\text{ri}} \left\{ \left| u_{ri}^{(\tau+1)} - u_{ri}^{(\tau)} \right| \right\} < \varepsilon$, or $\max_{r} \left\{ \left| c_r^{(\tau+1)} - c^{(\tau)}_{r} \right| \right\} < \varepsilon$ where ε is a user-defined termination criteria, whereas τ is the iteration steps. This procedure converges to a local minimum or a saddle point of *J*.

Instead of using the fuzzifier α, a fuzzy partitioning of a given dataset can be also produced with the help of technique of regularization. The objective function to adaptive such changes are then re-formulated by integrating an additional regularization term or penalty term. Some of the common regularization terms can be entropy regularization [55] and quadratic regularization [56] .

One disadvantage of FCM is that it is weak on outliers and noise detection. Another soft clustering method: Possibilistic C-Means (PCM) [57] was proposed to overcome this problem. Basically, PCM relaxes the constraint which the fuzzy membership of an object to all the clusters must sum to 1. Instead, just a non-negative degree between 0 and 1 will do. However, another drawback in PCM is that it tends to produce co-incidence clusters. An improved version of fuzzy-based clustering, called Possibilistic Fuzzy C-Means (PFCM), was introduced in [58]. The authors combine two techniques into one in order to take advantage of each, and solve the problems of both.

Nevertheless, they are still far from being efficient for document categorization. In [32], the author discussed about what can fuzzy cluster analysis contribute to clustering of high dimensional data. The improved version of FCM and PFCM called *Hypersperical* Fuzzy C-means (HFCM) [52] and *Hypersperical* Possibilistic Fuzzy C-Means (H-PFCM) [59] are developed accordingly. Similar to the changes made from original *kmeans* to spherical *kmeans*, they are reformulated by applying the cosine dissimilarity to the object-centroid measurement instead of the Euclidean distance for clustering normalized document vectors.

The modified objective function and updating rules for HFCM are written as following:

$$J_{HFCM} = \sum_{i=1}^{n} \sum_{r=1}^{k} u_{ri}^{\alpha} \left(1 - \sum_{j=1}^{m} d_{ij} \cdot c_{rj} \right) , \tag{2.12}$$

$$u_{ri} = \left(\sum_{l=1}^{k} \left(\frac{1 - \sum_{j=1}^{m} d_{ij} \cdot c_{rj}}{1 - \sum_{j=1}^{m} d_{lj} \cdot c_{rj}} \right)^{1/(\alpha-1)} \right)^{-1} , \tag{2.13}$$

$$c_r = \sum_{i=1}^{n} u_{ri}^{\alpha} \cdot d_i \left[\sum_{j=1}^{m} \left(\sum_{i=1}^{n} u_{ri}^{\alpha} \cdot d_{ij} \right)^2 \right]^{-1/2} . \tag{2.14}$$

One of the sub-categories in the fuzzy clustering family, called fuzzy co-clustering [60], is specifically designed for high-dimensional data clustering such as text, gene microarrays; will be discussed in details later.

2.4.4 Spectral Clustering

Spectral clustering, also be realized as graph-based partitioning methods in most of the cases uses an affinity matrix calculated from object data as the input data. From the an affinity matrix ***A***, a weighted graph $G(V;E,A)$ is constructed with each object as a vertex , each edge as the association between two vertices, and the pair wise similarity or dissimilarity a_{ij} as the edge weight. If the edges of a graph have no weight, the graph is considered as un-weighted. In such case, the degree of a node is counted by the number of its adjacent edges. In an undirected weighted graph, the degree of a node can also be defined as the sum of the weights of its adjacent edges. Once the graph has been constructed based on the matrix ***A***, an algorithm which iteratively performs the Eigen-decomposition of ***A*** is carried out, in order to find the optimal partitioning/ cutting of the graph, in other words, the optimal clusters. Overall, the algorithms can be classified according to two approaches due to the survey [61]: recursive two-way spectral clustering algorithms [62] and direct *k*-way spectral clustering algorithms[63] . The direct *k*-way spectral clustering algorithms may have become the more popular one in the last decades The difference in constructing the affinity matrix and selecting eigenvectors result to different approaches, such as ratio cut [64] , normalized cut [40], and min-max cut [65].

Suppose V_r denote a subset of vertex of V, corresponding to cluster r, then V_r is exactly equivalent to S_r we defined before. Ratio cut (RC) aims to minimize the inter-cluster similarity normalized by cluster size:

$$J_{RC} = \sum_{r=1}^{k} \frac{Sim\left(S_r, S \setminus S_r\right)}{n_r} . \tag{2.15}$$

The Normalized Cut (NC) aims to minimize the inter-cluster similarity, but normalize it with the measure of compactness from each cluster to the entire dataset.

$$J_{NC} = \sum_{r=1}^{k} \frac{Sim(S_r, S \setminus S_r)}{Sim(S_r, S)} \quad . \tag{2.16}$$

The Min-Max Cut (MMC) and is more informative as it simultaneously maximizes the intra-cluster similarity and minimizes the inter-cluster similarity at the same time:

$$J_{MMC} = \sum_{r=1}^{k} \frac{Sim(S_r, S \setminus S_r)}{Sim(S_r, S_r)} \quad . \tag{2.17}$$

The main disadvantage of graph-based spectral clustering is that the pairwise similarity of the vertices has to be explicitly defined, and the affinity matrix has to be pre-computed, so the memory cost and computational workload could be very high when handling some large and high-dimensional data. A famous spectral co-clustering algorithm called Bipartite Spectral Graph Partitioning (BSGP) [66] will be given in more details in the next section.

2.4.5 Clustering based on NMF

Non-negative matrix factorization (NMF) [67] is another important technique that has been shown not only significantly reducing the size of document vectors, but also improving the quality of resulting clusters. Different from fuzzy co-clustering framework, document clustering using NMF treat each cluster of a dataset as the embodiment of a coherent concept, and try to close the linear combination the product of the degree of word *j* and document *i* over all concepts to the weighted term frequency x_{ij} . More precisely, given a document corpus of *k* topics, with *m* words, *n* documents, NMF aims to minimize the objective function:

$$J = \frac{1}{2}\left\| X - FG^T \right\|_F^2 \text{ s.t. } F \in R_+^{m \times k}, G \in R_+^{n \times k} \quad . \tag{2.12}$$

Each element f_{jr} of matrix *F* represents the degree to term *j* belongs to cluster *r*, while each element g_{ir} of matrix *G* indicates to which degree object *i* is associated with cluster *r*. Original NMF is not required to be orthogonal, therefore, similar with fuzzy co-clustering, soft degrees is allowed to be assigned in different clusters as documents are allowed to span multiple topics which is often the truth in real world application. After [67] is proposed, a number of its variants [68, 69] have been proposed in the past ten years, such as convex [70] and semi-NMF [71]. A study on various NMFs is reported in [72]. More details can be found in [73], a well written survey chapter of performing clustering through NMF. ONMTF [68] is the interesting one among them. A tri-factorization is applied on the input term-document matrix X , and give dual orthogonal constraints on document and word domain, rather than usually only two matrices are factorized. In this version, the restriction on equal number of document and word cluster is released by a scale matrix factor *S* which provides a condensed view of *X*. It also provides the capability similar to the co-clustering. However, the orthogoality conditions listed in Eq. (2.13) enforce each row of *F* and *G* has only one nonzero

element, in other word, ambiguity is avoided. However, this may lead to the failure of the minimization of J [74]. For text data, in fact documents might span into multiple topics, so in this case the underlying semantic variables in F and G will not be orthogonal [20]. Moreover, recently some more complicated clustering approaches [75] have also been developed based non-negative matrix factorization . It cooperates with the co-clustering concepts and some available prior knowledge to have multiple types of clusters for handling high dimensional data with a heterogeneous data structure. More details will be given in Section 2.5.

$$F^T F = I$$
$$G^T G = I \tag{2.13}$$

The main disadvantage with NMF is that it relies on random initialization. As a result, the same data might produce very different results across runs. This problem brought from random initialization is not only for clustering high dimensional text data, but also encountered in low dimensional data. As most clustering process is formulated into an optimization problem, given a "bad" initialization, they may converge to a not-so-good local optimum, hence, leading to bad clustering results. Therefore, in some of the NMF algorithms, the centroid produced from *kmeans* algorithm is used as seeds for NMF [20, 74]. However, some experiments show that this pre-processing task may reduce the chance of NMF fall into a bad result caused by random initialization, but not completely avoid it yet.

2.4.6 Model-based Clustering

Most of the clustering techniques we review above can be categorized to discriminative methods, as the clustering is carried out by optimizing a particular objective function which is formulated based on certain pre-defined explicit or implicit object similarity measure. On the other hand, another type of clustering, called generative methods, or model-based methods do not require the similarity for iteratively procedure, but tries to fit the data into some probabilistic distribution models for object assignment. One of the main advantages of the probabilistic approaches is that the partition can be interpreted from a statistical point of view. The most widely used probabilistic generative model-based clustering is the Gaussian mixture models (GMMs). "For the similar reasons as spherical *kmeans*, a mixture of vonMises-Fisher models approach is proposed in [76] for text clustering, where the vonMises-Fisher distribution could be seen as a spherical analog of Gaussian ".

It is noted the model-based clustering usually interfaces with maximum likelihood function Expectation-Maximization (EM) algorithm [77] to estimate the parameters. However, in a high dimensional feature space, most of the classic model-based methods may be disappointed due to the over-parameterized problem caused by the curse of dimensionality. Therefore, dimension reduction like PCA or regularization is frequently used before the real

clustering process is carried. This problem may be also addressed with the help of some prior knowledge in terms of constraints available to the user. Other than that, model-based subspace clustering techniques try to exploit the "empty space" to ease the discrimination between groups of objects. As reported by Charles Bouveyron [31], they are mostly related to the factor analysis model which the observation space is assumed to be linked to a latent space. A comparable study on the generative model-based document clustering is presented in [78].

Last but not least, topic modeling approaches may be another good choice for text mining. Popular topic modeling approaches include Latent Dirichlet allocation (LDA) [79] and probabilistic latent semantic indexing (PLSI) [80]. In LDA model, documents are assumed as random mixtures over latent topics, and each topic is a probability of distribution over a group of words. PLSA is similar to LDA, except the topic distribution of the latter one is assumed to have a Dirichlet prior, which leads to more reasonable mixtures of topics in a document.

LDA assumes the following generative process for each document d_i in the corpus ***D***.

Giving:

n is the total number of documents,

m is the total number of words,

k is the total number of topics,

α is the parameter of the Dirichlet prior on the per-document topic distributions,

β is the parameter of the Dirichlet prior on the per-topic word distribution,

θ_i is the topic distribution for document *i*,

ϕ_r is the word distribution for topic *r*,

z_{ij} is the topic for the *j*th word in documents *i*, and w_{ij} is the specific word.

LDA assumes the following generative process for a corpus ***D*** consisting of *n* documents each of length m_i:

1. Choose $\theta_i \sim Dir(\alpha)$ where $Dir(\alpha)$ is the Dirichlet Distribution for α
2. Choose ϕ_r
3. For each of the words *j* in document *i*,:
   (a) Choose a topic $z_{i,j}$~ Multinominal(θ_i)
   (b) Choose a word w_{ij} ~ Multinominal(ϕ_r)

Bayesian Inference [81], such as Bayes approximation of the posterior distribution is commonly used to learn the various distributions listed above. Other alternative solutions include expectation propagation and Gibbs sampling. In other words, the LDA model is essentially the Bayesian version of PLSA model. The Bayesian formulation tends to perform better on small datasets because Bayesian methods can avoid overfitting the data. The LDA

model is also easily extended. For example, Hierarchical LDA [28] allows the topics to be joined together in a hierarchy by using the nested Chinese restaurant process.

For web document clustering, recently proposed models-based approaches such as [82], establish topic models of documents by utilizing additional useful information from the links among the documents. Meanwhile, many model-based clustering approaches have been also extended to its semi-supervised version [83, 84] with labelling.

2.4.7 Co-Clustering

Co-clustering is a very popular clustering technique which has unique strength on high dimensional data analysis, especially for textual data. In the Vector Space Model, it is recognized that the similarity between objects is calculated by their feature distributions, while the similarity between features can be measured by their occurrences (frequency) on the objects. Motivated by the duality between objects and features, co-clustering approaches, [66, 85-87], also called as bi-clustering in bioinformatics have been a popular direction with applications across various domains. Especially for document co-clustering, the underlying assumption is that words which co-occur together in documents tend to be associated with similar concepts. Therefore, simultaneously grouping of similar words and grouping of similar documents are equally important. Given a term-document occurrence matrix, a standard clustering approach only clusters columns corresponding to documents as shown in Figure 2.4(a), while a co-clustering method clusters both rows and columns to generate sub-matrixes as co-clusters as shown in Figure 2.4(b). The word cluster can be used to interpret or describe the related document cluster hence makes it easier to be understood. Meanwhile, different clusters of document have different subsets of words as valid dimensions which are used to update the document clusters.

The ability of handling high dimensional data may become weak when the number of cluster increases. This is because the possibility of using random guess or (since *kmeans* is also sensitive to its initialization) to get a good initialization (initial points are chosen near enough to the global solution) decrease as the number of clusters becomes large and it is easier to get stuck at local optimums before reaching the global solution. In these cases, simply repeating the number of trials fails to bring substantial improvement.

$$\left[\begin{array}{cc|cc|cc} 1 & 1 & 0 & 0.1 & 0 & 0 \\ 0.9 & 0.9 & 0 & 0 & 0 & 0.1 \\ 0.1 & 0.1 & 0.8 & 0.9 & 0 & 0 \\ 0 & 0 & 0.7 & 0.8 & 0.1 & 0 \\ 0.1 & 0 & 0 & 0 & 0.9 & 0.8 \end{array}\right] \qquad \left[\begin{array}{c|ccc|cc} 1 & 0.8 & 0 & 0.1 & 0 & 0 \\ 0.8 & 0.8 & 0 & 0.2 & 0 & 0.1 \\ \hline 0.1 & 0.2 & 0.8 & 0.9 & 0 & 0 \\ 0.1 & 0 & 0.7 & 0.8 & 0.1 & 0 \\ \hline 0 & 0.1 & 0 & 0.1 & 0.9 & 0.8 \end{array}\right]$$

(a) Standard Clustering (b) Co-clustering

Figure 2.4: The comparison between standard clustering and co-clustering

Co-clustering approaches can be also classified into several sub-types. Many of them are developed based on other well-known clustering modules. They include: spectral co-clustering [66], information-theoretical co-clustering (ITCC) [86], , fuzzy co-clustering [70], co-clustering on manifolds [88], hierarchical co-clustering [89], subspace co-clustering [90] , NMF-based co-clustering [68, 91]and model-based co-clustering [78, 92]. Some recent studies try to combine the strengths of two or more than two types of co-clustering together, such as [93], a Bayesian overlapping subspace co-clustering. Sara C. Madeira gave a good survey on co-clustering for biological data analysis [94] . As for text clustering, it is suggested in [95] that, BSGP and ITCC are two suitable examples. In the following paragraphs, we provided more details on certain co-clustering approaches which are most relevant to our own research work.

Information Theory-based approach

In this kind of approaches, important information measures or concepts like mutual information, *Kullback-Leibler* (KL) divergence is explored. For example, an two-stage algorithm was proposed using "information bottle neck" for documents [96]. In the first stage, words are found to maximally preserve the information on the documents, then in the second stage, documents are clustered based on the word clusters generated in the previous stage. In 2003, Dhillon et al. proposed an information-theoretic co-clustering (ITCC) [86] which maximizes the mutual information of a word-document probability-distribution matrix by simultaneously intertwining both the rows and columns at all stages. During the optimization process, word clusters and document clusters are literately dynamic updated to better qualities. These recursively mutual updating finally converges to the optimal co-clusters. As one of the earliest co-clustering approaches, ITCC shows that co-clustering is more effective than a plain clustering of just documents on highly sparse word-document matrices. It is also a good basic model that both the algorithm and main theorem can be easily extended to co-cluster multi-dimensional joint distributions. Moreover, the numbers of document and word clusters are supposed to be pre-specified and can be different. However, since the problem formulation is information-theoretic, we hope that an information-theoretic regularization procedure like MDL may allow the user to select the number of clusters in a data-driven fashion. Furthermore, ITCC also remains as a "hard" co-clustering approach which had not generalized to an abstract multivariate clustering setting that would be applicable when more complex interactions are present, as well as BSGP [66, 86].

spectral co-clustering

Co-clustering based on spectral graph partitioning [40, 64, 97] aims to find an optimal cut in a constructed bipartite graph which the vertices represent the document/word and the weighted edges represent the similarity between two documents/words. In BSGP [66], bipartition of words and documents are obtained as the left and right singular vectors

corresponding to the second largest singular value of the normalized co-occurrence matrix. This idea is also been extended in [98] to handle heterogeneous data represented in a rational data star-network. In [99], Derek gives an extended spectral co-clustering algorithm for cutting dynamic bipartite graphs. In [100], Laurence et al. try to adapt spectral co-clustering to document and terms using latent sematic analysis.

Fuzzy co-clustering

As we mentioned in Section 2.4.3, it is common that a single document may span multiple topics. Combing the fuzzy theory concept with the co-clustering techniques is a good idea to hold both the strength on handling the overlaps among a set of documents and the curse of dimensionality issue. In fuzzy co-clustering framework, the constraints applied on document and word membership distributions and the way to generate the fuzzy co-clusters becomes the key factor to distinguish different algorithms. The document cluster and word cluster are usually one-to-one corresponded. In literature, Fuzzy Clustering for Categorical Multivariate Data (FCCM) [60] and Fuzzy Co-clustering of Document and Keywords (Fuzzy CoDoK) [85] are two important co-clustering algorithms which share the same principle to simultaneously performed document partition and word ranking in co-clusters.

Like the fuzzy c-means algorithm, the fuzzy co-clustering variant optimizes a fuzzy objective function. The important difference is the notion of aggregation, in which the algorithm optimizes for co-clusters of documents and words rather than clusters of documents alone. The aggregation term can be written as:

$$\sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri} v_{rj} d_{ij}, \tag{2.18}$$

where u_{ri} denotes the fuzzy partitioning membership of document i in cluster r, while v_{rj} denotes the ranking score of words j in cluster r. With the common aggregation term, different fuzzy co-clustering approaches may apply different regularization terms in their co-responding objective functions to produce a fuzzy partitioning of a given dataset, such as Gini Index using the Fuzzy Codok and entropy regularization using in FCCM. Last but not least, the co-responding constraints applied on u_{ri} and v_{rj} need to be clearly specified, together with the objective function. They are essential for the updating rules derivation for the two updated memberships. The constraints using in Fuzzy Codok on the document partitioning membership and word ranking membership are shown in Eq. (2.19) and (2.20) as an example.

$$\sum_{r=1}^{k} u_{ri} = 1 \ , \ u_{ri} \in [0,1] \text{ for } i = 1,\ldots.n, \tag{2.19}$$

$$\sum_{j=1}^{m} v_{rj} = 1 \ , \ v_{rj} \in [0,1] \text{ for } r = 1,\ldots.k, \tag{2.20}$$

Compared with FSKWIC [101], FCCM is able to handle categorical features. On another aspect, Fuzzy CoDoK is more suitable for large text corpus than FCCM because FCCM may suffer from computational overflow when the value of *n* and *m* is large. Recently, beside the partitioning-ranking approaches, a fuzzy co-clustering algorithms based on dual-partitioning approach [55] is developed to solve other problems like datasets with overlapping feature clusters, the unnatural representation of some word clusters, unrealistically small work ranking membership values etc.

Similar to the FCM, we should be aware about the common weakness for all fuzzy-based clustering algorithms, which is the ability of detecting outliers, cannot be avoided when we explore the fuzzy co-clustering field. Meanwhile, although the performance of fuzzy co-clustering is not heavily relied on initialization, the optimal parameter setting of the fuzzifiers is left for manually tuning. Sometimes it is not trivial.

Why fuzzy co-clustering ?

Here, some of the advantages of the fuzzy co-clustering framework for text mining can be summarized, compared with other soft clustering or co-clustering approaches in literature.

Comparing with some soft clustering approaches such as fuzzy c-means, fuzzy co-clustering does not apply an explicit similarity measurement between the documents, and therefore, it avoids directly suffering from the "curse of dimensionality".

Comparing with those state-of-the-art soft (overlapping) co-clustering methods which also generate soft memberships for both documents and words, such as soft spectral co-clustering or overlapping model-based co-clustering, fuzzy co-clustering does not require very strong assumptions or user interference to the dataset itself. For example, while fuzzy co-clustering is applied on the data in the vector representation, the spectral co-clustering must work on a specific graph constructed based on the vector representation of data. How well the graph is constructed, that is one of the key factors to the performance of a spectral clustering method and it is quite dataset dependent. It may not be straightforward to the user in most of the cases.

Last but not least, the computational complexity of fuzzy co-clustering is usually significantly lower than either spectral co-clustering and model-based co-clustering [31, 78].

In conclusion, we realize the strength and potentials of using fuzzy co-clustering framework for the categorization of large high dimensional textual data, compared with other popular clustering techniques reviewed above. Therefore, we select it as the basis for one of the research directions, propose a few novel semi-supervised fuzzy co-clustering approaches, and report them in Chapter 3.

2.5 Existing Semi-Supervised Clustering Approaches

Semi-supervised clustering, which employs both supervised and unsupervised data for clustering purpose, has received significant amount of attention in recent years. By

incorporating prior knowledge on the supervised data available to users, it is aimed to mine and better understand the structure of unknown data and to more closely follow the user's preferences through limited supervision, therefore, improve the clustering performance. The prior knowledge can be represented into several different types, such as ground truth class label, pair-wise categorical constraints etc. It is summarized in [102] , "Existing methods for semi-supervised clustering can be also mainly grouped into two strategies: search-guiding based [103-106] and similarity-adapting based methods [107-110]." The former cases make good use of the prior knowledge to guide the clustering process which is more relied on the novel design of the algorithm itself; while the latter cases focus on improving the effectiveness of the similarity measure by learning new distance functions by another independent framework or algorithm, so that the knowledge, for example: pair-wise constraints, can be easily satisfied when the new measure is applied to the clustering task. Other than these two main categories, recently some non-linear methods using kernels [111-117], has also proposed and been proved powerful. In these methods, kernel functions are the key factor to map the data into a new feature space implicitly; therefore, a cluster assignment is performed with the help of the nonlinear boundary in the original space. However, similar with the drawback of fuzzy co-clustering, the selection of the kernel parameter need to be tuned manually due to no sufficient prior knowledge provided.

2.5.1 Search-Guiding based Approaches

Search-guiding based approaches usually make use of some available prior knowledge mainly in terms of class labels [118-120] or a group of pairwise constraints on a few objects [103, 106, 121, 122] to positively guide the clustering process towards a more appropriate data partitioning. The former approaches used the labeled data as a good initialization, and the constraints generated from the labels to guide the clustering process. Meanwhile, in the latter approaches, each constraint only specifies whether a pair of data points "must" or "cannot" be clustered together. Other than these two major forms, triple-wise constraints proposed in [110], co-training between labeled and unlabeled data [123], manifold discovery of the data structure [88] are all the useful ways that can be considered for semi-supervised clustering. Search-guiding based clustering approaches has been broadly developed by coming with *kmeans* [118], fuzzy clustering [105, 120, 124, 125], co-clustering [126-128], model-based clustering [103, 129], spectral clustering [115] and NMF [102, 130, 131]. In the following a few paragraphs, we look into some of clustering approaches that are very relevant to our own research work.

kmeans

Constrained-*kmeans* [118] and Seeded-*kmeans* [118] are two existing algorithms in semi-supervised clustering with class labels. In both algorithms, some labeled data are incorporated

into *kmeans*. In the case of Constrained-*kmeans*, this prior knowledge remains unchanged throughout the clustering process, while in Seeded-*kmeans* these labeled objects allows going to other clusters depends on the similarities. The former algorithm is only appropriate when the initial seed labeling is noise-free. However, the latter one is more robust to noise because the noisy labels can be abandoned during the course of the clustering process. Moreover, a disadvantage reflected from the performance is that the sensitivity to the distribution of the labeled objects. If the user only has an incomplete set of labeled objects, the performance may be badly affected.

Pair-wise Constrained *kmeans* (PC-*kmeans*) [106] is also proposed, when only pair-wise constraints between a few objects can be used. The constraint that indicates two objects should belong to the same cluster is referred to a *must-link* constraint, and so it is a member of ***MLS,*** while the constraint that indicates two objects should belong to different clusters is referred to a *cannot-link* constraint, and it is a member of ***CLS***. It said that these constraints must be strictly satisfied at the higher priority than the similarity values in each iteration of the partitioning. However, it is noted that some of the objects may be not able to partition into any of the clusters in the current search space. It is supposed related to the sequence of the partitioning of the objects. The steps of the algorithm are described in Table 2.1.

Fuzzy Clustering

From the literature review, we noted some clustering approaches had been exploring various forms of prior knowledge being incorporated into fuzzy clustering framework. While prior knowledge in the form of class labels is used in [105, 124], pair-wise constraints-based approaches such as PCCA, is proposed in [125]. More than that, [104] introduces an active selection mechanism to select proper constraints in order to reduce the impact to the performance caused by the selection of constraints which encountered in PCCA. However, with a fixed fuzzifier *z*, this kind of approach may not be suitable for clustering high-dimensional and sparse text datasets. Instead of directly incorporating the constraints into the FCM framework as discussed above, P-FCM [132] is an interesting method which augments FCM by adding another optimization step at each iteration. A number of proximity "hints" (values) serve as the pair-wise constraints, and the overall difference between the given proximity values and those computed ones based on the membership should be minimized by a separate gradient-driven optimization process. Obviously, the extra computation time is needed by the process. In 2010, Pedrycz et al. proposed another interesting approach [133] which make use of the certain knowledge call the "viewpoint". These viewpoints are actually in terms of some expected numeric values on a few feature dimensions.

Table 2.1: The algorithm of PC-*kmeans*

Input: Dataset ***D***, number of clusters *k*, *must-link* constraint set: ***MLS***, *cannot-link* constraint set ***CLS***
Output: Partition Matrix ***P***.
Method:

1. Let $c_1 ... c_k$ be the initial cluster centroids.
2. For each object d_i, assign it to the closest cluster c_r while the respective constraints involved d_i is not violated in both ***MLS*** and ***CLS***. **If no such cluster exists, fail. (return** empty**)**
3. For each cluster, update the center by averaging all d_i that have been assigned to it.
4. Iterate between (2) and (3) until convergence

Co-clustering

The research on both document co-clustering and semi-supervised clustering has attracted substantial attention in the past a few years. It is natural that researchers are exploring ways to combine the strength of both approaches to further improve the clustering performance. As co-clustering categorizes both the objects and features simultaneously to achieve an equivalent dimension reduction, in principle the most significant improvement made on the respective semi-supervised approaches could be: it is able to incorporate more prior knowledge from multiple sources, and make them helping each other to improve the overall clustering performance. Some efforts have been made on how to develop the semi-supervised version based on certain popular existing co-clustering methods to with incorporating class labels [127, 130] or pair-wise constraints [74, 102, 126, 134], accordingly. Two regularized co-clustering approaches with dual supervision (RCCDS) in the form of a few class labels are proposed in [127]. One is on manifold regularization with bipartite Graph Laplacian (MR), the other is on matrix approximation to potentially assist Spectral Bipartite Graph co-clustering (MA). The former one is good at speed but only able to handle small textual data, while the latter one has no restriction on the size of data but tends to be very slow due to an inner loop and an outer loop issue. Shi et al proposed another efficient Semi-supervised Spectral co-clustering method using pair-wise Constraints [128]. Huang proposed a semi-supervised hierarchical co-clustering method[135]. Another Probalistic-based co-clustering approach called HMRF-ITCC [126, 136] makes use of a two-sided Markov random filed to model both the document and word constraints based on un-supervised information-theory co-clustering. The constraints can be from some nearest neighbors pre-defined by the users. A summary of some semi-supervised co-clustering approaches are listed in Table 2.2.

Table 2.2: A summary on various Semi-supervised Co-clustering Approaches

Algorithms	Pre-processing required	Source of Prior knowledge	Form of Prior Knowledge	Partition
SS-WNMF	-	D	Label	Soft
PMFCC	kmeans	D	PW Constraints	Soft
SS-RNMF	-	D	PW Constraints	Soft
SS-NMF-CC	Metric learning	D	PW Constraints	Soft
CP	kmeans	W	Label	Hard
OSS-NMF	kmeans	D&W	PW Constraints (D) Label (W)	Hard
DRCC	-	D&W	Nearest Neighbor Label (D&W)	Hard
RCCDS	-	D&W	Label (D&W)	Soft
HMRF-ITCC	HMRF	D&W	PW Constraints (D&W)	Hard

NMF-based

Plenty of semi-supervised NMF-based approaches are conducted by incorporating with a set of pair-wise constraints. Penalized Matrix Factorization for Constrained Clustering (PMF) [74] is formulated by incorporating this type of information into the original NMF model in [67]. The objective function to include penalties for violated constraints. The objective function of PMFCC is shown as below:

$$J_{PMF} = \left\| X - FG^T \right\|_F^2 + tr\left(G^T \Theta G\right) \text{ s.t. } G \geq 0, F \in R_+^{m\times k},\ G \in R_+^{n\times k}, \Theta \in R_+^{n\times n} \quad , \quad (2.21)$$

Define matrix $\boldsymbol{\Theta}$ with its (i,j)-th entry Θ_{ij} where

$$\Theta_{ij} = \begin{cases} \tilde{\theta}_{ij}, & \left(d_i, d_j\right) \in \boldsymbol{CLS} \\ -\theta_{ij}, & \left(d_i, d_j\right) \in \boldsymbol{MLS} \\ 0, & otherwise \end{cases} .$$

where θ_{ik} and $\tilde{\theta}_{ik}$ are both non-negative values set by users which represent the penalties for violating the *must-link* constraints and *cannot-link* constraints, respectively. In fact, they serve as the weighing factors that control the degree of enforcement of the prior knowledge. F and G remain the word cluster indictor and document cluster indicator respectively. Then, the clustering process becomes solving a minimization problem of Eq. (2.21) by iteratively update F and G by the following two equations.

$$F = XG\left(G^T G\right)^{-1} \quad , \quad (2.22)$$

$$G_{ic} \leftarrow G_{ic} \sqrt{\frac{\left(X^T F\right)_{ic}^{+} + \left[G\left(F^T F\right)^{-}\right]_{ic} + \left(\Theta^{-} G\right)_{ic}}{\left(X^T F\right)_{ic}^{-} + \left[G\left(F^T F\right)^{+}\right]_{ic} + \left(\Theta^{+} G\right)_{ic}}} \quad , \tag{2.23}$$

where Θ^{+}, Θ^{-}, $\left(F^T F\right)^{+}$, $\left(F^T F\right)^{-}$, $\left(X^T F\right)^{+}$, $\left(X^T F\right)^{-}$ are all nonnegative, and satisfy the following three transformations.

$$\Theta = \Theta^{+} - \Theta^{-} \quad , \tag{2.24}$$

$$F^T F = \left(F^T F\right)^{+} - \left(F^T F\right)^{-} \quad , \tag{2.25}$$

$$X^T F = \left(X^T F\right)^{+} - \left(X^T F\right)^{-} \quad . \tag{2.26}$$

The same idea of penalties (cost) of violating a constraint is applied in SS-RNMF (stands for semi-supervised clustering on relational non-negative matrix factorization) [102] is developed, In this work, Chen et al. directly performed a symmetric tri-factorization on a symmetric data similarity matrix *R* with additional *must-link* and *cannot-link* pair-wise constraints. Each constraint refers to a pair of objects which is a entity in *R*. If object *i* and *j* belong to the same category, then the corresponding element r_{ij} in *R* is set to the value of the maximum similarity found in *R* while r_{ij} is set to the value of the minimum similarity if *i* and *j* belong to different clusters.

Some effort is also made on transforming the prior knowledge from word domain into the document domain. A representative work (named "CP") is proposed based on orthogonal nonnegative matrix tri-factorization [137]. The prior knowledge can be represented as F_0, which is a complete specification of the categorization of words. Then, the user enforces the final solution for *F* to be close to F_0 by an iterative optimization process. The orthogonality condition on both word and document domain are still imposed to avoid ambiguity. As we have discussed in section 2.4 that NMF could have some similar behavior with co-clustering dual knowledge. In [134], Ma et al. proposed a novel algorithm named OSS-NMF which consider dual constraints between documents and words. It made a simple modification based on [137] by adding in pairwise constraints on document domain which is exactly the same as PMFCC did. Lexical knowledge in the form of domain-dependent sentiment-laden terms is also been incorporated into the word domain in [131, 138]. It could be applied on a variety of real-world sentiment prediction tasks.

Moreover, in some cases, NMF may not always use pair-wise constraints to improve the clustering. For example in [130], Lee and Yoo developed multiplicative updates for semi-supervised NMF to minimize a sum of weighted residuals, each of which involves the

nonnegative 2-factor decomposition of the data matrix or the label matrix, constructed by a set of labeled objects in the dataset.

2.5.2 Similarity-Adapting based Approaches

Similarity-adapting approaches [107-109, 139] learn a more effective new distance (similarity) metric from the prior knowledge e.g.: some given proximity (similarity) values between a few reference objects, by an independent DML algorithm before the real clustering process is carried out. So it is also named "metric-based approach". This new similarity function should be a better measure to govern the relationship between the remaining objects, while the proximity "hints" given as the knowledge can be easily satisfied. in recent years, quite a few new similarity measures were explored under different DML algorithm, such as the Euclidean distance modified by a shortest-path algorithm [140], or Mahalanobis distances adjusted by convex optimization [141]. Many of them are focus on learning the family of Mahalanobis distances [29] which usually work well on low dimensional data, but are computationally expensive or even infeasible when handling high-dimensional data, or Bergman Distance [27, 43, 51]. Recently, Steven Ho [142] investigates how to learn a Bergman Distance function using a nonparametric approach. It is necessary to point out, while similarity-adapting methods appear to be more applicable to a wide range of applications, as pointed out in [104], these approaches need either significantly more supervision or specific strong assumptions regarding the target similarity measure.

2.5.3 Combined Strategy

Some recent semi-supervised clustering works tend to combine the strength of search-guiding, distance metric learning and kernels. For example, the semi-supervised co-clustering based on non-negative matrix factorization (SS-NMF-CC) reported in [143] is one of the recent efforts made under NMF framework. It is a triplet co-clustering approach (tri-factorization), which has an additional layer of complexity to handle heterogeneous data clustering, as compared to the other existing NMF approach report in [102] which is only performed on homogeneous datasets. The process is carried out by simultaneous co-clustering the *document-word* together with *document-categories*, inferring the clusters of categories, documents and words, respectively. In this SS-NMF-CC, firstly a distance metric L is learnt based on the provided *must-link* and *cannot-link* constraints on each pair of documents, and then the new *word-document* and *document-category* matrices can be calculated in the way as explained in [108]. Secondly, two modality importance factors α and β are selected based on the new metrics obtained by L, to decide the relative importance of "word" and "category", as "categories" and "words" may play a different role in the clustering of documents. It is noted these two objectives must be achieved simultaneously because the modality selection and distance

metric learning are strongly dependent on each other. Hence, an algorithm has been proposed to iteratively update L, α and β by certain optimization processes before the non-negative tri-factorization is performed. Therefore, the computational cost is higher than other NMF-based approaches. Quite a few distance metric learning approaches in fact conduct their semi-training process through (multiple) kernel(s). Yin [117] proposed a new adaptive semi-supervised clustering kernel method based on metric learning, which simultaneously performs clustering and metric learning. It successfully overcomes the pairwise constraints violation problem and automatically estimates the kernel parameters.

2.6 Transductive Learning Approaches

Other than the existing semi-supervised clustering approaches we reviewed above, a subfield of the semi-supervised learning techniques called transductive learning [144], is also revelant to semi-supervised clustering. It makes use of just a few labeled objects to predict the labels for the majority of un-labeled objects without a explicit predictive model. Therefore, it is useful when the number of training sample in terms of class labels is too small to train a learning (predictive) model. Meanwhile, the transductive learning method is able to make use of the information from both labeled and unlabeled objects to obtain the cluster labels for all the unlabeled objects.

In recent years, a number of graph-based transductive learning algorithms that achieve state of the art performance for image pattern recognition and document categorization have been developed [144-149]. The working principle of the Graph-based Transduction can be described as follows: first of all, a weighted graph (input) is constructed by treating objects as nodes and the weights of edges between any pairs of nodes are estimated using certain affinity functions. Then, with a few labeled nodes in the graph, a continuous classification function $\boldsymbol{F}$ (output) is estimated based on the graph by minimizing a cost function. This cost function enforces a tradeoff between two terms: the smoothness of $\boldsymbol{F}$ on the whole graph and the accuracy of $\boldsymbol{F}$ at fitting the available label information for the labeled nodes. Then, correctly propagating label information from the labeled nodes to unlabeled nodes for the desired output $\boldsymbol{F}$ is achieved by the greedy minimization of the cost function.

It is clear that this kind of the approaches are extremely dependent on the initial labels. Moreover, some popular graph-based transductive learning algorithms includes LGC [150] and GFHF [151] may suffer from a biased label prediction if the input labels are disproportionately imbalanced. A more robust method called GTAM [152] introduces an additional node regularizer to balance the influence of labels from different classes. Therefore, the side effect from the labels in low density and unreliable regions can be reduced. In addition, GTAM conducts iteratively optimization on both the label set and the cost function, to avoid prematurely committing to intractable prediction results. On the other aspect, the

label propagation in GTAM could oscillate and backtrack from predicted labelings in previous iterations without convergence guarantees. Therefore, GTAM performs a consistent label propagation [152]. In other words, the initial given labels must be considered as golden truth and thus never changed. In each iteration, once an unlabeled object is labeled, it is removed from the un-labeled set and added into the labeled set for the prediction for other un-labeled objects in the next iteration. Because of this, GTAM cannot handle the problem with mislabeled samples. GTAM-LDST [153] provides an additional label diagnosis through self-tuning to remove the noisy or unreliable label, and then use a set of refined labels to conduct the label propagation to all the other unlabeled objects. The self-tuning consists of a few iterations. The least confident label is removed before a new label is added in to maintain a fix number of labels, and each individual operation of labeling and unlabeling leads to the update of label regularization matrix **V**.

The time complexity can be considered as a main disadvantage of graph-based transductive learning. For example in GTAM, the algorithm's loops the alternative minimization at most *n-g* times due to the greedy assignment, where *g* is total number of labels given in the dataset. The total runtime of the algorithm is $O(n^3\text{-ln } n)$. Therefore, this kind of methods may be not efficient enough to handle large data.

Although transductive learning looks quite similar with the semi-supervised clustering, the cost of applying transductive learning is indeed more expensive than applying a semi-supervised clustering for the same categorization task.

It is very important to emphasis that, at least one class label must be provided from each of the classes to carry out the label propagation by a graph-based transductive learning method. To be more specific, if this requirement is not satisfied for GTAM, the element in node regularizer **V** will go to infinite and then the propagation immediately fails.

While semi-supervised clustering, also named as "semi un-supervised learning" are designed without such a restriction. Moreover, in many semi-supervised clustering approaches, the knowledge incorporated into the clustering process is a few pair-wise constraints, including *must-link* and *cannot-link* constraints between two objects.

2.7 Evaluation Metrics

Evaluation metrics or cluster validity measures are used to assess the performance of a clustering algorithm and make comparisons between different clustering approaches. In general, there are three types of evaluation metrics namely external metrics, internal metrics, and relative metrics. External metrics are defined to measure the degree of agreement between the clusters generated by a clustering algorithm and the pre-specified structure or reference partitions of the dataset so call "ground truth". Internal metrics measure the quality of clusters produced by a clustering algorithm with the degree of inter-cluster separation and

intra-cluster compactness. These metrics are usually used for exploring the underlying structures and discovering the true number of clusters, as they are defined only based on the dataset itself. Relative metrics are usually used to compare the clustering results of the same clustering algorithm but with different parameter settings. Extensive validation methods for various types of clustering algorithms are summarized in [154]. As we assume data collections using in our experiments are all with "ground truth" labels which is assigned by human experts, external metrics are used for evaluating the clustering results in our experiments to quantify the agreement between a set of human-generated labels and another set of algorithm-generated labels. A clustering algorithm is said to be "good" if its resulting clusters show high agreement with the ground truth and it is said to be "bad" if its resulting clusters seldom agree with the ground truth.

We focus on the review of several most widely used evaluation metrics of clustering. We use *cluster* refers to the clustering result created with algorithms and *class* refers to the ground truth. n is the total number of objects; n_r is the number of objects in the rth cluster; n_c is the number of objects in the cth class; and n_r^c is the number of objects that in both class c and cluster r. k and l are the number of clusters and classes, respectively, and usually they are set to equal in the experimental settings. The *precision* and *recall* of class c and cluster r denote as P and R are two intuitive statistic metrics of performance that are defined as below [155]:

$$P(c,r) = n_r^c / n_r \tag{2.27}$$

$$R(c,r) = n_r^c / n_c \tag{2.28}$$

From Eqs.(2.27) and (2.28), it can be known that there are $k \times l$ sets of precisions and recalls for each pair of classes and clusters. When computing the overall precision and recall, we use the post optimum document-to-cluster assignment (i.e. the one that results in the best precision and recall) to guide us in deciding which of the ground-truth classes a cluster represented and compute the average over all k clusters. Another measure *Purity* only counts the largest precision for each cluster, and the overall purity is calculated by the average of purities among all clusters. Although purity looks simple and compact, it may not be a reasonable metric when *precision* and *recall* are unbalanced. This situation is very often on clustering extremely unbalanced datasets. A high Purity value due to high precisions might come together with low recalls, and more than one cluster is dominated by the objects from the same class. To solve this bias, two more metrics can be introduced.

F-measure [156] reflects the overall quality of the resulting clusters with the weighted combination of *precision* and *recall* as in Eq.(2.29). Typically, *precision* and *recall* are given equal weight with α =1. The total *F-measure* is calculated as the average of the largest *F-measure* of each cluster to all the classes, just similar to the largest *precision & recall* assignment.

$$F-measure(c,r)_{\alpha}=\frac{(1+\alpha)P(c,r)\times R(c,r)}{\alpha P(c,r)+R(c,r)} \tag{2.29}$$

$$F-measure(c,r)_{\alpha}=\sum_{c=1}^{l}\frac{n_c}{n}\arg\max_r\left\{F-measure(c,r)\right\} \tag{2.30}$$

Accuracy is another popular metric, which measures the clustering performance from an established one-to-one match between the *clusters* and *classes* with the help of Hungarian algorithm, for cases when $k = l$. The *Accuracy* value is defined as:

$$Accuracy=\frac{1}{n}\sum_{r=1}^{k}n_r^p \tag{2.31}$$

where n_r^p is the number of common objects in the *r*th cluster and its matched class *p*.

Other two metrics drawn from the information theory Sometimes, in real application, the number of clusters found by a clustering algorithm might be not equal to the real number of classes. In such case, *NMI* [78], stands for normalized mutual information, defined in Eq. (2.32), is a superior measure.

$$NMI=\frac{\sum_{r=1}^{k}\sum_{c=1}^{l}n_r^c\log\left(\frac{n\cdot n_r^c}{n_r\cdot n_c}\right)}{\sqrt{\left(\sum_{r=1}^{k}n_r\log\frac{n_r}{n}\right)\left(\sum_{c=1}^{l}n_c\log\frac{n_c}{n}\right)}} \tag{2.32}$$

The final choice of the evaluation metric is dependent on the particular environment. Researchers in different domains may prefer different metrics, e.g. for researchers from IR field, *F-measure* might be favored while someone from AI field may choose *NMI*. The choice of metrics is also dependent on the technique or theory on which the clustering algorithm based, e.g. for information-theory based clustering, *entropy* and *NMI* might be chosen while for clustering adopts statistic theory, *purity* and *F-measure* might be used. Rather than selecting one particular evaluation metric, some studies use several metrics to provide a more convincible assessment of the clustering algorithms.

Chapter 3

Semi-Supervised Fuzzy Co-Clustering

3.1 Overview

As discussed in Section 2.4.7, fuzzy co-clustering is a kind of un-supervised machine learning technique which is capable for the automatic categorization of large and overlapped high-dimensional textual data collections, while requires a relatively low time complexity, compared with other popular clustering techniques such as NMF-based clustering [69, 75], spectral clustering [61] and model-based clustering [31]. To be more specific, It avoid using an explicit defined distance (similarity) function to measure In this chapter, we study how to effectively incorporate two types of available prior knowledge: class labels and pair-wise constraints into the fuzzy co-clustering framework to further improve its performance. Three different semi-supervised clustering approaches namely: Semi-Supervised Fuzzy Co-clustering with Labelling (SS-FCL) [157], Semi-Supervised Fuzzy Co-clustering with Constraints (SS-FCC) [158] and Dual Semi-Supervised Heuristic Fuzzy co-clustering with Respini Condition (DSS-HFCR) has been proposed and reported accordingly.

Based on the vector representation, we formulate the document clustering process as an optimization problem, build the objective function referring to the competitive agglomeration cost with fuzzy terms and the additional supervised constraints corresponding to the two types of knowledge, and design an iterative algorithm to infer the document and word clusters. Our objective is to increase the clustering accuracy, and reduce the sensitivity to the fuzzifier parameters with limited prior knowledge while the complexity of the algorithm may remain relatively low. Empirical study is then conducted on a number of publicly available benchmark textual datasets. The clustering result is compared with several state-of-the-art semi-supervised clustering methods to show the effectiveness of the proposed works.

The structure of this chapter is organized as follows: two partitioning-ranking based approaches are introduced in Section 3.2, including the problem statement, formulation of the objective functions, the derivation of the updating rules and the detailed steps of algorithms. Then, a dual-partitioning based approach which is able to make use of the prior knowledge in both document and word main is introduced by Section 3.3. Experimental results of these

three approaches are reported and discussed in Section 3.4. Finally we draw the conclusion in Section 3.5.

3.2 Partitioning-Ranking based Approaches

3.2.1 Problem Formulation

In the Vector Space Model, we assume a textual dataset ***S*** which consists of *n* documents, and *m* words can be represented using a matrix ***D*** where rows index the documents to be clustered and columns denote the distinct words. An entry d_{ij} in this matrix denotes the *tf-idf* value of word *j* in document *i*. For a fuzzy co-clustering task, taking ***D*** and a given cluster number *k* as the input, the goal of clustering process is to partition these *n* documents into meaningful sub-groups (co-clusters) based on the similarities of the documents in content. The output is the document membership matrix **U** and word membership **V**, which indicates the cluster assignment of each document and words in the textual dataset.

Before introducing the proposed methods, the notations used through this chapter and their brief descriptions are listed down in Table 3.1.

3.2.2 Formulation of SS-FCL and SS-FCC

Fuzzy CoDoK [85], stands for Fuzzy Co-clustering with Documents and Keyword, is an existing co-clustering algorithm for categorizing large text corpus. The objective function of Fuzzy CoDoK is written as:

$$J_{FuzzyCodok} = \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri} v_{rj} d_{ij} - T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri}^2 - T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj}^2 \quad (3.1)$$

Table 3.1: Notations for SS-Fuzzy Co-Clustering

Notation	Description
S	set of all the documents in textual corpus
D	matrix representation of ***S***
LS	set of all labeled documents
MLS/ CLS	set of all *must-link*/*cannot-link* constraints
ConS	set of documents/words with pair-wise constraints
FrS	set of documents/words without pair-wise constraints
U	document partitioning membership matrix
V	word partitioning/ranking membership matrix
d	normalized single document vector, $\|d\|=1$
n	total number of documents of the collection
m	total number of terms(words)
k	number of clusters
c	number of classes
u_{ri}	partitioning membership of document *i* in co-cluster *r*
v_{rj}	partitioning/ranking membership of word *j* in co-cluster *r*
T_u	fuzzifier for document partitioning membership
T_v	fuzzifier for word partitioning/ranking membership
T_d	weighting factor for the knowledge from document domain
T_w	weighting factor for the knowledge from word domain
l_{ri}	Initial label indication in terms of partitioning membership

The optimization of the Eq.(3.1) is subject to the following two constraints given in Eqs.(3.2) and (3.3).

$$\sum_{r=1}^{k} u_{ri} = 1 \ , \ u_{ri} \in [0,1] \text{ for } i = 1,\ldots, n, \tag{3.2}$$

$$\sum_{j=1}^{m} v_{rj} = 1 \ , \ v_{rj} \in [0,1] \text{ for } r = 1,\ldots, k \ , \tag{3.3}$$

where u_{rj} denotes the fuzzy partitioning membership of document i in co-cluster r, and v_{rj} denotes the ranking membership of word j in co-cluster r. The former term is a soft cluster indicator which measures how possible i is assigned to cluster r; while the latter term is a relative weight which measures how typical j is compared with all the other words with respect to cluster r.

First, as a fuzzy co-clustering approach, both document (object) membership u_{ri} and word (feature) membership v_{rj} should be taken into account in the objective function. In principle, it should be provided so as to group the documents and words which have a higher correlation to each other. Therefore, the degree of aggregation term $\sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri} v_{rj} d_{ij}$ should be maximized among clusters to accomplish the clustering task. In other words, the maximization of this term is intended to make highly related documents and words (as indicated by high d_{ij} values) to be co-clustered together (i.e. assigned to the same co-cluster). The motivation behind is that a high quality co-cluster should be the one with a strong coherence bonding among its members (i.e. documents and words).

The second term $T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri}^2$ and third term $T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj}^2$ are the fuzzifier terms using Gini Index. They will be maximized when u and v are equally distributed. The purpose of having them is similar to the parameter z in the fuzzy *c-means* algorithm, i.e. to fuzzify the resulting co-clusters. The parameters T_u and T_v can be used to adjust the levels of fuzziness of the document partitioning and word ranking memberships, respectively.

Now, as a semi-supervised approach, suppose the class labels of a few documents in the dataset (denoted by ***LS***) are available to the user; the membership u_{ri} of those labeled documents in ***LS*** can be easily obtained and used at the initial state of the clustering process. These memberships usually hold fixed binary values, i.e.: 1 in the cluster corresponding to its ground truth class, and 0 in the other clusters. As we are doing fuzzy clustering, each document in the corpus is assumed more or less related to several topics based on the fuzzy set principle. In this case, the prior knowledge may be also available in term of soft label indication by assigning a numeric number between 0 and 1 to the initial memberships in each of the clusters. A new term l_{ri} is used to denote the initial label indication in terms of the

membership value u_{ri} for every labeled document. The same constraint on u_{ri} also applies to l_{ri}, which is defined as below:

$$\sum_{r=1}^{k} l_{ri} = 1 , \quad \text{for } i = 1,\ldots n, \tag{3.4}$$

In principle, the cluster search of the unlabeled documents can be correctly guided by the initial memberships given on the labeled documents, and these membership values should not be changed during the clustering process, if they are trusted as the ground truth [118]. There are some discussions reported in literature about in which scenario the initial label indication should be fixed for crisp clustering. It is argued that the user may take the risk of choosing a noisy label if all the labels are fixed in their corresponding clusters. Meanwhile, using the labels or constraints at the initial state of the clustering process may sometimes encounter the violation of the prior knowledge.[117]

In this thesis, the interference of noisy label is out of our consideration in the problem formulation. However, as a fuzzy clustering approach, we believe it is better to get the fuzzy memberships of those labeled documents updated together with other unlabeled ones, especially when sometimes the initial membership of the labeled document is difficult to obtain precisely and have to roughly assign the binary values. Meanwhile, to avoid the label violation at the end of the clustering process, an additional supervised constraint term $T_u \times T_d \times \sum_{r=1}^{k} \sum_{d_i \in LS} \left(u_{ri} - l_{ri}\right)^2$ is added to the objective function of Fuzzy Codok, where T_d is the coefficient to control the relative weightage between the supervised information and the degree of aggregation on the whole dataset. The minimization of this term is able to keep the updated membership u_{ri} of a labeled document close to its respective initial value l_{ri}, but not necessarily equal to it.

On the other hand, to keep the memberships of labeled documents updated also helps the user to look for a suitable set of parameters. It is realized that the efforts required on the parameter tuning may be a common issue for all the fuzzy co-clustering approaches, compared with *kmeans* or fuzzy c-means. The experimental studies reported in many research works [55, 85] show that a set of suitable parameters is essential for a good clustering performance on fuzzy co-clustering and we expect it should not cause any label violation at the end of the clustering process. In other words, checking if any label violation occurs at the end of the clustering can be considered as a good guideline for finding out the suitable parameter for the semi-supervised fuzzy co-clustering.

Therefore, the overall design of SS-FCL ensures that each document will get a fuzzy membership distribution; meanwhile those labeled documents can play the correct role all the

time for guiding the cluster search during the clustering process. The objective function of SS-FCL can be finally formulated as:

$$J_{SS-FCL} = \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri} v_{rj} d_{ij} - T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri}^2 - T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj}^2 - T_u \times T_d \times \sum_{r=1}^{k}\sum_{d_i \in LS} \left(u_{ri} - l_{ri}\right)^2 \quad (3.5)$$

Sometimes, the available prior knowledge is not in the form of exact class labels, but only in the form of '*must-link*' or '*cannot-link*' constraint between a pair of documents. In this case, the dataset is firstly divided into two subsets: constrained set (***ConS***) and free set (***FrS***). The documents involved in at least one constraint will be put in ***ConS*****,** while the rest will be put in ***FrS***. We assume each document in a pair-wise constraint has a 'virtual label', which is a categorical variable. This variable can take two types of values: one is a user assigned category; the other is the ground truth class label if such specific information is available. These constraints are then grouped into two training sets. One set is used to specify the '*must-link*' constraints denoted as ***MLS;*** and the other is used to specify the '*cannot-link*' constraints denoted as ***CLS***. We make sure that each pair of documents in ***MLS*** very similar to each other in content, therefore, it indicates a *must-link* constraint; while each pair of the documents in ***CLS*** is dissimilar in content, so it indicates a *cannot-link* constraint. Moreover, the *must-link* constraints will represent an equivalence relation. Hence, it is possible to derive a collection of transitive closure in the ***MLS***, which any document pair in the same transitive closure must share the same 'virtual label'.

By assuming each document in the dataset is more or less related to several topics based on the fuzzy set principle, we realized the prior knowledge in terms of pair-wise constraints can be now reflected via the fuzzy membership values during the initialization. Each document in a pair-wise constraint is given a higher degree of membership to the category *c* which corresponds to its virtual categorical label, and lower degree of membership to the other categories. Then, based on the constraint $\sum_{r=1}^{k} u_{ri} = 1$, the term $\sum_{r=1}^{k} u_{ri} u_{ro}$ should be maximized if d_i and d_o has the same 'virtual label', and it should be minimized if d_i and d_o has a different 'virtual label' during the clustering process. The combined additional supervised term in the objective function can be expressed as: $\left(\sum_{(d_i,d_o)\in MLS} \sum_{r=1}^{k} u_{ri} u_{ro} - \sum_{(d_i,d_o)\in CLS} \sum_{r=1}^{k} u_{ri} u_{ro}\right)$, and should be maximized through the clustering process. The overall design of the SS-FCC ensures that each document will get a fuzzy membership distribution, and the violation of the pair-wise constraints is minimized at the end of the clustering process.

Hence, the objective function of the proposed SS-FCC can be now formulated as:

$$J_{SS-FCC}=\sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}u_{ri}v_{rj}d_{ij}-T_u\sum_{r=1}^{k}\sum_{i=1}^{n}u_{ri}^2-T_v\sum_{r=1}^{k}\sum_{j=1}^{m}v_{rj}^2$$
$$+T_u\times T_d\left(\alpha_d\sum_{(d_i,d_0)\in MLS}\sum_{r=1}^{k}u_{ri}u_{ro}-\beta_d\sum_{(d_i,d_o)\in CLS}\sum_{r=1}^{k}u_{ri}u_{ro}\right) \quad (3.6)$$

where α_d and β_d are the coefficient to adjust the relative weightage (importance) between the *must-link* constraint and the *cannot-link* constraint.

While the design of SS-FCL is to optimize the objective function based on a small set of class labels, the design of SS-FCC is to optimize the objective function based on two sets of pair-wise constraints. The supervision provided by partial label indication or pairwise relationships are now reflected by the new supervised constraint terms in the objective function of SS-FCL and SS-FCC, respectively.

The constraints in Eq.(3.2) and Eq.(3.3) both apply to SS-FCL and SS-FCC. The former one conforms to the Ruspini's condition [159]. Due to this constraint, the document memberships computed by SS-FCL / SS-FCC reflect how the documents are partitioned across different co-clusters. Meanwhile, the latter constraint enforces the word memberships computed by SS-FCL/SS-FCC reflect the words' ranks in the co-clusters. In this case, in every co-cluster, the membership value of a word is decided based on where its similarity to the co-cluster ranks in comparisons to all the other words. Words with higher similarities to the co-cluster will be assigned higher memberships (i.e. higher ranks). Thus, Eq. (3.5) and (3.6) are both essentially the objective function of a fuzzy co-clustering algorithm that combines document partitioning with word ranking, respectively.

3.2.3 Updating Rules

Now we need to solve the above two maximization problems by finding the optimal values of **U** and **V** subject to the corresponding set of constraints, where **U** and **V** denote for the entire document and word membership matrix, respectively. Since *u* and *v* are continuous variables, we use the method of Lagrange multipliers with the first order necessary condition derive the updating rules for *u* and *v*. Therefore, the Lagrangian functions of SS-FCL, SS-FCC are first constructed accordingly as:

$$L_{SS-FCL}=\sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}u_{ri}v_{rj}d_{ij}-T_u\sum_{r=1}^{k}\sum_{i=1}^{m}u_{ri}^2-T_v\sum_{r=1}^{k}\sum_{j=1}^{m}v_{rj}^2$$
$$-T_u\times T_d\times\sum_{r=1}^{k}\sum_{d_i\in LS}\left(u_{ri}-l_{ri}\right)^2+\sum_{i=1}^{n}\lambda_i\left(\sum_{r=1}^{k}u_{ri}-1\right)+\sum_{r=1}^{k}\gamma_r\left(\sum_{j=1}^{m}v_{rj}-1\right) \quad (3.7)$$

$$L_{SS-FCC} = \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri}v_{rj}d_{ij} - T_u\sum_{r=1}^{k}\sum_{i=1}^{n}u_{ri}^2 - T_v\sum_{r=1}^{k}\sum_{j=1}^{m}v_{rj}^2$$
$$+T_u \times T_d\left(\alpha_d \sum_{(d_i,d_0)\in MLS}\sum_{r=1}^{k} u_{ri}u_{ro} - \beta_d \sum_{(d_i,d_o)\in CLS}\sum_{r=1}^{k} u_{ri}u_{ro}\right) + \sum_{i=1}^{n}\lambda_i\left(\sum_{r=1}^{k}u_{ri} - 1\right) + \sum_{r=1}^{k}\gamma_r\left(\sum_{j=1}^{m}v_{rj} - 1\right) \tag{3.8}$$

where λ_i and γ_r are Lagrange multipliers corresponding to the constraints in Eq.(3.2) and (3.3).

From the necessary conditions for the optimality of the Lagrangian function $\boldsymbol{L}$, we take the partial derivatives of $L_{SS\text{-}FCL}$ or $L_{SS\text{-}FCC}$ with respect to u_{ri}. By calculating $\partial L / \partial u_{ri} = 0$, the updating rule for u_{ri} at iteration τ can be derived as below:

For SS-FCL:

$$u_{ri}^{\tau} = \frac{1}{1+T_d}\left\{\frac{1}{k} + \frac{1}{2T_u}\left[\sum_{j=1}^{m} v_{rj}^{\tau-1}d_{ij} - \frac{1}{k}\sum_{f=1}^{k}\sum_{j=1}^{m} v_{fj}^{\tau-1}d_{ij}\right] + T_d \cdot l_{ri}\right\}. \tag{3.9}$$

For SS-FCC:

$$u_{ri}^{\tau} = \frac{1}{k} + \frac{1}{2T_u}\left\{\sum_{j=1}^{m} v_{rj}^{\tau-1}d_{ij} - \frac{1}{k}\sum_{f=1}^{k}\sum_{j=1}^{m} v_{fj}^{\tau-1}d_{ij}\right\} + \frac{T_d}{2}\left[\begin{array}{l}\left(\alpha_d \sum_{(d_i,d_s)\in MLS} u_{rs}^{\tau-1} - \beta_d \sum_{(d_i,d_t)\in CLS} u_{rt}^{\tau-1}\right) \\ -\frac{1}{k}\left(\alpha_d \sum_{f=1}^{k}\sum_{d_i,d_s\in MLS} u_{fs}^{\tau-1} - \beta_d \sum_{f=1}^{k}\sum_{d_i,d_t\in CLS} u_{ft}^{\tau-1}\right)\end{array}\right] \tag{3.10}$$

For the word ranking membership v_{rj} in both approaches, by similarly taking the partial derivation of $L_{SS\text{-}FCL}$ or $L_{SS\text{-}FCC}$ with respect to v_{rj}, the value at iteration τ can be calculated by the updating rules derived as below:

$$v_{rj}^{\tau} = \frac{1}{m} + \frac{1}{2T_v}\left[\sum_{i=1}^{n} u_{ri}^{\tau-1}d_{ij} - \frac{1}{m}\sum_{f=1}^{m}\sum_{i=1}^{n} u_{ri}^{\tau-1}d_{if}\right]. \tag{3.11}$$

From the derived linear updating rules, it can be seen that negative values of u and v might occur during the successive optimization process. In order to guarantee non-negative assignment of membership value to all document and word clusters, the final updating rules need to be derived with the Karush-Kuhn-Tucker conditions to consider the non-negative constraints in Eqs.(3.2) and (3.3) when constructing the Lagrangian function in Eq.(3.5) or Eq.(3.6). Alternatively, a simplified version of the optimization for SS-FCL and SS-FCC can be also implemented by the guideline given in [85]. That is, if any reassigned membership value becomes negative during the clustering process, we simply reset it to 0 and re-normalize the remaining positive ones, so the constraints are still strictly followed.

3.2.4 Algorithms

Now the iterative algorithm of SS-FCL approach can be stated as follows: starting with a 'ground-truth' assignment on each of the labeled documents and a nonnegative random initialization on other unlabeled documents, **V** and **U** are iteratively updated with Eq.(3.11) and Eq.(3.9) respectively in an alternating manner until the successive estimates of **U** is close enough or reaching the maximum number of iterations. Similarly, the SS-FCC algorithm starts with a non-negative soft membership assignment, **V** and **U** are iteratively updated with Eq.(3.11) and Eq.(3.10) respectively in an alternating manner until the successive estimates of **U** is close enough or reaching the maximum number of iterations. It is important to emphasis that the initial memberships for the documents belong to ***ConS*** must follow the instructions given in Section 3.2.2 and satisfy the pair-wise constraints. During the clustering process, the quality of the partition in terms of the criteria defined in Eq. (3.5) and (3.6) are successively improved through reassignment of documents to clusters based on the current word ranking and reforming of word cluster based on the current document partition. The detailed steps of these two algorithms are now given in Table 3.2 and Table 3.3 respectively.

Table 3.2: The SS-FCL Algorithm

Input: Dataset ***S***, the label set ***LS***, number of clusters k, T_u, T_v T_d , stopping threshold ε , number of maximum iteration τ_{max}
Output: Document partitioning membership matrix: **U**, word ranking membership matrix **V**.
Method:

1. Set the initial document memberships of labelled documents:
2. Randomly initialize memberships for the documents belongs to ***S/LS***;
3. **REPEAT**
   3.1 Update v_{rj} with Eq. (3.11);
   3.2 Update u_{ri} with Eq. (3.9);
   3.3 $\tau = \tau + 1$

UNTIL($\max_r \left| u_{ri}^{\tau} - u_{ri}^{\tau-1} \right| \le \varepsilon$) or $\tau > \tau_{max}$

Table 3.3: The SS-FCC Algorithm

Input: Dataset ***S***, Training set ***MLS*** & ***CLS***, number of clusters k, T_u, T_v T_d, α_d, β_d , stopping threshold ε , number of maximum iteration τ_{max}
Output: Document partitioning membership matrix: **U**, word ranking membership matrix **V**.
Method:

1. Manually adjust the initial u_{ri} for the documents in ***ConS*** to satisfy all the constraints
2. Randomly initialize memberships for the rest of the documents;
3. **REPEAT**
   3.1 Update v_{rj} with Eq. (3.11);
   3.2 Update u_{ri} with Eq.(3.10);
   3.3 $\tau = \tau + 1$

UNTIL($\max_r \left| u_{ri}^{\tau} - u_{ri}^{\tau-1} \right| \le \varepsilon$) or $\tau > \tau_{max}$

3.3 Dual Partitioning based Approaches

3.3.1 Formulation of SS-HFCR & DSS-HFCR

SS-FCL and SS-FCC are both partitioning-ranking based clustering approaches for categorizing large textual data, by only incorporating the prior knowledge from document domain. We believe the performance of a semi-supervised co-clustering approach could be further improved if some prior knowledge from word domain can be also incorporated together with the knowledge from document domain. Similar to the prior knowledge from document domain, we believe the knowledge from word domain can also be reflected by the constraints via the partitioning membership value of words. In such situation, the objective function may become over complicated based on either SS-FCL or SS-FCC approaches, as besides the membership for documents, we need to consider two types of memberships for words at the same time in the function. In this case, we consider turning another direction, which directly develops the new approaches based on the dual-partitioning-based fuzzy co-clustering to incorporate dual domain knowledge, instead of the current partitioning-ranking based formulation in either SS-FCL or SS- FCC.

HFCR [55], stands for Heuristic Fuzzy Co-clustering with Ruspini's condition, is a dual partitioning based approach in literature, and shows the strength of handling high dimensional textual dataset through extensive experimental studies. The objective function of HFCR is written as:

$$J_{HFCR} = \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri} v_{rj} d_{ij} - T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri} \ln u_{ri} - T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj} \ln v_{rj} \qquad (3.12)$$

Be different from Eq.(3.5) and (3.6), the optimization of Eq.(3.12) is then subject to the constraint given in Eq.(3.2) and a different constraint on word partitioning membership given in Eq.(3.13). The constraint in Eq. (3.3) is no longer used.

$$\sum_{r=1}^{k} v_{rj} = 1 \ , \ v_{rj} \in [0,1] \text{ for } j=1,\ldots m \ , \qquad (3.13)$$

In HFCR model, the first degree of aggregation takes the same physical meaning as we discussed for Fuzzy Codok. The second and third terms are the fuzzifier terms based on entropy regularization, rather than Gini Index in the objective function of SS-FCL and SS-FCC. Meanwhile, as a dual partitioning method, the constraints followed in Eq. (3.2) and (3.13) both conform the Ruspini's condition, which indicates u_{ri} and v_{rj} computed by this approach reflect how the documents/words are partitioned across diverse co-clusters, respectively.

Before developing the new dual-knowledge based approach based on HFCR, a single domain-knowledge based approach called SS-HFCR in [160] can be easily explored out by

adding the same supervised constraint term in SS-FCC to Eq.(3.12). The objective function of SS-HFCR is expressed as below:

$$J_{SS-HFCR} = \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri} v_{rj} d_{ij} - T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri} \ln u_{ri} - T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj} \ln v_{rj}$$
$$+T_u \times T_d \left(\alpha_d \sum_{(d_i,d_o)\in MLS} \sum_{r=1}^{k} u_{ri}u_{ro} - \beta_d \sum_{(d_i,d_o)\in CLS} \sum_{r=1}^{k} u_{ri}u_{ro} \right) \quad (3.14)$$

Now, we focus on how to further incorporate the knowledge from word domain into Eq.(3.14). In literature, most of the related works incorporate lexical knowledge by only using the label information of the selected words in a particular dataset. This can be obtained by referring some online reference systems such as *ACM Keyword taxonomy*. The entire word collection is usually divided into several classes, and each word is definitely related to a specific class, in other words, a 'ground truth' for the words is explored. Other than the label information, we believe the pairwise relation on words is another type of knowledge, can be not only decided by referring the online taxonomy, but also easily figured out by human judgment. For example, we think *computer* and *hardware* should be in the same class, whereas *soccer* and *algorithm* should be in different classes. In the sentiment analysis field, *terrible* and *bad* can be both considered as 'negative', while *wonderful* and *graceful* can be both considered as 'positive'. In other words, as the same as the pair-wise constraint from the document domain used in SS-FCC, a pair-wise constraint from the word domain is determined by either the virtual categorical index by the user himself or the 'ground truth' of the whole vocabulary set referring the taxonomy. In this thesis, we decide to formulate the new objective function by adding the supervised constraint terms based on the pair-wise relationship, rather than the individual label indication of every word, as it is considered as a more generic formulation that both types of prior knowledge can be successfully incorporated.

Therefore, we construct two more training sets ***MLS*** and ***CLS*** which consist of the *must-link* word pairs and *cannot-link* word pairs, respectively. Under this assumption, the same idea of maximizing (minimizing) $\sum_{r=1}^{k} u_{ri}u_{ro}$ for all the *must-link (cannot-link)* document pairs can be adopted into the word domain. Hence, the objective function of the Dual-Semi Supervised Heuristic Fuzzy Co-clustering (DSS-HFCR) is formulated as:

$$J_{DSS-HFCR} = \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri} v_{rj} d_{ij} - T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri} \ln u_{ri} - T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj} \ln v_{rj}$$
$$+T_u \times T_d \left(\alpha_d \sum_{(d_i,d_o)\in MLS} \sum_{r=1}^{k} u_{ri}u_{ro} - \beta_d \sum_{(d_i,d_o)\in CLS} \sum_{r=1}^{k} u_{ri}u_{ro} \right) + T_v \times T_w \left(\alpha_w \sum_{w_j,w_o\in MLS} \sum_{r=1}^{k} v_{rj}v_{ro} - \beta_w \sum_{w_j,w_o\in CLS} \sum_{r=1}^{k} v_{rj}v_{ro} \right)$$
(3.15)

where *d* denotes for document and *w* denotes for word. Similar with the weighting factor T_d a new weighting factor T_w is added to adjust the relative importance of the prior knowledge from word domain to prior knowledge from document domain. Furthermore, α_w and β_w are the corresponding weighting factors to adjust the relative importance of two types of pair-wise constraints.

Therefore, SS-HFCR published in [160] is considered as one of the simplified version to DSS-HFCR, when prior knowledge coming from document domain only. Similarly, another simplified version of DSS-HFCR can be observed when prior knowledge coming from word domain only. To clearly distinguish these two simplified versions, the SS-HFCR published in [160] is now referred to as SS-HFCR-D in the thesis, and the other version is called as SS-HFCR-W from now on. The corresponding objective function of SS-HFCR-W is then written in Eq.(3.16).

$$\begin{aligned} J_{SS-HFCR-W} &= \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri} v_{rj} d_{ij} - T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri} \ln u_{ri} - T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj} \ln v_{rj} \\ &+ T_v \times T_w \left(\alpha_w \sum_{w_j, w_o \in MLS} \sum_{r=1}^{k} v_{rj} v_{ro} - \beta_w \sum_{w_j, w_o \in CLS} \sum_{r=1}^{k} v_{rj} v_{ro} \right) \end{aligned} \tag{3.16}$$

As discussed in [55], unlike the partitioning-ranking based approaches SS-FCL and SS-FCC, the dual-partitioning schemes without a careful design may suffer from a fundamental flaw that prevents an effective fuzzy co-clustering from being accomplished. The maximization of the degree of aggregation may be biased towards the construction of the co-clusters with larger aggregation value of $\sum_{r=1}^{k} u_{ri} v_{rj}$, and therefore not lead to the correct direction for the desired clustering result. The reason could be shown more clearly when we denote the term into the component-wise inner product of two matrices, i.e. $\boldsymbol{G} : \boldsymbol{D} = \sum_i \sum_j g_{ij}\, d_{ij}$, g_{ij} (each element of $\boldsymbol{G}$) is defined as $\sum_{r=1}^{k} u_{ri} v_{rj}$. It is noted the value of $\sum_{i=1}^{n} \sum_{j=1}^{m} g_{ij}$ can vary from 0 to *nm*. This variation implies that the maximization of the degree of aggregation in this case will be biased towards the construction of co-clusters with larger $\sum_{i=1}^{n} \sum_{j=1}^{m} g_{ij}$ values. Meanwhile, it does not entirely depend on the partitioning on $\boldsymbol{D}$. Therefore, it is not necessary for the co-clusters to have a large $\sum_{i=1}^{n} \sum_{j=1}^{m} g_{ij}$ value in order to capture the real inherent grouping structure of a given dataset.

It is also noted, $\sum_{i-1}^{n} \sum_{j=1}^{m} g_{ij}$ always equals to a constant (i.e. *n*) in the partitioning-ranking based approaches; for this reason they are spared from the bias problem. Therefore, in a dual partitioning-based approach, two auxiliary functions with different normalized degree of aggregation terms shown in Eq. (3.17) and (3.18) need to be used to replace Eq. (3.15).

$$J_{DSS-HFCR-u} = \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri} \frac{v_{rj}}{\sum_{q=1}^{m} v_{rq}} d_{ij} - T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri} \ln u_{ri} - T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj} \ln v_{rj}$$
$$+T_u \times T_d \left(\alpha_d \sum_{d_i, d_o \in MLS} \sum_{r=1}^{k} u_{ri} u_{ro} - \beta_d \sum_{d_i, d_o \in CLS} \sum_{r=1}^{k} u_{ri} u_{ro} \right) \quad (3.17)$$

$$J_{DSS-HFCR-v} = \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{u_{ri}}{\sum_{p=1}^{n} u_{rp}} v_{rj} d_{ij} - T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri} \ln u_{ri} - T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj} \ln v_{rj}$$
$$+T_v \times T_w \left(\alpha_w \sum_{w_j, w_o \in MLS} \sum_{r=1}^{k} v_{rj} v_{ro} - \beta_w \sum_{w_j, w_o \in CLS} \sum_{r=1}^{k} v_{rj} v_{ro} \right) \quad (3.18)$$

From the formulation of above two equations, it is clear that although both of them share similar principles with $J_{SS\text{-}HFCR}$, the original degree of aggregation of every co-cluster c is normalized by $\sum_{q=1}^{m} v_{rq}$ in $J_{DSS\text{-}HFCR\text{-}u}$, while it is normalized by $\sum_{p=1}^{n} u_{rp}$ in $J_{DSS\text{-}HFCR\text{-}v}$. By taking the normalization, we now have $(g_1)_{ij} = \sum_{r=1}^{k} u_{ri} \frac{v_{rj}}{\sum_{q=1}^{m} v_{rq}}$ and $(g_2)_{ij} = \sum_{r=1}^{k} \frac{u_{ri}}{\sum_{p=1}^{n} u_{rp}} v_{rj}$ in the two auxiliary functions, respectively. Then, we are able to get the constant value for the aggregation term as in partitioning-ranking based approaches. In other words, the bias is removed by eliminating the variation in the values of these two terms. Therefore, this normalization process is essential in the formulation in order to avoid the bias and also reduce the possibility of computational overflow.

3.3.2 Updating Rules

Similar to what we did in Section 3.2.3 for SS-FCL and SS-FCC, now we derive the co-responding updating rules for DSS-HFCR. As we got two derived auxiliary functions for u and v separately, the respective Lagrangian functions of DSS-HFCR for deriving u and v are firstly constructed as:

$$L_{DSS-HFCR-u} = \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ri} \frac{v_{rj}}{\sum_{q=1}^{m} v_{cq}} d_{ij} - T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri} \ln u_{ri} - T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj} \ln v_{rj}$$
$$+T_u \times T_d \left(\alpha_d \sum_{d_i, d_o \in MLS} \sum_{r=1}^{k} u_{ri} u_{ro} - \beta_d \sum_{d_i, d_o \in CLS} \sum_{r=1}^{k} u_{ri} u_{ro} \right) + \sum_{i=1}^{n} \lambda_i \left(\sum_{r=1}^{k} u_{ri} - 1 \right) \quad (3.19)$$

$$L_{DSS-HFCR-v} = \sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{u_{ri}}{\sum_{p=1}^{n} u_{rp}} v_{rj} d_{ij} - T_u \sum_{r=1}^{k}\sum_{i=1}^{n} u_{ri} \ln u_{ri} - T_v \sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj} \ln v_{rj}$$
$$+T_v \times T_w \left(\alpha_w \sum_{w_j, w_o \in MLS} \sum_{r=1}^{k} v_{rj} v_{ro} - \beta_w \sum_{w_j, w_o \in CLS} \sum_{r=1}^{k} v_{rj} v_{ro} \right) + \sum_{j=1}^{m} \gamma_j \left(\sum_{r=1}^{k} v_{rj} - 1 \right) \quad (3.20)$$

where λ_i and γ_j are Lagrange multipliers corresponding to the constraints in Eq.(3.2) and (3.13).

By taking the partial derivation of $L_{DSS-HFCR-u}$ with respect to u_{ri} and calculating $\partial L_{DSS-HFCR-u} \Big/ \partial u_{ri} = 0$, we have the updating rule of u_{ri} at iteration τ:

$$u_{ri}^{\tau} = \frac{\exp\left\{\frac{\sum_{j=1}^{m} v_{rj}^{\tau-1} d_{ij}}{T_u \sum_{j=1}^{m} v_{rj}^{\tau-1}} + T_d \left[\alpha_d \sum_{(d_i,d_s)\in MLS} u_{rs}^{\tau-1} - \beta_d \sum_{(d_i,d_t)\in CLS} u_{rt}^{\tau-1}\right]\right\}}{\sum_{f=1}^{k} \exp\left\{\frac{\sum_{j=1}^{m} v_{fj}^{\tau-1} d_{ij}}{T_u \sum_{j=1}^{m} v_{fj}^{\tau-1}} + T_d \left[\alpha_d \sum_{(d_i,d_s)\in MLS} u_{fs}^{\tau-1} - \beta_d \sum_{(d_i,d_t)\in CLS} u_{ft}^{\tau-1}\right]\right\}} \tag{3.21}$$

By taking the partial derivation of $L_{DSS-HFCR-v}$ with respect to v_{rj} and calculating $\partial L_{DSS-HFCR-v} \Big/ \partial v_{rj} = 0$, we have the updating rule of v_{rj} at iteration τ:

$$v_{rj}^{\tau} = \frac{\exp\left\{\frac{\sum_{i=1}^{n} u_{ri}^{\tau-1} d_{ij}}{T_v \sum_{i=1}^{n} u_{ri}^{\tau-1}} + T_w \left[\alpha_w \sum_{(w_j,w_y)\in MLS} v_{ry}^{\tau-1} - \beta_w \sum_{(w_j,w_z)\in CLS} v_{rz}^{\tau-1}\right]\right\}}{\sum_{f=1}^{k} \exp\left\{\frac{\sum_{i=1}^{n} u_{fi}^{\tau-1} d_{ij}}{T_v \sum_{i=1}^{n} u_{fi}^{\tau-1}} + T_w \left[\alpha_w \sum_{(w_j,w_y)\in MLS} v_{fy}^{\tau-1} - \beta_w \sum_{(w_j,w_z)\in CLS} v_{fz}^{\tau-1}\right]\right\}} \tag{3.22}$$

For the two simplified versions of DSS-HFCR, the updating rules for u_{ri} in SS-HFCR-W and v_{rj} in SS-HFCR-D can be both simplified by removing the supervised constraint term, accordingly.

In SS-HFCR-D, v_{rj} at iteration τ is updated by

$$v_{rj}^{\tau} = \frac{\exp\left\{\frac{\sum_{i=1}^{n} u_{ri}^{\tau-1} d_{ij}}{T_v \sum_{i=1}^{n} u_{ri}^{\tau-1}}\right\}}{\sum_{f=1}^{k} \exp\left\{\frac{\sum_{i=1}^{n} u_{fi}^{\tau-1} d_{ij}}{T_v \sum_{i=1}^{n} u_{fi}^{\tau-1}}\right\}} \tag{3.23}$$

In SS-HFCR-W, u_{ri} at iteration τ is updated by:

$$u_{ri}^{\tau} = \frac{\exp\left\{\frac{\sum_{j=1}^{m} v_{rj}^{\tau-1} d_{ij}}{T_u \sum_{j=1}^{m} v_{rj}^{\tau-1}}\right\}}{\sum_{f=1}^{k} \exp\left\{\frac{\sum_{j=1}^{m} v_{fj}^{\tau-1} d_{ij}}{T_u \sum_{j=1}^{m} v_{fj}^{\tau-1}}\right\}} \tag{3.24}$$

3.3.3 Algorithms

There are two alternative ways to carry out the DSS-HFCR Clustering. First of all, starting with a set of given pair-wise constraints on the document and the word domain, the initialized memberships for the documents and words belongs to ***ConS*** are manually adjusted to satisfy the constraints accordingly. Then, the user can decide to update **V** first using an initialized **U** or update **U** first using an initialized **V**. The initial membership values of document/words belong to ***FrS*** are still randomly assigned. At the 1st iteration, only the memberships for the documents/words belong to ***FrS*** will be updated. During the clustering process, **V** and **U** are iteratively updated with Eq. (3.22) and Eq. (3.21) respectively in an alternating manner until either the successive estimates of **U** are close enough or it reaches the maximum number of iterations. The quality of the partition in terms of the criteria defined in Eq.(3.17) and (3.18) are successively improved through reassigning the documents into *k* clusters based on the current word partition and similarly, reforming of word clusters based on the current document partition. The detailed steps of the DSS-HFCR algorithm are given in Table 3.4 and Table 3.5, respectively. For the two simplified versions of DSS-HFCR, SS-HFCR-D follows the steps given in Table 3.5 and SS-HFCR-W follows the steps given in Table 3.4. The updating rules for v_{rj} and u_{ri} need to be changed to Eq. (3.23) and (3.24), accordingly.

Time Complexity

We can see from Eq. (3.21) that the computations of the numerator and the denominator can be performed independently. For a given *r* and *j*, the computation of un-supervised part of the numerator of Eq. (3.21) requires O*(n)*. There are $k \times m$ of such numerators. Therefore, the

Table 3.4: The DSS-HFCR Algorithm: scenario 1:

Input: Dataset ***S***, number of clusters *k*, weighting factors $T_u, T_v, T_d, T_w, \alpha_d, \beta_d, \alpha_w, \beta_w$, ***MLS*** and ***CLS*** from document and word domain respectively, stopping threshold ε , and maximum iteration number $\tau_{\max}$

Output: Document membership matrix: **U**, and word membership matrix **V**.

Method:

1. Manually assign the membership v_{rj} for $j \in$ ***ConS***
2. Randomly initialize memberships v_{rj} for $j \in$ ***FrS***;
3. **REPEAT**

   3.1 if $\tau = 1$:
   For every document $i \in$ ***FrS*** , update u_{ri} with Eq. (3.21)
   else
   For every document $i \in$ ***S*** , update u_{ri} with Eq. (3.21)

   3.2 For every word *j*, update v_{rj} with Eq. (3.22);

   3.3 $\tau = \tau + 1$

UNTIL($\max_r \left| u_{ri}^{\tau+1} - u_{ri}^{\tau} \right| \le \varepsilon$)

Table 3.5: The DSS-HFCR Algorithm: scenario 2:

Input: Dataset $\boldsymbol{S}$, number of clusters k, weighting factors $T_u, T_v, T_d, T_w, \alpha_d, \beta_d, \alpha_w, \beta_w$, ***MLS*** and ***CLS*** from document and word domain respectively, stopping threshold ε , and maximum iteration number $\tau_{\max}$

Output: Document membership matrix: **U**, and word membership matrix **V**.

Method:

1. Manually assign the membership u_{rj} for $i \in$ ***ConS***
2. Randomly initialize memberships u_{rj} for $i \in$ ***FrS***;
3. **REPEAT**

3.1 if $\tau = 1$:
For every word $j \in$ ***FrS*** , update v_{rj} with Eq. (3.22)
else
For every word $j \in \boldsymbol{S}$, update v_{rj} with Eq. (3.22)
3.2 For every document i, update u_{ri} with Eq. (3.21);
3.3 $\tau = \tau + 1$

UNTIL($\max_r \left| u_{ri}^{\tau} - u_{ri}^{\tau-1} \right| \le \varepsilon$)

computations of all different numerators for one iteration require O(kmn). As for the denominator, to compute each one of them requires O*(kn)*, and there are m different denominators. Regarding about the additional computational workload due to the additional constraint terms given by the prior knowledge, we may realize that, for a given r and i, the computational workload for $\left[\sum_{(d_i, d_s) \in MLS} u_{rs} - \sum_{(d_i, d_t) \in CLS} u_{rt} \right]$ part is much less than the un-supervised part and we can simply nelgect it when computing the big O. Therefore, the time complexity of updating all document memberships per iteration is $O(kmn)$. Similarly the time complexity of updating all the word memberships per iteration is also $O(kmn)$.

Finally, taking into account the number of iterations, the time complexity of DSS-HFCR becomes $O(kmn \cdot \tau)$.

Similarly, we are able to get the time complexity for both SS-FCL and SS-FCC, which is the same as DSS-HFCR. Last but not least, the actual runtime can be much less, since that the number of nonzero entries in matrix ***D*** is considerably much smaller than mn since the matrix is typically very sparse.

3.4 Experimental Results and Discussions

In order to show the strength of the semi-supervised fuzzy co-clustering family compared to some state-of-the-art semi-supervised clustering methods, extensive experimental study has been conducted on a number of benchmark textual datasets and the results are presented and discussed in the following sections

3.4.1 Datasets

No perfect clustering approach can perform the best outcome in every single application exists. These approaches often perform differently in different domains and on different datasets. Hence, in order to conduct intensive examinations of proposed clustering methods to prove the effectiveness and efficiency on high dimensional textual data, the simulation results on 10 real world benchmark textual datasets are presented in this section. The brief categorical information of these datasets is listed in Table 3.6. The *Balance* column provides the ratio of the smallest class size to the largest class size in a particular dataset.

Most of the selected datasets are the subsets sampling from a few most widely used test collections for text categorization, such as *20newsgroups* [161], *Reuters-215783* [162], *WEBKB* [163]. They had been used extensively for testing many text classification and clustering systems in previous works, for examples [164]. They vary in content, number of topics, size, and vocabulary, creating a very diverse set of data on which clustering task is performed. "BankResearch" is another large benchmark dataset reported in [165]. Other datasets such as *k1b*, *sports*, which do not belong to the above four, they can be found in CLUTO [24], a popular on-line clustering package developed by University of Minnesota for document clustering and gene analysis.

For example, *classic* is a frequently used dataset for testing the information retrieval systems. It contains four categories, the abstracts collected from computer systems papers *CACM*, information retrieval papers *CISI*, medical journal *MEDLINE* and aeronautical systems papers *CRANFIELD*. Each set of the abstracts is considered as one of the four categories of topics. Some of the documents may appear in more than one category. For example, *Binary* consists of documents which the content is related to *politics* or *Mideast*.

Table 3.6: Brief information of the benchmark datasets

Dataset	Source	Balance	*k*	*n*	*m*	Brief Descriptions
Binary	20newsgroups	1	2	500	3776	politics.misc(500), politics.middle east(500)
Multi10	20newsgroups	1	10	500	2115	Atheism(50), Hardware(50), Forsale(50), Rec.autos(50), Hockey(50), Srypt(50), Politics(50), Electronics(50), Medical (50), Space(50)
webkb4	WEBKB	0.345	4	4199	11909	Student (1641), Project(504), faculty(1124), course(930)
CB	BankResearch	1	2	2000	4791	Commercial Bank(1000), Building Societies(1000)
SM	BankResearch	1	2	2000	5450	Soccer(1000), Motorsports(1000)
WPD6s	BankResearch	1	6	600	2660	Commercial Bank(100), C++(100), Astronomy(100), Biology(100), Soccer(100), Sport(100)
reuters3	Reuter	0.752	3	1076	2837	Trade (361), Money-fx (307), Crude (408)
sports	TREC	0.036	7	8580	18324	Baseball(3412), Hockey(809), Basketball(1410), Football(2346), Boxing(122), Bicycle(145), Golf(236),
classic	CACM/CISI CRAN/ MED	0.323	4	7089	12009	Cran(1398), Med(1033), Cacm(3203), cisi(1455)
k1b	WebACE	0.043	6	2340	13859	Health(494), Entertainment(1389), Sports(141), Politics(114), Technology(60), Business(142)

250 documents are contained in each of the two categories. It can be predicted that some overlapping manual may appear in both categories. *webkb4* is a subset of *WebKB*, a collection of 7 groups of web pages collected from computer science department of various universities. *webkb4* covers only 4 categories of topic: *student*, *faculty*, *course* and *project*. Due to the high relevance among them, it is also a very good sample dataset for testing the fuzzy clustering approaches.

3.4.2 Experimental Setting when Prior Knowledge from Document Domain Only

The experimental setting of each of the proposed clustering approaches is elaborated in this section. The scenario when a group of class labels is available is presented first, followed by the scenario when a group of random pair-wise constraints is available. The corresponding experimental settings of the compared state-of-the-art clustering approaches in both scenarios are also given in details in the corresponding sub-sections.

Scenario1: Class Label is Available

In this scenario, all the three proposed approaches can be applied, as the constraints can be always generated from the labeled documents and incorporated into SS-FCC and SS-HFCR-D,respectively. A collection of pair-wise constraints could be formed by pairing any two labeled documents and the corresponding ***MLS*** and ***CLS*** can be built accordingly based on the ground truth labels of the documents in each of the pair-wise constraints. In other words, beside SS-FCL, these two pair-wise constraint-based approaches can be also fairly compared with the existing label-based semi-supervised clustering approaches. Seeded-*kmeans* [118] developed based on Sph*kmeans* [38], on semi-supervised Fuzzy c-means (SFCM) [124] developed based on HFCM [52] and Weighted Semi-Supervised Nonnegative Matrix Factorization with Normalized Cut Weighting [130] (WNMF-NCW) developed based on NMF [67]are selected in this scenario. The amount of prior knowledge used in experiments is measured by the proportion of *n* in a particular dataset. It starts with 5%, ends with 15% in step of 5% by randomly picking up the documents from the whole dataset.

On the other aspect, in order to achieve good and reasonable clustering performance, it is very important to assign a suitable set of parameters for every dataset. T_v is set to be 1.5 in both SS-FCL and SS-FCC for all the datasets since the change on the value of T_v does not affect the results very much, as well as the original Fuzzy Codok. For SS-HFCR-D, T_v is set to 0.005 and T_d is set to 1 for all datasets. For SS-FCC and SS-HFCR-D, α_d and β_d are both set to 1. The appropriate value of T_u and T_d for each dataset is tuned through the empirical study. The finalized values used for the clustering results presented in the thesis are listed in Table 3.7. The equivalent setting is applied to SFCM and WNMF-NCW. More detailed

Table 3.7: Parameters Settings on T_u and T_d

		Binary	Multi10	webkb4	CB	SM	WPD6s	reuters3	sports	classic	k1b
Fuzzy CoDoK	T_u	1E-5	1E-3	1E-3	1E-3	1E-3	1E-3	1E-3	1E-3	1E-3	1E-3
SS-FCL	T_u	1E-3	1E-2	1E-2	1E-3	1E-3	1E-3	1E-3	1E-3	1E-2	1E-3
	T_d	0.5	10	1	4	2	2	1	1	1	1
SS-FCC	T_u	1E-3	1E-3	1E-3	1E-2	1E-2	1E-2	1E-2	1E-2	1E-2	1E-2
	T_d	0.5	10	1	6	2	2	1	1	1	1
SS-HFCR-D	T_u	1E-3	1E-2	1E-2	1E-2	1E-2	1E-3	1E-3	1E-2	1E-3	1E-2

discussions on the impact of each parameter to the clustering results will be presented in Section 3.4.8, and some general guidelines for selecting the appropriate parameter are also summarized. The stopping threshold is set to 1E-5 and τ_{max} is equal to 200.

For the initialization, the same set of label documents is given to an individual experimental trial for all the clustering approaches. The initial membership values of the labeled documents in SS-FCL, SS-FCC, SS-HFCR-D and SFCM are set to the value of l_{ri} in SS-FCL. The values of l_{ri} are then suggested and defined in Eq.(3.25), and L_{main} can be any value in the interval of (0.5,1]. In our experiments, it is fixed at 0.8.

$$l_{ri} = \begin{cases} L_{main} & \text{for the cluster } r^* \text{ represented its ground truth class} \\ \dfrac{1-L_{main}}{k-1} & \text{for the other clusters } r' \end{cases} \tag{3.25}$$

The initial membership values for the unlabeled documents are randomly assigned. For WNMF-NCW, the clustering result of spherical *kmeans* is used as the initial values. Then, ten independent experimental trials with different initial centroids or membership values will be considered as one test run, and the best trial in terms of the largest criterion function value obtained among the ten is chosen to represent that test run.

Pair-wise Constraint Available

For the second scenario, we focus on comparing SS-HFCR-D with four relevant and popular semi-supervised clustering approaches incorporating with pair-wise constraints proposed in recent years. They are Relational Non-negative Matrix Factorization for Semi-supervised clustering (SS-RNMF) [102], Penalized Matrix Factorization (PMF) [74], ONTMF-D [134] and Pair-wise Constrained Spherical *kmeans* (PC-Sph*kmeans*) [103]. ONTMF-D refers to the Orthogonal Non-negative Matrix Factorization for Semi-supervised Clustering approach (OSS-NMF) where the prior knowledge is only provided from the document domain. The direct comparisons with other clustering approaches such as information-theoretical co-clustering, semi-supervised spectral clustering which were reported in [86, 115, 118, 166, 167] are not provided in the thesis, as they were outperformed by the above four compared approaches.

For preparing the pair-wise constraints, we randomly pick up two documents from the dataset to form the pair. Then each of them is assigned a 'virtual label' which is the categorical value, and we place the constraint into either ***MLS*** or ***CLS*** based on whether they have the same or different 'virtual label', respectively. The above step is repeated many times to produce enough number of constraints. The amount of prior knowledge is measured by the ratio of the total number of constraints formed to the number of all possible combinations of document pairs (*n(n-1)/2*) for a particular dataset. The ratio increases from 0 to 5% in step of 1% for all the datasets, except *webkb4* (0-0.2%). 0% indicates the original un-supervised version of four approaches, which proposed in [38, 67, 68] respectively, are applied.

For SS-HFCR-D, we set T_u = 0.001, T_v = 0.005 and T_d = 1 for all datasets except for *SM* and *k1b*, T_v = 0.001. For PMF, clustering result obtained by spherical *kmeans* is served as the initial values. Similarly, the result of spherical *kmeans* is applied to both document and word clustering to enforce orthogonality condition on both domains for ONTMF-D. The stopping threshold $\varepsilon = 10^{-5}$ is used in all experiments. $\tau_{\max}$ is set to 200 in SS-HFCR-D and PC-Sph*kmeans*, 500 in PMF, 1000 in ONTMF-D and SS-RNMF, respectively. We can see a significantly higher $\tau_{\max}$ is set in the NMF-based approaches due to its sensitivity to the initialization, and the specific $\tau_{\max}$ is estimated by empirical study.

A conditional random initialization which should not contradict with the knowledge is applied to SS-HFCR-D. If document *a* and *b* are given the same 'virtual label' which is cluster *c*, then we manually assign a larger numeric value to u_{ca} and u_{cb} , compared to the all the other membership values assigned to the rest of the clusters. On the other aspect, if *a* and *b* are given the different 'virtual label', e.g.: *p* and *q*, the respective membership u_{pa} and u_{qb} should be given a relatively large value, while u_{pb} and u_{qa} should be as small as possible.

For SS-RNMF, a bad local minimum may be hit at a high possibility due to its sensitivity to the initialization and the selection of those pair-wise constraints as reported in [102]. Therefore, it often results in a low accuracy (usually below 60%). To avoid this problem, the authors chose the clustering result of the trial with the minimal objective value as the valid one from every three independent trials with different initial values. In this study, we follow the same way which is used for SS-RNMF in [102] closely.

All the pair-wise constraint-based approaches use the same set of constraints in each test run, which also consists of 10 independent trials given by 10 randomly initialized values. Then, the best trial in terms of the largest criterion function value obtained among the ten is chosen to represent that test run. A different set of constraints is then applied on different test runs to study the impact from the pair-wise constraints.

To give a finalized quantitative evaluation to all the non-crisp clustering algorithms in both scenarios, a de-fuzzification process that assigns every document to the cluster which

has the highest membership is required, and the results reported in this chapter are the mean of 10 different test runs.

3.4.3 Results & Discussions When Prior Knowledge is Available in the form of Class Labels of Documents

The clustering performance in *Accuracy* and *NMI* measure of 6 different approaches on 10 large benchmark datasets are shown in Table 3.8 and 3.9, respectively. The value in bold and underlined is the best result, while the value in bold only is the second best for each specific prior knowledge level.

Semi-supervised Version v.s Original Version

Firstly, the results confirm that all the semi-supervised fuzzy co-clustering approaches obviously outperform the corresponding un-supervised version across all datasets. Secondly, the performance generally keeps improving when the ratio of prior knowledge continuously increases; however, the improvement is not proportional to the amount of prior knowledge provided. Large datasets like *CB*, *webkb4*, *sports* achieved remarkable improvement with only 5% or less labels, but then the trend becomes much slower after that. The improvement is not obvious when the proportion of the labels increases from 10% to 20% if the *Accuracy* already reaches a relatively high level with only 5% labels, for example, *sports*. On the other hand, the influence caused by only 5% labels may not be strong enough to benefit those relatively small or unbalanced datasets. More labels are required for these cases. There are several possible reasons to explain such phenomena. Since the labelled documents are randomly selected from the whole dataset, these labels are not uniform distributed in all

Table 3.8: Clustering results in *Accuracy* on six label-based clustering approaches

label rate	**Dataset**	**Algorithm**						**Dataset**	**Algorithm**					
		SS-HFCR-D	SS-FCL	SS-FCC	Sd-kmeans	SFCM	WNMF-NCW		SS-HFCR-D	SS-FCL	SS-FCC	Sd-kmeans	SFCM	WNMF-NCW
0%	Binary	**81.6**		**78.2**	68.8	70.2	72.8	CB	**66.5**		61.8	**69.0**	61.9	64.4
5%		**87.4**	86.3	**87.0**	76.6	78.4	80.4		**84.1**	**83.1**	76.2	76.6	68.8	76.8
10%		**91.9**	91.2	**92.3**	79.4	82.6	84.2		**87.1**	**84.6**	80.6	78.6	70.0	78.7
15%		**93.5**	92.8	93.3	81.4	86.7	**95.6**		**90.2**	**86.1**	83.0	80.9	74.2	81.6
0%	k1b	**82.4**		72.8	77.9	70.5	**82.4**	webkb4	58.6		65.5	58.4	**70.1**	**67.6**
5%		**89.8**	87.6	88.1	85.1	79.5	**88.6**		**78.3**	**81.0**	77.5	71.2	76.4	73.6
10%		**90.7**	88.4	88.8	**90.2**	84.3	90.0		**82.4**	**82.8**	81.8	74.9	77.6	74.6
15%		**91.6**	90.1	90.3	90.3	86.4	**90.6**		**84.4**	**85.2**	83.0	74.9	80.6	78.2
0%	sports	**74.1**		72.5	69.7	**76.2**	68.4	classic	**73.6**		**68.9**	61.8	66.3	64.4
5%		86.9	**88.4**	**92.0**	75.8	85.2	81.8		**87.1**	82.4	**85.7**	71.6	76.3	73.6
10%		91.7	**92.6**	**95.3**	77.6	87.8	82.4		**90.3**	85.7	**88.9**	71.6	79.2	74.6
15%		93.0	**93.4**	**96.3**	80.3	90.3	86.3		**91.9**	87.4	**90.8**	71.6	82.5	78.2
0%	reuters3	**82.5**		**83.9**	74.7	80.1	65.2	Multi10	68.2		**72.1**	47.5	**68.8**	64.7
5%		**91.0**	89.4	**91.5**	76.2	85.4	80.6		73.9	**77.5**	**74.4**	63.6	71.2	70.3
10%		**93.3**	91.2	**92.0**	76.3	86.7	84.5		79.7	**80.7**	**79.8**	73.7	75.6	73.1
15%		**94.3**	92.3	**92.6**	76.3	88.1	88.1		81.4	**83.7**	**83.5**	76.5	76.9	75.8
0%	WPD6	**72.5**		**64.9**	56.4	59.6	60.2	SM	**81.2**		68.0	65.9	62.8	**72.1**
5%		**76.2**	63.4	**66.4**	58.0	62.4	65.7		**95.7**	83.1	**84.7**	76.5	74.6	80.5
10%		**80.3**	67.3	**69.3**	64.7	66.7	67.1		**97.5**	88.5	**88.7**	82.7	82.5	86.4
15%		**83.0**	71.7	**74.1**	66.3	66.9	72.8		**98.4**	91.4	**93.5**	88.7	85.4	89.8

Table 3.9: Clustering results in *NMI* on six label-based clustering approaches

label rate	Dataset	Algorithm						Dataset	Algorithm					
		SS-HFCR-D	SS-FCL	SS-FCC	Sd-kmeans	SFCM	WNMF-NCW		SS-HFCR-D	SS-FCL	SS-FCC	Sd-kmeans	SFCM	WNMF-NCW
0%	Binary	**32.9**	**27.8**		17.2	21.4	24.6	CB	**9.5**	5.4		**13.8**	6.0	7.0
5%		**52.7**	50.2	**49.5**	23.5	25.8	29.0		**39.6**	**32.9**	24.6	24.8	17.2	24.0
10%		**61.2**	60.4	**62.4**	26.7	36.4	42.1		**47.2**	**35.7**	30.9	26.9	21.5	26.2
15%		62.2	63.1	**64.0**	30.1	48.8	**88.2**		**50.3**	**47.4**	36.5	30.8	24.6	30.4
0%	k1b	**57.4**	52.3		54.7	49.9	**57.5**	webkb4	39.7	40.2		37.1	**41.7**	**41.0**
5%		**72.3**	70.2	70.6	68.4	54.2	**70.9**		**47.8**	**51.3**	47.2	41.8	44.2	42.6
10%		**76.0**	71.3	71.6	**71.8**	62.0	71.7		**53.2**	**53.5**	52.8	42.9	47.0	43.3
15%		**77.4**	**75.8**	75.1	75.1	68.4	75.6		**54.6**	**54.8**	53.0	43.3	51.0	46.8
0%	sports	**69.8**	68.9		65.7	**70.3**	66.5	classic	**62.0**	**60.4**		57.7	59.7	59.3
5%		77.2	**77.5**	**80.5**	68.0	76.4	71.1		**71.4**	67.8	**70.1**	64.3	65.5	64.0
10%		78.0	**81.2**	**84.4**	70.9	76.8	74.3		**73.8**	70.1	**73.0**	64.6	66.2	64.8
15%		79.4	**82.0**	**85.2**	72.2	80.1	77.6		**74.6**	71.7	**74.2**	64.9	67.8	66.9
0%	reuters3	**53.8**	**56.4**		46.4	52.0	39.7	Multi10	**59.2**	**60.1**		35.2	57.2	52.6
5%		**69.7**	68.5	**70.1**	47.1	59.8	52.1		61.4	**62.9**	**61.7**	48.1	59.4	58.8
10%		**75.0**	73.7	**74.1**	47.1	63.2	57.9		67.9	**69.8**	**69.5**	58.7	62.3	61.5
15%		**78.9**	74.8	**75.6**	47.1	65.4	65.2		69.1	**71.7**	**71.6**	62.0	62.8	62.8
0%	WPD6	**53.4**	**50.5**		43.3	76.3	88.9	SM	**32.5**	12.8		15.8	12.4	**17.2**
5%		**58.6**	49.8	**51.0**	49.5	49.6	51.0		**73.9**	35.9	**38.5**	28.5	24.4	29.8
10%		**62.9**	51.7	**54.1**	47.8	51.5	51.9		**84.5**	53.5	**54.7**	42.8	42.0	44.7
15%		**66.7**	56.6	**58.5**	51.0	51.6	57.4		**90.1**	59.7	**62.3**	54.6	52.8	56.3

clusters. For the equal balanced datasets, such as *Multi10*, although a few ground truth labels are located into each of the clusters, it may not be enough to influent the other unlabeled documents if the total number of documents in each cluster is relatively small. In this case, assigning a larger value on the weighting factor T_d may be a solution in SS-FCL/SS-FCC for a better performance.

SS-Fuzzy Co-clustering v.s Other Approaches

On the other hand, the proposed methods significantly outperform the other three semi-supervised clustering approaches on most of the datasets, except *Binary* and *k1b*. Moreover, we also realize that for some datasets e.g.: *sports, classic, reuters3*, seeded-*kmeans* may not work well no matter how many labeled documents are available to the clustering process. The reason could be the label violation issues. In seeded-*kmeans*, although a set of better initial virtual centroids can be obtained by the labeled documents, it does not guarantee that these documents to stay at the correct clusters to guide the cluster search of the rest of the documents during the clustering process. It is very likely that some of the labeled documents might go to other clusters after a few iterations, and therefore, no longer paly the guiding roles. Therefore, the clustering performance cannot be improved. Meanwhile, in the proposed fuzzy co-clustering approaches, the membership of a labeled document in its 'ground truth' cluster always holds the largest membership value among all the clusters when a suitable set of parameters is assigned. In other words, the incorporated knowledge is effectively guide the cluster search all the time. The parameter selection is discussed in details in Section 3.4.8.

Moreover, the proposed fuzzy co-clustering algorithms converge much faster than the compared algorithms, although the time complexities of the algorithms are the same. The

Table 3.10: Average number of iteration until converges

Algorithms	Average no. of iteration needed until converges									
	Binary	Multi10	k1b	CB	SM	WPD6s	webkb4	classic	sports	reuters3
SS-FCL	**12**	184	**32**	**15**	36	**64**	**12**	**58**	**54**	36
SS-FCC	14	**16**	33	46	**33**	152	48	65	57	**34**
SS-HFCR-D	**21**	**134**	46	**8**	**18**	**8**	**6**	**62**	**24**	**13**
Sd-kmeans	28	200	**25**	56	46	132	134	84	64	46
SFCM	36	200	63	32	67	168	48	136	152	67
WNMF	39	200	49	82	56	200	64	92	166	82

average number of iterations needed until the convergences by incorporating 10% labeled documents for all the 6 algorithms are shown in the Table 3.10.

3.4.4 Results & Discussions When Prior Knowledge is Available in the form of Pair-wise Constraints of Documents

This group of experimental study runs SS-HFCR-D by using pair-wise constraints generated from user assigned categorical values. Each pair-wise constraint is formed by randomly pairing two documents in the dataset. Each document in the constraint has a user-assigned virtual categorical value (label index).

The clustering performance in *Accuracy* and *NMI* measure on 8 datasets are shown in Table 3.11. Experimental results on more datasets can be found in [160]. From these two tables, some advantages of SS-HFCR-D can be summarized. Firstly, SS-HFCR-D outperforms HFCR for all the datasets. It is able to make significant improvement by quickly

Table 3.11: Clustering results on five pair-wise constraint-based clustering approaches

label rate	**Dataset**	**Accuracy**					**NMI**				
		SS-HFCR-D	PMF	SS-RNMF	OSS-NMF-D	PC-Sphkmeans	SS-HFCR-D	PMF	SS-RNMF	OSS-NMF-D	PC-Sphkmeans
1%	Binary	**100**	**100**	83.6	72.6	76.7	**100**	**100**	40.4	21.5	28.1
2%		**100**	**100**	97.3	92.3	85.3	**100**	**100**	92.4	65.7	45.2
3%		**100**	**100**	98.4	94.4	94.7	**100**	**100**	96.6	80.5	80.6
1%	CB	**100**	**100**	75.4	83.2	62.5	**100**	**100**	25.6	35.7	66.1
2%		**100**	**100**	91.3	86.7	80.3	**100**	**100**	69.4	48.6	74.1
3%		**100**	**100**	93.0	95.0	90.2	**100**	**100**	86.8	85.0	80.6
1%	SM	**100**	**100**	69.8	76.6	81.6	**100**	**100**	18.4	24.7	36.5
2%		**100**	**100**	89.5	94.8	97.5	**100**	**100**	59.0	81.2	84.8
3%		**100**	**100**	96.2	96.2	99.3	**100**	**100**	88.5	88.6	95.4
1%	WPD6	**89.4**	**70.0**	57.8	68.5	65.1	**79.6**	**65.4**	52.8	60.3	58.2
2%		**98.2**	**86.4**	66.8	70.2	67.4	**94.3**	**71.5**	58.1	65.8	59.0
3%		**100**	**91.3**	76.0	73.7	75.8	**100**	**81.1**	70.9	70.6	70.5
1%	Multi10	**72.8**	**73.2**	72.8	68.8	53.7	**65.5**	**65.3**	63.8	60.3	43.7
2%		**79.7**	**77.4**	60.4	58.9	60.0	**72.4**	**70.4**	52.3	50.6	52.6
3%		**84.9**	**82.7**	51.6	74.2	64.2	**80.0**	**78.8**	40.7	65.6	57.2
1%	reuters3	**100**	**99.2**	60.4	56.6	**75.5**	**100**	**97.8**	30.3	17.7	54.3
2%		**100**	**100**	68.2	64.9	77.7	**100**	**100**	45.8	41.6	58.6
3%		**100**	**100**	77.2	76.4	84.4	**100**	**100**	58.4	56.2	63.1
1%	k1b	**98.8**	**94.2**	73.1	69.8	67.2	**96.7**	**91.7**	58.4	53.2	49.1
2%		**100**	**97.1**	71.0	73.5	75.1	**100**	**93.0**	56.3	55.8	60.7
3%		**100**	**99.2**	75.4	64.1	80.4	**100**	**97.2**	60.1	49.8	67.2
0.05%	webkb4	**69.8**	64.9	67.7	**70.4**	69.5	**49.8**	45.2	41.5	45.1	**45.6**
0.1%		**87.3**	**77.6**	73.5	73.4	74.3	**62.2**	56.6	49.7	51.5	**57.4**
0.15%		**96.4**	78.5	75.5	**80.2**	76.7	**88.4**	**64.0**	51.8	52.0	61.7

learning from only a few constraints (less than 1%) provided for these datasets. Especially on a few relatively simple datasets such as *Binary*, *CB*, *SM*, *reuters3*, 100% accuarcy is easily achieved with less than 1% constraints provided. Therefore, more simulation results at a lower prior knowledge level is presented in Table 3.12&3.13 to show the performance improvement by increasing the number of pair-wise constraints from 50-250 in steps of 50. Secondly, although HFCR may not be the best choice when there is no available prior knowledge, SS-HFCR-D generally outperforms all other four clustering approaches on all datasets at all prior knowledge level from Table 3.11 to Table 3.13. Thirdly, SS-HFCR-D shows the consistency in achieving an improved accuracy when more pair-wise constraints are provided. However, this may not be guaranteed using the NMF-based approaches. The clustering performance of SS-RNMF and ONTMF-D on *Multi10* and *k1b* shows significant variations when more available pair-wise constraints is incorporated into the clustering process. It implies a large number of constraints sometimes may even impose certain restriction to those NMF-based clustering process. When the size of the constraint set is beyond a certain value, the quality of SS-RNMF fluctuates. Fourthly, SS-HFCR-D shows the advantage in terms of computational speed. It usually converges within 100 iterations, which is much faster than the three NMF-based approaches .

3.4.5 Results & Discussions of Word Cluster Representation for the Partitioning-Ranking based Approaches

As a partitioning-ranking based fuzzy co-clustering approach, we also obtain a group of word ranking clusters simultaneously with the document clusters at the end of the clustering process. However, for these word clusters, there is no ground truth available to compute

Table 3.12: Performance comparison between SS-HFCR-D and PMF in *Accuracy*

No.of pairwise constraints	Binary		CB		SM		reuters3	
	SS-HFCR	PMF	SS-HFCR	PMF	SS-HFCR	PMF	SS-HFCR	PMF
50	**87.1**	62.1	**67.8**	63.4	**82.8**	67.5	74.0	**75.5**
100	**93.2**	65.4	**67.9**	62.2	**85.3**	78.6	75.6	**75.7**
150	**93.9**	72.4	**69.5**	63.9	**86.7**	79.1	**77.4**	75.8
200	**95.6**	80.4	**70.0**	64.3	**87.5**	83.6	**78.2**	76.1
250	**96.2**	81.8	**71.1**	64.8	**89.4**	83.9	**78.2**	76.3

Table 3.13: Performance comparison between SS-HFCR-D and PMF in *NMI*

No.of pairwise constraints	Binary		CB		SM		reuters3	
	SS-HFCR	PMF	SS-HFCR	PMF	SS-HFCR	PMF	SS-HFCR	PMF
50	**43.2**	14.3	**17.1**	14.2	**31.4**	17.0	53.8	**54.6**
100	**62.2**	17.2	**18.0**	14.2	**37.5**	26.9	54.9	**54.8**
150	**72.4**	20.1	**18.7**	13.6	**40.2**	30.0	**56.2**	55.0
200	**87.2**	30.9	**23.9**	13.8	**43.8**	32.6	**56.8**	55.0
250	**88.9**	32.7	**25.0**	15.2	**54.8**	35.0	**57.2**	56.0

Accuracy or *NMI*. Here, we use the clustering results of SS-FCC on three overlapping datasets *Binary, CB* and *webkb4*; selected the "top" 10 words with the highest ranking membership values from every word cluster, and listed them in Table 3.14. Each of the word clusters is associated with a particular document cluster with a meaningful topic. The ranking value of each word in the cluster indicates the importance of that word to the respective topic. The higher the ranking value is, the more important the particular word is to the topic. These keywords with top "impact" can be exactly used to represent the underlying "concept" of the corresponding document clusters, and they are useful as the good input in other text mining techniques, such as text summarization. In other words, it is another important benefit or strength of using the partitioning-ranking based fuzzy co-clustering to categorize the overlapping text corpus in real world applications.

Furthermore, we could see that when some of the representative keywords (highlighted as the *italic* words) are actually shared in more than one cluster, e.g. the two topics in *Binary*, it implies the topics of different categories in a dataset are relevant. The number of shared keywords may be used to show the relevancy of two or more overlapped topics. On the other hand, the distinct keywords shown in different word clusters (highlighted as the underlined words) could be used to distinguish the different topics for documents, especially when the all the topics in a dataset are closely related to each other, e.g.: *student* in *Student* cluster, *professor* in *Faculty* cluster in *webkb4*.

Table 3.14: Word clusters: top ten words for each cluster

Binary		**CB**		**webkb4**			
Politics. misc	***Politics. middleeast***	***Commercial Bank***	***Building Society***	***Student***	***Project***	***Faculty***	***Course***
jews	*israel*	nbsp	*mortgage*	home	*systems*	*research*	homework
israel	*jews*	*mortgage*	nbsp	page	*research*	*systems*	class
government	israeli	document	*pound*	*research*	project	parallel	office
people	arab	nav	society	student	group	professor	programming
gun	*war*	*account*	*account*	university	programming	university	assignment
children	jewish	*pound*	rate	Austin	parallel	science	assignments
fbi	turkish	menu	interest	*systems*	page	computing	*systems*
cramer	mr	var	*loan*	science	home	software	due
state	armenians	*loan*	isa	department	information	engineering	hours
war	muslin	banking	fee	thu	software	home	lecture

3.4.6 Special Experimental Settings for DSS-HFCR on Selected Datasets

In this section, we demonstrate the effectiveness of DSS-HFCR by observing the expected performance improvement on the top of SS-HFCR-D by further incorporating the prior knowledge from word domain. The performance is compared with OSS-NMF [134]. A few semi-supervised co-clustering approaches with label information or pairwise constraints [126, 127] have not been compared due to high time complexity issue. As we can see from Table

3.11-3.13, some of the datasets have already achieved very good performance with very limited prior knowledge from the document domain. Therefore, we further reduce the amount of prior knowledge from document domain by limiting the maximum number of random pair-wise constraints to 250.

Preparing Pairwise Constraints on Words by Human Judgment

In this approach, the way of preparing the prior knowledge in the word domain may be considered as the most significant difference from the existing related works in literature. We would like to firstly give a brief review on how the lexical knowledge in word domain is obtained in the existing approaches. First of all, the user needs to decide the size of the words. The word selection is done by some pre-processing tools, e.g.: Mallet toolkit; with information gain criteria [168] or term frequency criteria [137]. Then, using the *ACM Keywords Taxonomy*, the category information for words is obtained (a similar format with the 'ground truth' for documents), and used as the prior knowledge. Although preparing certain available knowledge in word domain by some machine learning techniques is reasonable and less time cost, we may see there are still some drawbacks. First, those on line taxonomy system may not be able to cover the knowledge from all fields. For example, *ACM Keywords Taxonomy* is restricted in dealing with the web content in computer science and the related fields. Other than that, some web pages like music blogs, sports news are out of its scope. Secondly, similar with the document domain, the available prior knowledge in the word domain must be assumed to be 100% correct. However, as reported in [137], the prior knowledge obtained purely from machine learning may be not very accurate. To be more specific to the fuzzy co-clustering technique, the motivation is that highly related documents and words (as indicated by high d_{ij} values) should be intended to be co-clustered together (i.e. assigned to the same co-cluster) by maximizing the aggregation term $\sum_{r=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}u_{ri}v_{rj}d_{ij}$. One word cluster is co-responding to one document cluster. Hence, it is possible that two words should be clustered together if they appear together frequently in a few documents, even though they are automatically partitioned into different categories by a particular taxonomy system purely based on the semantic meaning. In contrast, selecting the keyword by human judge can also help to collect certain keywords which have significant semantic contribution to the document clustering process but they may be ignored under the fuzzy co-clustering framework if they only appear in a low term frequency. Thirdly, an accurate number of the word cluster must be known in prior to get such category information on the vocabulary set.

From the above discussion, we can see some of the potential advantages of using human judgment rather than machine learning for preparing the lexical knowledge. In this way, the criterion of selecting the keywords does not simply rely on either their semantic meaning or

the term frequencies. We firstly pick up a few representative keywords from the dataset, and named the word set as "***ARK***". Then, the pairwise constraints are generated by pairing any two words from "***ARK***". In addition, the initial word labels are then given to the keywords by assigning a user-defined initial membership value of each of the keywords to all the clusters. This value is usually set to 1 in the cluster which corresponds to its categorical label and 0 to all the other clusters if it appears only in single cluster. If it appears in $\bar{k}$ clusters where $1 < \bar{k} < k$, an average value $1/\bar{k}$ will be assigned to the respective clusters.

Based on the instructions given above, we study the five datasets: *Binary, CB, SM, WPD6, webkb4* carefully to obtain as much prior knowledge as we can from word domain, before conduct the experimental study.The number of labeled keywords in these five datasets based on human judgment is listed in Table 3.15. It is usually only a small fraction of the total number of the words *m*.

Parameter Settings

For all the five datasets, we keep the values of T_u and T_v given in Section 3.4.4, set T_d to 1, T_w to 0.5 as we believe that a better clustering result tends to rely on the prior knowledge from document domain a bit more since the pair-wise constraint from the document domain is always referred to the ground truth, while human judgment on words may be still bias. The value of α_d, β_d, α_w, β_w are all set to 1.

Table 3.15: Keywords Information in each dataset

Dataset	CB	Binary	SM	webkb4	WPD6
No. of labeled keywords	120	56	100	146	132
Total No. of words	4791	3376	5450	11909	2660
Fraction of keywords labeled	2.5%	1.6%	1.8%	1.2%	5.0%

3.4.7 Results & discussions for DSS-HFCR

Performance Comparison

The clustering performance in *Accuracy* and *NMI* measure for both approaches is shown in Table 3.16. Two main observations can be made. Firstly, the clustering results of DSS-HFCR are significantly improved by effectively combining the prior knowledge from both word and document domain. Secondly, DSS-HFCR outperforms OSS-NMF on four datasets, and achieves comparable results on *webkb4*.

Impact of the Word Knowledge

We also conduct the experiments to investigate the impact of using different number of pair-wise constraint from the word domain to the clustering performance. Suppose the user already prepares *m*' representative keywords from the dataset, there are 2 alternative ways to construct the ***MLS*** and ***CLS*** for the word pairs. One way is to pair up any two words in *'**ARK**'*

Table 3.16: Results of DSS-HFCR and OSS-NMF

Algorithm	Accuracy					NMI				
	CB	Binary	SM	webkb4	WPD6	CB	Binary	SM	webkb4	WPD6
ONTMF	58.5	66.0	67.0	58.3	62.3	2.1	8.6	9.2	36.6	49.3
ONTMF-W	62.9	72.6	72.8	61.4	66.7	5.4	17.2	22.0	43.6	54.6
ONTMF-D	59.1	66.2	69.1	60.8	62.9	1.7	8.8	13.9	42.1	51.6
OSS-NMF	68.2	76.0	78.3	**64.5**	68.8	11.3	20.5	26.8	**47.2**	58.4
HFCR	67.1	81.2	81.2	58.6	72.5	9.5	12.2	29.6	33.5	53.4
SS-HFCR-W	72.2	84.3	83.2	61.4	75.9	21.9	24.5	44.3	38.6	62.1
SS-HFCR-D	71.1	95.2	88.8	58.7	75.4	15.6	73.4	52.7	34.0	60.0
DSS-HFCR	**79.2**	**96.5**	**93.4**	63.8	**82.4**	**29.5**	**78.1**	**84.5**	44.1	**72.7**

and then randomly pick up a few from the total $C_{m'}^2$ pairs; the other way is to randomly pick up a subset of the words from '***ARK***' and generate a complete pair-wise relationship map among the words in this subset. In this chapter, the first way is applied.

Firstly, we test on SS-HFCR-W by only incorporating the pair-wise constraints from the word domain. Figure 3.1(a) and (b) show the results on the five datasets in *Accuracy* and *NMI* measure, respectively. The clustering performance generally continuously improved along with the increment of the number of pair-wise constraints. Meanwhile, a great variation is showed with the increment of the pair-wise constraint from the word domain for OSS-NMF. Secondly, we further add up the prior knowledge in terms of pair-wise constraints from the document domain. Figure 3.1(c) and (d) shows the experimental results with 100 constrained

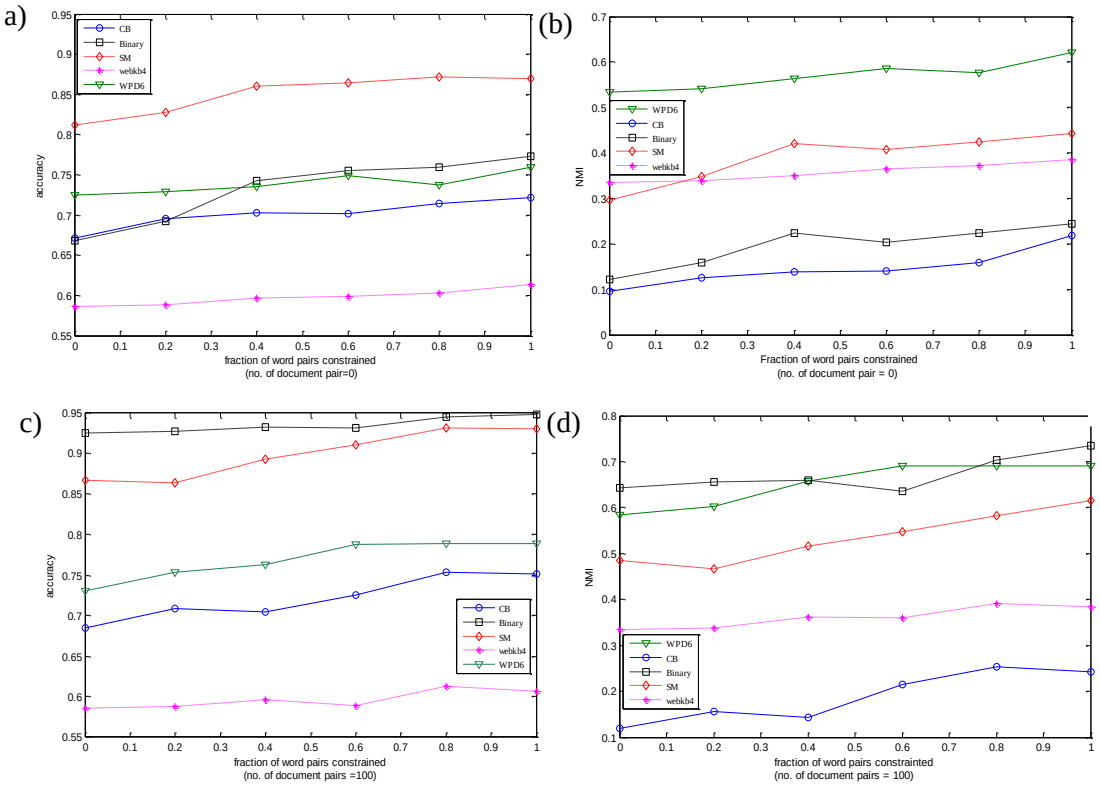

Figure 3.1: Performance of DSS-HFCR with different number of pairwise word relations

document pairs. Comparing Figure 3.1(c) with 3.1(a), Figure 3.1(d) with 3.1(b), we find out that the improvement becomes more significant on most of the datasets with the help of the prior knowledge from the document domain. Moreover, the performance improvement is not accumulated by simply adding up the contribution from each single domain. This also proves that the prior knowledge on one domain can help to get a better cluster representation on the other domain iteratively.

3.4.8 General Guidelines for Parameter Tuning

In literature, parameter tuning is not always straightforward and can be tedious in many semi-supervised clustering/learning techniques. Unfortunately, a set of suitable parameters is also essential for the fuzzy co-clustering approaches to achieve a good performance. In this section, we aim to find out some useful guidelines of the parameter selection by studying the physical meaning and impact of different parameters in SS-FCL and SS-FCC in details. We believe in principle these guidelines are also applicable to SS-HFCR and DSS-HFCR. As we discussed in the previous sections, one of the basic criteria we follow to find the suitable parameters is that the final membership distribution of the clustering process should not violate the initial label indication or the pair-wise constraints.

Referring back to the objective function of SS-FCL and SS-FCC given in Eq.(3.5) and (3.6), T_u controls the magnitude of the un-supervised term (degree of aggregation) since we fix T_v , while T_d controls the magnitude of supervised constraint term. The values of T_u and T_d need to be adjusted manually to provide a balance between supervised and un-supervised terms in the objective function to satisfy all the prior knowledge during the clustering process for a better performance in the end. In other words, if the prior knowledge is given in terms of class labels in SS-FCL and SS-FCC, for each labeled document *i*, its final membership value u_{r^*i} should be assigned with the largest to the correct cluster label *r**, than all the membership values to the other clusters.

The guideline of choosing the two most important parameters T_u and T_d for a particular dataset is provided after explaining the role of each of them in the equation.

$\mathbf{T_u}$

Through our observation, we found that better performance is always achieved when T_u is set to a larger value in SS-FCL and SS-FCC than that in Fuzzy CoDoK for most of the datasets. Here, we use the updating rules of SS-FCC for illustration purpose, and the same principle should work on SS-FCL as well. The reason can be explained as follows: Firstly, the aggregation term $\sum_{j=1}^{m} v_{rj}\, d_{ij} - \frac{1}{k}\sum_{l=1}^{k}\sum_{j=1}^{m} v_{rj}\, d_{ij}$ (refer to Eq. (3.10)) for a particular document *i* may produce a positive or negative membership value to a cluster *r*. This term calculated for *r** may be smaller than the term calculated for any *r'*. Secondly, the supervised term $\left(\sum_{f=1}^{k}\sum_{(d_i,d_s)\in mls} u_{fs} - \sum_{l=1}^{k}\sum_{(d_i,d_t)\in cls} u_{ft}\right)$ based on the label set is designed to lead

the search by increasing the membership value u_{r*i} to the correct cluster r* while decreasing the membership value to other incorrect clusters for each labeled document i. However, when a smaller T_u value is selected, the unsupervised term $\frac{1}{2T_u}\left(\sum_{j=1}^{m} v_{rj}\, d_{ij} - \frac{1}{k}\sum_{r=1}^{k}\sum_{j=1}^{m} v_{rj}\, d_{ij}\right)$ may result in a much larger value to an incorrect cluster r' comparing with the value to the correct cluster r*. In this case, compared with the unsupervised term, the supervised term may become insignificant to play its role as it is defined in the objective function. It is noted, once a labeled document is missing during the clustering process, the search of other unlabeled documents may be badly biased. Therefore, a larger T_u may help to balance the weight of both terms in the clustering process. Throughout our experimental study, T_u is set to 1 as the base value, and it is fine-tuned by multiplying 10 to the power of -1 each time to search for the suitable value for each dataset.

T_d

Once the value of T_u and T_v are both decided, the value of T_d is the key factor to eliminate the label violation effect during the clustering process. Increasing the value of T_u directly raises the impact of supervision given by the same amount of knowledge. The purpose is also making sure the supervised term $T_d \bullet l_{ri}$ is significantly large enough compared with the unsupervised term to guide the cluster search correctly. In Table 3.17, we show how the label violation phenomena can be gradually reduced by increasing T_d to a suitable value on some of the datasets, while it is unachievable in seeded-*kmeans*, and not guaranteed in SFCM and WNMF-NCW.

For SS-FCC, assuming α_d and β_d are both equal to 1, the magnitude of the supervised constraint term is proportional to the number of labeled documents which are used to form the constraints. Therefore, rising the percentage of labeled documents significantly increases the weight of supervised term; hence generally creates a bigger influence on the clustering process. Other than that, the magnitude of constraint term is also linearly proportional to the average size of each cluster (n/k). In other words, it is related to the total number of documents and the number of clusters. This can be used to explain when the proportion of the labeled documents in use is the same for two different datasets, e.g. 5%, the larger dataset may be obviously benefited from the prior knowledge, e.g. *CB, sports* while the small dataset shows almost no significant improvement e.g. *Multi10 and WPD6*.

Table 3.17: Fraction of label violation by different value on T_d

Datasets	**Value of T_d**				
	1	***2***	***3***	***4***	***5***
Multi10	0.142	0.067	0.039	0.015	0.004
CB	0.05	0.045	0.038	0.014	0.01
WPD6	0.13	0.08	0.03	0.02	0
sports	0.006	0.004	0	0	0
classic	0.004	0.003	0.001	0	0

Therefore, we may further balance the weight of the supervised term by fine tuning T_d after T_u and T_v are fixed. In the latter case, it is helpful to give a larger T_d value to increase the weight on the supervised term, and therefore make further improvement at a given percentage of prior knowledge level. The maximum value of T_d is empirical set to 10 as a reference in our experimental study. The reason is that this improvement in the performance made by enlarging T_d could be also limited. Giving a large value to T_d may impose a kind of restriction to the clustering on other unlabeled documents. We generally set a relatively large T_d to a dataset which has a smaller class size and set a relatively small T_d to a datasets which has a larger class size. By default, T_d is set to 1 if there is no big difference on the performance by tuning T_d.

3.5 Conclusion and Future Work

In this chapter, we presented three novel semi-supervised clustering approaches under the fuzzy co-clustering framework for categorizing large scaled, sparse, high dimensional data. SS-FCL and SS-FCC are the partitioning-ranking based approaches developed based on Fuzzy Codok. Two types of prior knowledge: class labels and pair-wise constraints from the single document domain are incorporated into these two approaches, respectively. The output of these two clustering approaches also includes a group of work ranking clusters, which are useful for other data mining techniques other than clustering, such as text summarization Meanwhile, a heuristic dual-partitioning semi-supervised fuzzy co-clustering approach called DSS-HFCR is also proposed in order to make full use of the available prior knowledge from both document and word domain. Extensive experimental study shows better accuracies and time efficiency can be achieved by the proposed methods on a number of benchmark datasets, with a few popular label-based/constrained-based semi-supervised clustering approaches.

We also see the potential of a high order co-clustering approach [75] incorporated with some prior knowledge to conduct more complicated heterogeneous data analysis. In the future, we may also explore innovative approaches which have less computational complexity for clustering data in a heterogeneous network.

Chapter 4

Semi-Supervised Clustering with Multi-Viewpoint based Similarity Measure

4.1 Overview

In Chapter 3, we focus on developing novel semi-supervised clustering approaches under the fuzzy co-clustering framework. Co-clustering is a typical type of clustering methods that does not take any explicit similarity measure among the data. Meanwhile, in many other types of clustering methods, a well-defined similarity measure is essential to discover the true underlying structure of data. Especially, when the clustering task is described as an optimization problem, the optimal partition of the data documents is usually found by optimizing a particular criterion function on the sum of certain explicit similarity among the data. Therefore, the similarity measure does play a very important role for the effectiveness of the methods, other than the design of the clustering algorithm. In this chapter, we turn our focus on developing more effective and feasible similarity measures with the help of a small amount of prior knowledge, and applying them to new clustering approaches for the categorization of high dimensional textual documents.

While the similarity between two documents is measured by using only one reference point, which is the origin in most traditional ways, we believe a more accurate and feasible similarity measure could be conducted by using a set of meaningful reference points, called viewpoints, coming from the dataset itself, as in principle it would carry more useful information to describe the proximity among the documents. Nguyen et al [1] proposed a novel multi-viewpoint based similarity (MVS) measure in 2011, which utilizes many different viewpoints at the same time to assess the similarity between two documents in a sparse and high-dimensional feature space. In the MVS, each document assumed not belonging to the same cluster as the two documents being measured is treated as a viewpoint. Two criterion functions based on the MVS are formulated for clustering purpose (namely MVSC). The capability of MVSC for achieving better performance than a series of single viewpoint-based similarity (SVS) clustering approaches in terms of accuracy and computational speed is also demonstrated over a number of large benchmark datasets.

In this chapter, a new semi-supervised MVS-based clustering framework, namely SS-MVSC, has been investigated to see how clustering approaches can be developed by directly incorporating two types of prior knowledge using the MVS measure. Different from the conventional similarity-adapting based approaches which are often restricted to improve the similarity measure through either an automatic [108] or active distance metric learning process (DML) [169], we now look for the alterative solution with the help of MVS concept. In other words, the purpose of this study is to find the appropriate ways of utilizing the prior knowledge to formulate a more effective measure in the MVS manner which can be immediately applied to the clustering and enhance the overall performance; rather than making use of knowledge to improve the similarity measure through a separated DML process before the actual clustering process is carried out. With the new framework, we also aim to overcome a few potential limitations in original MVSC as stated in [1] and make the new clustering approaches more efficient and feasible by adopting the strength of both search-guiding and similarity-adapting based semi-supervised clustering in one step.

The main contribution of this work includes the followings: two semi-supervised clustering approaches have been proposed, which the similarity between two document vectors is assessed from multiple appropriate viewpoints with the help of a small number of labeled documents (LMVS measure) or pair-wise *cannot-link* constraints (PMVS measure) in the dataset during the clustering process, namely LMVS Clustering and PMVS Clustering, respectively. Two new clustering criterion functions are formulated from each of the similarity measures, accordingly. The effectiveness of LMVS and PMVS measure is shown through some validity tests on the relevant similarity matrices. Then, theoretical analysis are conducted systematically, to explain how the knowledge is used for similarity measure enhancement, in addition to search-guiding purpose; and why the misleading effects caused by some inappropriate viewpoints in the original MVSC [1] can be successfully addressed. The algorithms for optimizing LMVS and PMVS Clustering are presented afterwards. The common key points of both algorithms are discussed and the key difference between them is also elaborated. We finally compare the performance of LMVS&PMVS Clustering with other baseline and state of the art semi-supervised clustering/learning methods on a number of benchmark textual datasets.

4.2 Related Works

First of all, Table 4.1 summarizes the basic notations that will be used throughout this chapter. Each document in a text corpus $\boldsymbol{D}$ corresponds to an m-dimensional normalized vector d, where m is total number of terms (words) that the document corpus has.

Table 4.1: Notations for MVS-based Clustering

Notation	Description
$\boldsymbol{D}$	matrix representation of the dataset
$\boldsymbol{P}$	partition matrix
CS	cosine similarity
n	total number of documents
n_r	number of documents in cluster r
n_{cs} / n_{fs}	number of documents with/without constraints
g	number of labeled documents of the collection
g_r	number of labeled documents in cluster r
w_r	number of viewpoints for cluster r in PMVS
m / c / k	number of terms / classes / clusters
d	document (refer to document vector) , $\|d\|=1$
$\boldsymbol{S} = \{d_1, \ldots, d_n\}$	set of all the documents
S_r	set of documents in cluster r
$\boldsymbol{LS}$	set of all labeled documents
LS_r	set of labeled documents in cluster r
VS_r	set of *cannot-link* constraint-based viewpoints for cluster r
$\boldsymbol{ConS}$	set of documents with pair-wise constraints
$\boldsymbol{FrS}$	set of documents without pair-wise constraints
$SS= \{c_1,\ldots,c_k\}$	set of the seeds
$D = \sum_{d_i \in S} d_i$	composite vector of all the documents
$D_r= \sum_{d_i \in S_r} d_i$	composite vector of documents in cluster r
$L = \sum_{l_i \in LS} l_i$	composite vector of all the labeled documents
$L_r = \sum_{l_i \in LS_r} l_i$	composite vector of labeled documents in cluster r
V_r	composite vector of *cannot-link* constraint-based viewpoints for cluster r
$C_r=D_r/ n_r$	centroid vector of cluster r
$\boldsymbol{MLS/CLS}$	set of all *must-link*/*cannot-link* constraints
ml_i	number of documents must link to d_i
MS_i	set of documents must link to d_i
M_i	composite vector of documents *must link* to d_i
SOM	set of missed documents
n_m	number of the missed documents

4.2.1 Single Viewpoint-based Similarity

Before the existing Multi-Viewpoint based Similarity is reviewed, a few most popular traditional distance measures using single viewpoint are briefly covered. In the literature, Euclidean distance given in Eq. (4.1) and cosine similarity (CS) given in Eq.(4.2) are two popular measures. The former one is widely applied to many areas, such as sensor network localization [170]; while the latter one is used in spherical *kmeans* [38] which is shown in Eq.(4.3), to handle those data represented in a sparse and high dimensional space, such as text documents. The Euclidean distance between documents to its cluster center should be minimized, while the cosine similarity between them should be maximized. CS is widely applied in many other document clustering methods as a core similarity measurement, such as the graph-based clustering: Normalized Cut [40] or Min-Max cut [65]. Other popular measures like the extended Jaccard coefficient [24] combines the feature of CS and Euclidean distance. Pelillo argued that the symmetry and non-negativity assumption of the similarity measure in text clustering was actually a limitation. While cosine similarity keeps value non-negative, Pearson Correlation Coefficient is special measure allows the similarity ranges from

+1 to -1. Moreover, Kullback-Leibler divergence is a good example of non-symmetric measure, which is a widely applied measure for evaluating the difference between to probability distributions in information theory-based clustering. In general, CS is the most popular one because of its simple interpretation. It is also the base of the MVS measure.

$$Dist\left(d_i, d_j\right) = \left\|d_i - d_j\right\| \quad . \tag{4.1}$$

$$Sim\left(d_i, d_j\right) = \cos\left(d_i, d_j\right) = d_i^t d_j \quad . \tag{4.2}$$

$$\max \sum_{r=1}^{k} \sum_{d_i \in S_r} \frac{d_i^t C_r}{\left\|C_r\right\|} \quad . \tag{4.3}$$

4.2.2 The Existing MVS

As pointed out in [1], the CS shown in (4.2) can be understood by calculating the cosine angle between two normalized document vectors as measuring them at the origin i.e. vector 0. Hence, it is a single viewpoint-based measure. The motivation of MVS stands that it is possible to obtain a more informative assessment of how close or distant a pair of documents (d_i and d_j) is, if we could measure them by standing at more than just one viewpoint as reference. For example, from a third point d_h, the direction and distances to d_i and d_j are indicated by two new vectors ($d_i - d_h$) and ($d_i - d_h$) respectively. Therefore, working on different distance vectors by standing at a number of different viewpoints, the similarity between two documents d_i and d_j which are located in the same cluster r, is defined as the average of similarities measured relatively from the views of all the other documents outside cluster r. In other words, the third point d_h which works as the viewpoint to establish this measurement must be outside of the cluster r. Therefore, the final form of MVS can be derived as below:

$$\begin{aligned}
\underset{d_i, d_j \in S_r}{MVS}\left(d_i, d_j\right) &= \frac{1}{n - n_r} \sum_{d_h \in S \setminus S_r} Sim\left(d_i - d_h, d_j - d_h\right) \\
&= \frac{1}{n - n_r} \sum_{d_h \in S \setminus S_r} \left(d_i - d_h\right)^t \left(d_j - d_h\right) \\
&= \frac{1}{n - n_r} \sum_{d_h \in S \setminus S_r} \cos\left(d_i - d_h, d_j - d_h\right) \left\|d_i - d_h\right\| \left\|d_j - d_h\right\| \\
&= \frac{1}{n - n_r} \sum_{d_h \in S \setminus S_r} \left(d_i^t d_j - d_i^t d_h - d_j^t d_h + d_h^t d_h\right)
\end{aligned} \tag{4.4}$$

In Eq.(4.4), each individual relative similarity is defined by dot-product between two different vectors ($d_i - d_h$) and ($d_i - d_h$). It is equivalent to the product of cosine angle between d_i and d_j by viewing from d_h, and the Euclidean distance product from d_h to d_i and d_j.

Therefore, the MVS measure does not only reflect the intra-similarity between d_i and d_j based on d_h , but also provided a measure of inter-similarity between d_i / d_j and d_h by the Euclidean distances since in principle d_h is assumed not belong to cluster r.

4.2.3 Clustering Criterion Functions based on MVS

Two clustering criterion functions were formulated based on this MVS measure. The first one, called I_R, is the cluster size-weighted sum of average pairwise similarities of documents in the same cluster. The sum can be expressed as:

$$F = \sum_{r=1}^{k} n_r \left[\frac{1}{n_r^2} \sum_{d_i, d_j \in S_r} MVS\left(d_i, d_j\right) \right], \tag{4.5}$$

According to the last expression of $MVS(d_i, d_j | d_i, d_j \in S_r)$ in Eq.(4.4), we have:

$$F = \sum_{r=1}^{k} \frac{1}{n_r} \left[\sum_{d_i, d_j \in S_r} d_i^t d_j - \frac{2n_r}{n - n_r} \sum_{d_i \in S_r} d_i^t \sum_{d_h \in S \setminus S_r} d_h + n_r^2 \right]$$

$$= \sum_{r=1}^{k} \frac{1}{n_r} \left[D_r^t D_r - \frac{2n_r}{n - n_r} D_r^t \left(D - D_r\right) + n_r^2 \right] = \sum_{r=1}^{k} \frac{1}{n_r} \left[\frac{n + n_r}{n - n_r} \|D_r\|^2 - \left(\frac{n + n_r}{n - n_r} - 1 \right) D_r^t D \right] + n$$

It is noted this formulation is expected to be quite sensitive to cluster size n_r without the help of a regulating factor α. [1] In addition, by removing the constant n, the final form of I_R can be expressed as below:

$$I_R = \sum_{r=1}^{k} \frac{1}{n_r^{1-\alpha}} \left[\frac{n + n_r}{n - n_r} \|D_r\|^2 - \left(\frac{n + n_r}{n - n_r} - 1 \right) D_r^t D \right]. \tag{4.6}$$

In this formulation with α, a better cluster quality is able to obtain by a higher I_R value, although it may be still sensitive to the cluster size. The second criterion function I_V, which considers similarity between each document vector and its cluster's centroid instead may prevent this problem. It is expressed in criterion function G as below:

$$G = \sum_{r=1}^{k} \sum_{d_i \in S_r} \frac{1}{n - n_r} \sum_{d_h \in S \setminus S_r} Sim\left(d_i - d_h, \frac{C_r}{\|C_r\|} - d_h \right), \tag{4.7}$$

Similar to I_R, the final formulation of I_V given in (4.8) can be derived by exploring the vector dot product in a few steps.

$$I_V = \sum_{r=1}^{k} \left[\frac{n + \|D_r\|}{n - n_r} \|D_r\| - \left(\frac{n + \|D_r\|}{n - n_r} - 1 \right) \frac{D_r^t D}{\|D_r\|} \right]. \tag{4.8}$$

From Eq.(4.6) and (4.8), we realized that, unlike only one intra-cluster similarity term given in the criterion function of spherical *kmeans*, the optimization on I_R or I_V is actually achieved by calculating the weighted difference between an intra-cluster similarity term and

an inter-cluster similarity term. The former is presented by $\|D_r\|^2$ in I_R or $\|D_r\|$ in I_V, respectively; while the latter is presented by $D_r^t D$ in I_R or $D_r^t D/\|D_r\|$ in I_V. Therefore, it is the additional inter-cluster similarity term plays the key role to help the MVS measure becoming more informative than CS.

4.2.4 Weakness in Existing MVS

With Eq.(4.6) and (4.8), the optimal cluster assignment is achieved by two steps: Initialization and Refinement. During each iteration of Refinement, the documents are visited one by one in a random order. Each document is either moved to the cluster that leads to the highest improvement in terms of the values of the respective criterion functions I_R or I_V, or stays in the current cluster if no improvement can be achieved.

As the MVS measure combining the feature of Euclidean distance and cosine similarity together, it has been briefly argued in [1], while most of viewpoints are assumed useful, there may be some of the "inappropriate" ones giving misleading information during the clustering process. It suggests that a large number of viewpoints are usually required to balance and overcome the misleading effect of these viewpoints; to make MVS be a more informative measure than the CS measure. In this section, we try to understand where these "inappropriate" viewpoints come from, how they mislead the clustering process, and therefore look for the available solutions accordingly.

Firstly, according to Section 4.2.2, we consider a document which is located "far away" from both documents been measured in cluster *r* as a real outsider of cluster *r* is the appropriate viewpoint for these two documents. However, it is noted that MVSC is also sensitive to the initialization. During the clustering process, a viewpoint d_h for measuring the MVS between d_i and d_j in the cluster *r*, may be in fact belonging to cluster *r*, too. In this case, two small Euclidean distances $\|d_i - d_h\|$ and $\|d_j - d_h\|$ are expected, and the relatively measure based on d_h also becomes small to reflect this potential. Finally, this small similarity value may lead to a wrong move decision. Therefore, a document that is in fact belonging to the same cluster *r* with the two documents been measured, always serves as an inappropriate viewpoint when it is temporal outside cluster *r* during the clustering process. Detailed analysis based on validity test is also given in Section 4.4 to help us further looking into the impact of these "inappropriate" viewpoints.

Secondly, the cluster centers are implicitly updated immediately with the move of any documents in MVSC, and the set of viewpoints for each of the clusters are also dynamic updated. Therefore, unlike *kmeans*, the MVS of a document may be not measured based on the same set of viewpoints, if it is placed in a different position in the visiting order, and therefore lead to a different cluster assignment eventually. We believe this is also the potential weakness

when the existing MVS is applied to clustering, i.e.: the performance of MVSC may be also sensitive to the visiting order of the documents, other than the initialization.

Based on the critical review of MVS, we conclude that, it would be helpful to make the MVS measure be independent of the visiting order of documents, even less sensitive to the initialization; if we are able to get a few appropriate viewpoints for all the documents when their pair-wise similarities are measured, no matter which cluster they are placed during the clustering process. In the next section, we aim to present how to achieve this goal with the help of some available prior knowledge to users, and develop some new semi-supervised MVS-based clustering approaches. To highlight our motivation again, unlike the existing semi-supervised clustering approaches, the semi-supervised MVS Clustering proposed in this chapter does not fall into either the conventional search-guiding or similarity-adapting strategy. It is not a simple combination of the two, either; but carries the equivalent strengths of both strategies.

4.3 Semi-Supervised MVS Clustering Framework

4.3.1 MVS with Labeling Information

Through the review and discussion in Section 4.2, it is clear that the key factor to develop an effective MVS-based clustering is to get a relatively stable and appropriate viewpoint set for every cluster and every document in the dataset through the whole Refinement process. However, the existing MVSC [1] approach which is developed in a un-supervised environment, cannot guarantee for such a viewpoint set.

Suppose the class labels of a small subset of the dataset (denoted by ***LS***) is available to the user, these labeled documents can be usually used as seeds of the initial clusters instead of random initialization. Hence, a group of initial clusters with better quality is formed. Consequently, more appropriate viewpoints can be found for each pair of documents; thereby the clustering performance could be improved. However, it may not be the best way yet to utilize the labels in the MVS concept, because the risk of those inappropriate viewpoints still remains, especially when the given labels do not cover every class of the dataset.

Now the interesting question is, is it possible to reduce the misleading caused by the inappropriate viewpoints, if we can create a less number but in average 'higher qualified' appropriate viewpoint set for each cluster and document, rather than using a very large number of viewpoints in the existing MVSC?

We believe, ***LS*** itself is such a set of appropriate viewpoints as required. In this way, the candidates of viewpoint are now narrowed down from any document from the dataset to the documents from ***LS*** only. In other words, during the clustering process, the similarity of two documents in the same cluster r is now re-defined as the average of similarities measured

relatively from the views of all the documents with a known class label which do not belong to cluster *r*. The principle behind is the members in a particular cluster should always stay "far away" from those documents which are already known to belong to other clusters, in order to obtain a lower inter-similarity value between them. Therefore, a labeled document will have a much higher chance to serve as an appropriate viewpoint than an unlabeled document, reflected by a larger distance value on $\|d_i - d_h\|$.

The expression of $LMVS(d_i, d_j | d_i, d_j \in S_r)$ can be then written as below:

$$LMVS_{d_i, d_j \in S_r}(d_i, d_j) = \frac{1}{g - g_r} \sum_{d_h \in LS \setminus LS_r} Sim(d_i - d_h, d_j - d_h)$$

$$= \frac{1}{g - g_r} \sum_{d_h \in LS \setminus LS_r} \left(d_i^t d_j - d_i^t d_h - d_j^t d_h + d_h^t d_h \right) \quad .. \quad (4.9)$$

However, from Eq.(4.9), it does not show clearly yet that the LMVS measure is better compared to the original MVS measure. Therefore, in the next a few sections, we are going to justify its potential advantage by looking into the formulation of the corresponding criterion functions and conducting the validity test on the multi viewpoint-based similarity matrix.

4.3.2 New Criterion Functions: LMVS-I_R and LMVS-I_V

Similar with Section 4.2.3, two new clustering criterion functions can be formulated accordingly, by calculating the cluster size-weighted sum of the average pairwise similarities of documents in the same cluster and the sum of the similarity between each document and its cluster's centroid using Eq.(4.9). Since only those labeled documents can be used as the viewpoints now, we denote this new method by LMVS Clustering, stands for Label-based Clustering with Multi-Viewpoint based Similarity. Subsequently, the new criterion functions are named as LMVS-I_R and LMVS-I_V, respectively, or I_{RL} and I_{VL} in short.

The general form of the former function is now expressed by:

$$F_L = \sum_{r=1}^{k} n_r \left[\frac{1}{n_r^2} \sum_{d_i, d_j \in S_r} LMVS(d_i, d_j) \right] \quad (4.10)$$

where

$$\sum_{d_i, d_j \in S_r} LMVS(d_i, d_j)$$

$$= \sum_{d_i, d_j \in S_r} \frac{1}{g - g_r} \sum_{d_h \in LS \setminus LS_r} \left(d_i^t d_j - d_i^t d_h - d_j^t d_h + d_h^t d_h \right)$$

$$= \sum_{d_i, d_j \in S_r} d_i^t d_j - \frac{2n_r}{g - g_r} \sum_{d_i \in S_r} d_i^t \sum_{d_h \in SL \setminus SL_r} d_h + n_r^2 = \|D_r\|^2 - \frac{2n_r}{g - g_r} D_r (L - L_r) + n_r^2$$

After adding the regulating factor α and removing the constant *n*, we get the finally criterion function to be maximized as below:

$$I_{RL} = \sum_{r=1}^{k} \frac{1}{n_r^{1-\alpha}} \left[\|D_r\|^2 - \frac{2n_r}{\mathrm{g} - \mathrm{g}_r} D_r^t \left(L - L_r \right) \right]. \tag{4.11}$$

Similarly, the general form of the latter criterion function can be revised based on Eq.(4.7) as below:

$$G_L = \sum_{r=1}^{k} \sum_{d_i \in S_r} \frac{1}{g - g_r} \sum_{d_h \in LS \setminus LS_r} Sim\left(d_i - d_h, \frac{C_r}{\|C_r\|} - d_h \right), \tag{4.12}$$

Since $\frac{C_r}{\|C_r\|} = \frac{D_r}{\|D_r\|}$, exploring the vector dot product, we have:

$$\sum_{d_i \in S_r} \sum_{d_h \in LS \setminus LS_r} Sim\left(d_i - d_h, \frac{C_r}{\|C_r\|} - d_h \right) = \sum_{d_i \in S_r} \sum_{d_h \in LS \setminus LS_r} \left(d_i^t \frac{C_r}{\|C_r\|} - d_i^t d_h - d_h^t \frac{C_r}{\|C_r\|} + d_h^t d_h \right)$$

$$= (g - g_r) D_r \frac{D_r}{\|D_r\|} - \left(1 + \frac{n_r}{\|D_r\|} \right) D_r^t \left(L - L_r \right) + n_r (g - g_r)$$

Substituting the above expression into Eq.(4.12), and eliminating the constant *n*, we found maximizing *G* is equivalent to maximizing I_{VL} shown as below:

$$I_{VL} = \sum_{r=1}^{k} \left[\|D_r\| - \frac{1}{g - g_r} \left(1 + \frac{n_r}{\|D_r\|} \right) D_r^t \left(L - L_r \right) \right]. \tag{4.13}$$

The strength of the LMVS Clustering can be observed from the comparisons made between the intra-cluster similarity term and the inter-cluster similarity term in Eq.(4.11) or (4.13) with the respective terms in Eq.(4.6) or (4.8). Firstly, an new intra-cluster similarity term which is no longer sensitive to the cluster size n_r is obtained. More importantly, we see how the prior knowledge in terms of class label has now been successfully incorporated into the MVS-based clustering framework via a more appropriate formulated inter-cluster similarity term $D_r^t(L - L_r)$. To obtain the optimal partitioning of a dataset by the LMVS Clustering, the term $D_r^t(L - L_r)$ is minimized to ensure the documents in a particular cluster *r*, represent by D_r, will keep "far away" from a set of documents that are already known as the real "outsiders" of cluster *r*, denoted by $(L - L_r)$. In principle, this is what we expect if the similarity measure is able to appropriately capture the underlying structure of the dataset. While in MVSC, $D_r^t D$ is no longer good enough to precisely formulate this relationship, since *D* is a constant vector.

Moreover, it is realized in this LMVS model, the main role of each labeled document is no longer guiding the document search inside its own cluster, but serve and help the documents outside its clusters. In the opposite point of view, each unlabeled document can be

generally benefited from enough number of labels from all the *k-1* classes. Therefore, while most traditional search-guiding based semi-supervised clustering have high demand on the distribution of the labeled documents, to avoid performance degradation due to the imbalanced labelling discussed in [150] and incomplete seeding discussed in [118], we believe in principle the performance of LMVS Clustering may not be affected much from these issues.

On the other hand, it is important to point out that in practice, if all the labeled documents come from one class only, there is no viewpoint can be selected for that corresponding cluster during the clustering process. In the objective formulation level, the second term in I_{RL} and I_{VL} goes infinite and prevent the algorithm from a normal convergence. This implies the LMVS model is unable to handle this special case.

4.3.3 MVS with Pair-wise Constraints

LMVS measure and clustering can be formulated and applied only when some class labels is provided. However, the cost of exact label information may be sometimes so expensive to the user. Instead, only a few pair-wise constraints are available to the user. For example in text clustering applications, both *must-link* and *cannot-link* constraints between the documents, may be easily obtained in various ways, such as by computing the links between web documents, or by mining the search engine query logs.

In this section, we present how to make use of a set of pair-wise constraints to improve the MVS measure, and therefore boost the clustering performance. Similar to the LMVS measure, we aim to find out a group of more appropriate viewpoints rather than a normal 'outsider' used in original MVSC with the help of these constraints. We firstly divide the dataset into two subsets: constrained set (***ConS***) and free set (***FrS***). The documents involved in at least one constraint will be put in ***ConS***, while the rest will be put in ***FrS***. Based on the definition of an appropriate viewpoint given in Section 4.2.4, we denote that two documents in a *cannot-link* constraint can always be an appropriate viewpoint for each other. The reason is no matter which clusters they are during the clustering process, they won't be placed in the same cluster. Therefore, for a given document d_i in ***ConS***, we only consider all the documents which have a direct *cannot-link* constraint with it as its viewpoints. Furthermore, with a *must-link* constraint(d_i, d_j), d_j is able to share all its viewpoints with d_i, vice versa. In other words, more viewpoints are possibly captured from the *must-link* constraints. Lastly, these selected viewpoints are also shared with any other documents which are currently partitioned in the same cluster with d_i. The propose is to generalize these appropriate viewpoints to those documents involved in only *must-link* constraints in ***ConS*** and the documents in ***FrS***, as in principle all the documents should be fairly benefited from the help of pair-wise constraints. Therefore, similarity of two documents in the same cluster *r* is defined as the average of

similarities measured relatively from the views of all the documents which have a *cannot-link* constraint to any of the documents in cluster *r*. In other words, we obtain a common set (VS_r) consists of w_r *cannot-link* constraint-based viewpoints for each document in the same cluster *r*. Last but not least, it is noted PMVS measure should work under an assumption which the *cannot-link* constraint can cover every cluster of the dataset. If this condition applies, the expression of $PMVS(d_i, d_j | d_i, d_j \in S_r)$ can be written as:

$$\begin{aligned} PMVS_{d_i,d_j \in S_r}(d_i, d_j) &= \frac{1}{w_r} \sum_{d_h \in VS_r} Sim(d_i - d_h, d_j - d_h) \\ &= \frac{1}{w_r} \sum_{d_h \in VS_r} \left(d_i^t d_j - d_i^t d_h - d_j^t d_h + d_h^t d_h \right) \end{aligned} \quad . \tag{4.14}$$

4.3.4 New Criterion Functions PMVS-I_R and PMVS-I_V

Similar to what we presented in Section 4.3.2, another two new MVS-based clustering criterion functions can be formulated, based on Eq.(4.14) accordingly. Since the selection of the viewpoints depends on the pair-wise constraints now, we denote the method by PMVS Clustering, which stands for Pair-wise constraint-based Clustering with Multi-Viewpoint based Similarity. Subsequently, we name the two criterion functions as PMVS-I_R and PMVS-I_V, respectively, or I_{RP} and I_{VP} in short.

Once again, the cluster-size weighted sum of the average pair-wise similarities of documents in the same cluster is now expressed in Eq.(4.15) as below:

$$F_P = \sum_{r=1}^{k} n_r \left[\frac{1}{n_r^2} \sum_{d_i, d_j \in S_r} PMVS(d_i, d_j) \right] \tag{4.15}$$

where

$$\sum_{d_i, d_j \in S_r} PMVS(d_i, d_j)$$

$$= \sum_{d_i, d_j \in S_r} \frac{1}{w_r} \sum_{d_h \in V_r} \left(d_i^t d_j - d_i^t d_h - d_j^t d_h + d_h^t d_h \right)$$

$$= \sum_{d_i, d_j \in S_r} d_i^t d_j - \frac{2n_r}{w_r} \sum_{d_i \in S_r} d_i^t \sum_{d_h \in V_r} d_h + n_r^2 = \|D_r\|^2 - \frac{2n_r}{w_r} D_r V_r + n_r^2$$

For the sum of each document-to-centroid similarities using PMVS, the criterion function is written as:

$$G_P = \sum_{r=1}^{k} \sum_{d_i \in S_r} \frac{1}{v_r} \sum_{d_h \in VS_r} Sim\left(d_i - d_h, \frac{C_r}{\|C_r\|} - d_h \right) \tag{4.16}$$

Exploring the vector dot product, we have:

$$\sum_{d_i \in S_r} \sum_{d_h \in V_r} Sim\left(d_i - d_h, \frac{C_r}{\|C_r\|} - d_h\right)$$

$$= \sum_{d_i \in S_r} \sum_{d_h \in V_r} \left(d_i^t \frac{C_r}{\|C_r\|} - d_i^t d_h - d_h^t \frac{C_r}{\|C_r\|} + d_h^t d_h\right)$$

$$= w_r D_r \frac{D_r}{\|D_r\|} - \left(1 + \frac{n_r}{\|D_r\|}\right) D_r^t V_r + n_r w_r$$

Following the same derivation in Section 4.3.2, Eq.(4.15) and (4.16) can be rewritten as follows as the final form of the criterion functions to be maximized:

$$I_{RP} = \sum_{r=1}^{k} \frac{1}{n_r^{1-\alpha}} \left[\|D_r\|^2 - \frac{2n_r}{w_r} D_r^t V_r \right] \tag{4.17}$$

$$I_{VP} = \sum_{r=1}^{k} \left[\|D_r\| - \frac{1}{w_r} \left(1 + \frac{n_r}{\|D_r\|}\right) D_r^t V_r \right] \tag{4.18}$$

From Eq.(4.17) and (4.18), the two advantages we observed for LMVS Clustering can be also found in PMVS Clustering. Similarly, the inter-cluster similarity term $D_r^t V_r$ is minimized to ensure the documents in a particular cluster *r*, represent by D_r, will keep "far away" from a set of documents that are already known as the "outsiders" of some documents currently stay in cluster *r* via the *cannot-link* constraints, denoted by V_r. While the viewpoints in LMVS Clustering must be the documents in ***LS***, the viewpoints in PMVS Clustering must be selected from ***ConS***. In addition, two more variable w_r and V_r, denoted for the number and composite vector of viewpoints for cluster *r* are introduced in Eq. (4.17) and (4.18). It is noted that the viewpoints defined for PMVS Clustering are still not stable as the labeled documents in LMVS Clustering, as they are allowed to move from one cluster to another in the middle of the clustering process, as long as its constraints can be still satisfied at that moment. In other words, similar to MVSC, the set of viewpoints for each cluster in PMVS Clustering may be also dynamically updated in any iteration of the Refinement process. A document may be assessed by a different group of viewpoints, if it is placed at a different position in the visiting order and therefore leads to a different move decision due to a distinct MVS value. Especially when the number of viewpoints is greatly reduced in PMVS Clustering, on average each viewpoint carries much more weightage to affect the clustering performance, compared to a normal viewpoint used in MVSC. Therefore, the optimization towards Eq. (4.17) or Eq. (4.18) may not be easily achieved without a carefully designed algorithm.

4.4. Analysis on Similarity Matrices

In this section, we present analytical study to show that, LMVS and PMVS measure formulated in Section 4.3 with the help of two types of prior knowledge could be more effective to reflect the underlying structure of data, compared with original the MVS and CS measure. Several validity tests are carried out for this purpose by checking how much these similarity measures coincide with the true class labels. It is based on one principle: if a similarity measure is appropriate for the clustering problem, for any of a document in the dataset, the documents that are closest to it based on this measure should be in the same cluster with it.

For each type of similarities, a similarity matrix $\mathbf{A} = \{a_{ij}\}_{n\times n}$ is created first. Then, the validity score computed based on each of the matrices reflects how much this measure coincides with the true class.

Now we introduce how to form the pair-wise similarity matrices for each of the similarity measures. It is noted for all the MVS-based similarity matrices, we have to refer to the ground truth of the dataset to indicate the corresponding viewpoints (real outsiders) for each document and the category it belongs to. Given a small subset of the dataset to be used as viewpoints, the procedure for building the LMVS matrix (named as MVS_L) is described in Fig.4.1. Firstly, the vector composite of the viewpoints w.r.t. each class is determined, then, for each row $\mathbf{a}_i$ of $\mathbf{A}$, i= 1,…,*n*, which corresponds to the pair-wise similarity between a particular d_i and d_j, j = 1,…,*n*. If d_j is one of the viewpoints of d_i, a_{ij} is calculated as in line 10, Fig. 4.1. Otherwise, it is calculated as in line 12, Fig. 4.1. Similarly, given a set of pair-wise constraints, the procedure for building the PMVS matrix (named as MVS_P) is described in Fig. 4.2. Meanwhile, the corresponding procedures for building the other two matrices MVS_I and MVS_R are described in Fig. 4.3. Be different from the MVS_L and MVS_P, these two MVS Matrices are built by making use of all the 'outsiders' as viewpoints. MVS_I stands for the ideal MVS matrix, which assumes all the documents are placed at the right category denoted by its ground truth label, so that everyone works as an appropriate viewpoint and the misleading effect is minimized. This test helps us to observe the impact of the heavily reduction of the number of viewpoints to the effectiveness of MVS measure. On the other aspect, the ideal MVS seems never to be available in the real world applications. Therefore, another MVS matrix MVS_R is built by taking document labels assigned by an actual MVS-based clustering($MVSC\text{-}I_R$) instead of the ground truth labels, to investigate how and how much the inappropriate viewpoints can affect the effectiveness of MVS-based clustering. We believe this may be a more appropriate validity than MVS_I to simulate how the data partitioning is manipulated by MVS measure in real cases, as some wrongly assigned documents are naturally participated in the clustering process, and mislead the similarity

1**procedure** BULIDLMVSMATRIX(A_L)
2 for $r \leftarrow 1 : c$ do
3 $L_{LS\backslash LS_r} \leftarrow \sum_{l_i \notin LS_r} l_i$
4 $g_{LS\backslash LS_r} \leftarrow |LS\backslash LS_r|$
5 end for
6 for $i \leftarrow 1{:}n$ do
7 $r \leftarrow class\ of\ d_i$
8 for $j \leftarrow 1{:}n$ do
9 if $d_j \in LS\backslash LS_r$then
10 $a_{ij} \leftarrow d_i^t d_j - d_i^t \frac{L_{LS\backslash LS_r} - d_j}{g_{LS\backslash LS_r} - 1} - d_j^t \frac{L_{LS\backslash LS_r} - d_j}{g_{LS\backslash LS_r} - 1} + 1$
11 else
12 $a_{ij} \leftarrow d_i^t d_j - d_i^t \frac{L_{LS\backslash LS_r}}{g_{LS\backslash LS_r}} - d_j^t \frac{L_{LS\backslash LS_r}}{g_{LS\backslash LS_r}} + 1$
13 end if
14 end for
15 end for
16 return $A_L = \{a_{ij}\}_{n \times n}$
17 **end procedure**

Figure 4.1: Procedure: Build MVS_L

1 **procedure** BULIDPMVSMATRIX(A_P)
2 for $r \leftarrow 1 : c$ do
3 $V_r \leftarrow \sum_{d_i \in VS_r} d_i$
4 $w_r \leftarrow |VS_r|$
5 end for
6 for $i \leftarrow 1{:}n$ do
7 $r \leftarrow class\ of\ d_i$
8 for $j \leftarrow 1{:}n$ do
9 if $d_j \in VS_r$then
10 $a_{ij} \leftarrow d_i^t d_j - d_i^t \frac{V_r - d_j}{w_r - 1} - d_j^t \frac{V_r - d_j}{w_r - 1} + 1$
11 else
12 $a_{ij} \leftarrow d_i^t d_j - d_i^t \frac{V_r}{w_r} - d_j^t \frac{V_r}{w_r} + 1$
13 end if
14 end for
15 end for
16 return$A_P = \{a_{ij}\}_{n \times n}$
17 **end procedure**

Figure 4.2: Procedure: Build MVS_P

measure and the categorization of the documents. Last but not least, the CS matrix is also built by computing $a_{ij} = d_i \cdot d_j$, to compare with the four MVS matrices.

The complexity of computing all the four MVS-based similarity matrices is the same. For a randomly picked document pair (d_i, d_j), MVS(d_i,d_j) require m^2 times of multiplication as d_i and d_j are both a *m*-dimensional vector. For the full similarity matrix ***A***, there are n(n-1)/2 distinct pair-wise similarities need to be computed. Therefore, the total computation work for constructing ***A*** is $m^2n(n-1)/2$ times of multiplication. In terms of Big O, the "Complexity" of constructing the matrix is $O(m^2n^2)$. It is also noted that, the actual computational workload of

the matrix and the following validity process can be much faster because only a small number of non-zero values in the *m*-dimensional vector are in use.

After the similarity matrices are formed, the procedure shown in Fig. 4.4 is used to compute the respective validity scores. For each document d_i corresponding to row $\mathbf{a}_i$ of A, q_r documents which are the closest to d_i are selected. The value of q_r is chosen based on the size of the class c that contains d_i. Then, validity w.r.t. d_i is calculated by the fraction of these q_r documents having the same class label with d_i, as in line 13, Fig.4.4. It is clear that validity score is bounded within 0 and 1. The higher validity score a similarity measure has, the more suitable it should be for the clustering task.The final validity score is determined by averaging over all the rows of **A**, as in line 15, Fig. 4.4.

Four real-world benchmark datasets are used as examples in the validity test. They are *Reuters7, k1b, tr11* and *tr23*. The first one is the subset of Reuters-21578 Distribution 1.0, of Reuter's newswire articles. *k1b* is a collection of 2,340 web pages from the Yahoo! Subject hierarchy. *tr11* and *tr23* come from the "Text Retrieval Conference Data", which is co-sponsored by NIST[1]. These datasets were preprocessed by the standard procedure for text corpus, including stemming, the removal of stop words. Moreover, the words that appear in less than two documents or more than 99.5% of the total number of documents are also removed. Finally, the documents were weighted by TF-IDF weighting and normalized to unit length. The full characteristics of these four datasets are presented in Fig. 4.5.

For MVS_L, we randomly select 5% documents from each dataset and only these documents are used as viewpoints. For MVS_P, the documents in the *cannot-link* constraints are used as viewpoints. We firstly calculate the number of *cannot-link* constraints can be formed between any two documents from the 5% labeled documents used for MVS_L, and then

1 **procedure** BULIDMVSMATRIX(A_I or A_R)
2 for $r \leftarrow 1:c$ do
3 $D_{S\backslash S_r} \leftarrow \sum_{d_i \notin S_r} d_i$
4 $n_{S\backslash S_r} \leftarrow |S\backslash S_r|$
5 end for
6 for $i \leftarrow 1:n$ do
7 $r \leftarrow class\ of\ d_i$
8 for $j \leftarrow 1:n$ do
9 if $d_j \in S_r$ then
10 $a_{ij} \leftarrow d_i^t d_j - d_i^t \frac{D_{S\backslash S_r}}{n_{S\backslash S_r}} - d_j^t \frac{D_{S\backslash S_r}}{n_{S\backslash S_r}} + 1$
11 else
12 $a_{ij} \leftarrow d_i^t d_j - d_i^t \frac{D_{S\backslash S_r} - d_j}{n_{S\backslash S_r} - 1} - d_j^t \frac{D_{S\backslash S_r} - d_j}{n_{S\backslash S_r} - 1} + 1$
13 end if
14 end for
15 end for
16 return$A_I(A_R) = \{a_{ij}\}_{n \times n}$
17 **end procedure**

Figure 4.3: Procedure: Build MVS_I and MVS_R

Require: 0< percentage ≤1

procedure GetValidaity (validaity, A, percentage)

  For $r \leftarrow 1 : c$ do

    $q_r \leftarrow |percentage \times n_r|$

    if $q_r = 0$ then $percentage$ too small

      $q_r \leftarrow 1$

    End if

  end for

  for $i \leftarrow 1{:}n$ do

    $\{a_{iv[1]}, \ldots, a_{iv[n]}\} \leftarrow$ Sort $\{a_{i1}, \ldots, a_{in}\}$

    $s.t.\ a_{iv[1]} \geq a_{iv[2]} \geq \cdots \geq a_{iv[n]}$

      $\{v[1], \ldots, v[n]]\} \leftarrow permute\{1, \ldots, n\}$

    $r \leftarrow$ class of d_i

    $validity(d_i) \leftarrow \dfrac{|\{d_{v[1]}, \ldots, d_{v[q_r]}\} \cap S_r|}{q_r}$

  end for

  $validity \leftarrow \dfrac{\sum_{i\leftarrow 1}^{n} validity(d_i)}{q_r}$

  return *validity*

end procedure

Figure 4.4: Procedure: Get validity score

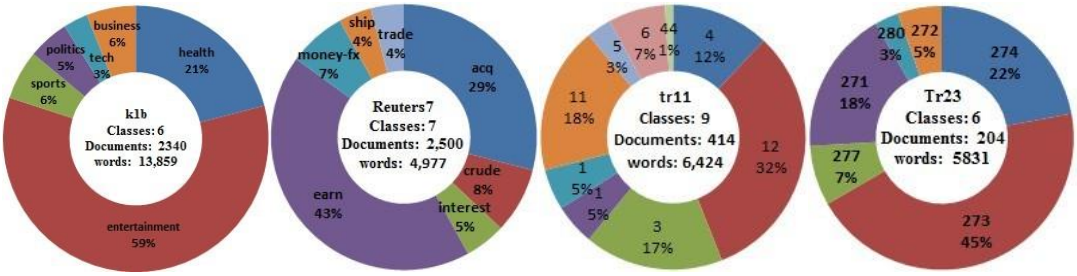

Figure 4.5: Characteristics of four datasets

random select the same number of constraints from the whole dataset for each dataset. The *must-link* constraint is not necessary to take account in for the validity test, as the viewpoints for building both MVS_L and MVS_R will be automatically shared by each document in the same class.

Meanwhile, it is noted the scores of CS matrix and MVS_I for a given dataset are fixed, but that of MVS_L, MVS_P and MVS_R are dependent on the selection of labeled documents, pair-wise *cannot-link* constraints and the clustering results of MVSC-I_R. Due to this concern, the mean score of 10MVS_R based on 10 independent experimental test run is computed. Each test run selects the best trial in terms of the corresponding criterion function value among 10 independent trials. Similarly, we also compute the mean score of 10 MVS_L or MVS_R with 10 different label set or pair-wise constraints set, respectively, for a fair comparison.

Fig. 4.6 shows the validity scores of all five matrices on these datasets relative to the parameter *percentage*. The value of percentage is set at 0.001, 0.01, 0.05, 0.1, 0.2… 1.0. The strength of LMVS measure and PMVS measure can be reflected from the corresponding validity scores. Firstly, with a much less number of viewpoints, the validity scores of both MVS_L and MVS_P are very close to the scores of MVS_I. It implies that the quantity of the

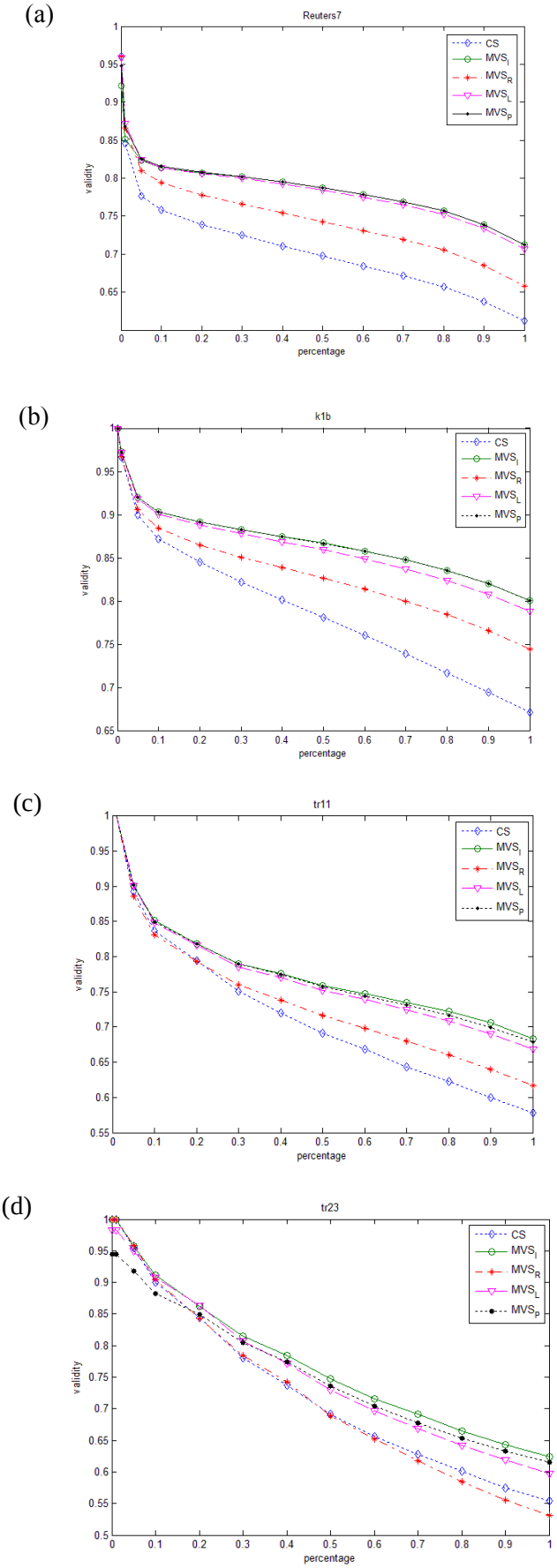

Figure 4.6: Validity scores of the simiarlity matrices

viewpoints is not the most important factor to an effective MVS at all. Secondly, the score of MVS_L is always higher than that of MVS_R and CS. The gap between the score of MVS_R and the score of MVS_L is continuously enlarged with the increase of the validity percentage, due to the misleading of the inappropriate viewpoints. For example, with *Reuters7* at percentage = 1.0, the score of MVS_I and MVS_L are both 0.71, while that of MVS_R reduces to 0.66.

On the other hand, the scores of MVS_R on *k1b*, *Reuters7*and *tr11* are higher than the scores of CS matrix. For example, the score of MVS_R on *k1b* is 0.74, while that of CS matrix is 0.67. On the other hand, we could see the score of MVS_R on *tr23* drop down to the lowest value among all the five matrices. It shows that how the wrongly assigned documents disturb the MVS measure as inappropriate viewpoints. Some possible reasons can be given if we refer back to the performance of MVSC-I_R on these datasets. It is noted the corresponding clustering performances on these four datasets in terms of *Accuracy* are: 0.704, 0.836, 0.732 and 0.489, respectively. The *accuracy* values on first three datasets reach over 70%, which means at least 70% documents can be the appropriate viewpoints. In addition, when a document which should belong to cluster r_1 is wrongly assigned to another cluster r_2, it is still able to serve as an appropriate viewpoint for the documents in the rest (*k-2*) clusters. In this case, most of the documents are still able to play the roles of the appropriate viewpoints, and therefore; the validity score may not decrease much. Meanwhile, the ratio of the inappropriate viewpoints may significantly increase if the validity test is conducted based on a poor clustering performance, e.g.: *tr23*, then the score of MVS_R becomes even worse than the CS matrix, since the MVS measure can be heavily misled due to large portion of inappropriate viewpoints.

In fact, we realize that, the exact class label information is no longer important to the user for applying the MVS measure, therefore, MVS_L can be considered as a special case of the MVS_P. The effect of viewpoints given by a MVS_L will be identical to a MVS_P if the *cannot-link* constraints in this MVS_P cover any two documents with in a subset of the dataset and only concentrate in this subset. In general, a MVS_P built using the exact same number of *cannot-link* constraints which can be formed by the class labels in a MVS_L on the same dataset, usually provides more appropriate viewpoints for each document than the MVS_L, because some *cannot-link* constraints formed in MVS_L may be automatically obtained by the sharing process in the respective MVS_P. This could explain why in general the MVS_P is able to give higher validity scores than MVS_L.

However, it is important to point out that although the score of MVS_P may be slightly higher than that of MVS_L in validity test, it could be quite different in the real MVS-based clustering process. In the validity test, we ensure that all selected viewpoints are the appropriate ones, except for MVS_R. However, the viewpoints in the actual PMVS Clustering

are still allowed to move from one cluster to another during the real clustering process, so they may be not guaranteed to be the appropriate ones as LMVS Clustering for all the other documents in the dataset all the time.

In conclusion, we see the potential advantages of using a less number but 'high qualified' appropriate viewpoint set to carry out the MVS measure during the real clustering process through the validity tests. Therefore, we would like to make a very important conclusion that is as long as such a viewpoint set is available with the help of prior knowledge; a large number of viewpoints are no longer necessary for an effective MVS-based clustering. In other words, the quantity of the viewpoints is not the most critical issue, but the overall quality is.

4.5 Optimization Algorithms and Complexity

In this section shows how to perform clustering by newly proposed MVS-based semi-supervised algorithm to optimize the derived criterion functions. We denote the clustering framework at topic level by SS-MVSC, meaning Semi-supervised Clustering with Multi-Viewpoint based Similarity, which includes both LMVS Clustering and PMVS Clustering as two applicable options due to the different types of prior knowledge provided in real applications.

4.5.1 Optimization Algorithm for LMVS Clustering

The criterion function LMVS-I_R & LMVS-I_V depend only on two variables: n_r and D_r. r = 1,…,k. Hence, they can be written in the general form:

$$I = \sum_{r=1}^{k} I_r \left(n_r, D_r \right) \tag{4.19}$$

where $I_r(n_r , D_r)$ corresponds to the criterion function value of cluster r. With this general form, the algorithm includes two major steps, Initialization with labeled documents and Refinement on the unlabeled documents, as described in Fig.4.7. At Initialization, a few labeled documents are selected from the dataset and their labels are kept in the clusters which each of them truly belongs to. The mean vector of these labeled documents in each cluster is computed and treated as the virtual center. The first partitioning of unlabeled documents is formed based on the CS measure between each document and the centers. Refinement is a procedure that consists of a number of iterations. The unlabeled documents are visited one by one in a random order without repeat at each iteration. Each document is checked to see if its move to another cluster would result in any positive improvement of the criterion function value. If yes, the document is moved to the cluster which leads to the highest improvement. Otherwise, the document stays in its current cluster. The Refinement process terminates when an iteration completes without any document being moved to a new cluster. As L and L_r are fixed during the whole clustering process, there will be no change on the set of viewpoints for

Input: Dataset ***D***, labeled set ***LS***
Output: Partitioning Matrix ***P***
1 **procedure** INITIALIZATION
2 Initialize the centers $c_{1,\dots}c_k$ using the labels
3 Randomly select the centroids for the clusters which is not covered by *LS*
4 $cluster[d_i] \leftarrow p = \arg max_r\{s_r^t d_i\}, \forall i \in S \backslash LS$
5 $D_r \leftarrow \sum_{d_i \in S_r} d_i, n_r \leftarrow |S_r|, \forall r = 1, \dots, k$
6 **end procedure**
7 **procedure** REFINEMENT
8 **repeat**
9 $\{v[1: (n\text{-}g)]\} \leftarrow$ random permutation of $\{i | i \in S \backslash LS\}$
10 for $j \leftarrow 1: (n\text{-}g)$ do
11 $i \leftarrow v[j]$
12 $p \leftarrow cluster[d_i]$
13 $\Delta I_p \leftarrow I(n_p - 1, D_p - d_i) - I(n_p, D_p)$
14 $q \leftarrow arg \max_{r, r \neq p}\{I(n_r + 1, D_r + d_i) - I(n_r, D_r)\}$
15 $\Delta I_q \leftarrow I(n_q + 1, D_q + d_i) - I(n_q, D_q)$
16 if $\Delta I_p + \Delta I_q > 0$ then
17 Move d_i to cluster q: $cluster[d_i] \leftarrow q$
18 Update D_p, n_p, D_q, n_q
19 end if
20 end for
21 **Until** No move for all (n-g) unlabeled documents
22 **end procedure**

Figure 4.7: Detailed Steps of LMVS Algorithm

any unlabeled documents in the same cluster. Therefore, the sensitivity of LMVS Clustering to the random visiting order of the documents is significantly reduced compared with the original MVSC.

4.5.2 Optimization Algorithm for PMVS Clustering

Compare to LMVS Clustering, the formula of respective I_R and I_V of PMVS Clustering have two additional variables: w_r and V_r, other than n_r and D_r. $r = 1,\dots,k$. Hence, the general form of PMVS Clustering could be formulated to:

$$I = \sum_{r=1}^{k} I_r \left(w_r, n_r, V_r, D_r \right), \tag{4.20}$$

where I_r (w_r, n_r, V_r, D_r) corresponds to the criterion function value of cluster r. In general, the algorithm also includes the Initialization and Refinement as the two major steps. However, the design of Refinement in PMVS is more complicated than that in LMVS. A detailed illustration of the algorithm is given in Fig.4.8.

Starting from k random initialized centroids, the documents are partitioned into k clusters according to their CS to the centroids, while the constraints must be satisfied. Then, the viewpoints for the documents in the same cluster are selected according to the pair-wise constraints. Refinement consists of a number of iterations. Each iteration can be further divided into 3 steps. Frist of all, given a random visiting order of the documents in ***FrS***, each of them is visited without repeat to decide if it should move out or remain in the current

Input: Dataset ***D***, ***MLS*** & ***CLS***
Output: Partition Matrix: ***P***

procedure INITIALIZATION
  Select *k* seeds $s_1, s_2, \ldots s_k$ randomly
  $cluster[d_i] \leftarrow p = \arg max_r\{s_r^t d_i\}, \forall i \in S/SS$
  $D_r \leftarrow \sum_{d_i \in S_r} d_i, n_r \leftarrow |S_r|, \forall r = 1, \ldots, k$
  SOM formed if any missed document is recognized.
end procedure
procedure REFINEMENT
  repeat
    $\{v[1:(n_{fs})]\} \leftarrow$ random permutation of $\{i | i \in FrS\}$
    for $j \leftarrow 1: n_{fs}$ do
      $i \leftarrow v[j]$; $p \leftarrow cluster[d_i]$
      $\Delta I_p \leftarrow I(n_p - 1, D_p - d_i) - I(n_p, D_p)$
      $q \leftarrow arg \max_{r, r \neq p}\{I(n_r + 1, D_r + d_i) - I(n_r, D_r)\}$
      $\Delta I_q \leftarrow I(n_q + 1, D_q + d_i) - I(n_q, D_q)$
      if $\Delta I_p + \Delta I_q > 0$ then
        Move d_i to cluster q: $cluster[d_i] \leftarrow q$
        Update D_p, n_p, D_q,n_q
      end if
    end for
    $\{v[1:(n_{cs})]\} \leftarrow$ random permutation of $\{i | i \in ConS\}$
    checklist$\leftarrow$ { } // empty set
    for $j \leftarrow 1: n_{cs}$ do
      $i \leftarrow v[j]$; p$\leftarrow cluster[d_i]$;*igc*$\leftarrow$ { }
      $\Delta I_p \leftarrow I(n_p - (1 + ml_i), D_p - (d_i + M_i)) - I(n_p, D_p)$
      for every$(d_i, d_j) \in CLS$ &&$d_j \in checklist$
        $q \leftarrow cluster[d_j]$ // find *igc* for *i*
        igc= *igc* + *q*
      end for
      $q \leftarrow arg \max_{r, r \neq p, r \notin igc} \begin{Bmatrix} I\begin{pmatrix} n_r + (1 + ml_i), \\ D_r + (d_i + M_i) \end{pmatrix} \\ -I(n_r, D_r) \end{Bmatrix}$
      $\Delta I_q \leftarrow I(n_q + (1 + ml_i), D_q + (d_i + M_i)) - I(n_q, D_q)$
      if $\Delta I_p + \Delta I_q > 0$ then
        Move d_i and MS_i to cluster q
        Update D_p, n_p, D_q, n_q, V_p, w_p, V_q, w_q
      End if
      Update *checklist*
    end for
    $\{v[1:(n_m)]\} \leftarrow$ random permutation of $\{i | i \in SOM\}$
    for $j \leftarrow 1: n_m$ do
      $i \leftarrow v[j]$
      for every$(d_i, d_j) \in CLS$ &&$d_j \in checklist$
        $q \leftarrow cluster[d_j]$ // find *igc* for *i*
        igc= *igc* + *q*
      end for
      $q \leftarrow arg \max_{r, r \notin igc}\{I(n_r + 1, D_r + d_i) - I(n_r, D_r)\}$
      Move d_i to cluster q: $cluster[d_i] \leftarrow q$
      Update D_p, n_p, D_q, n_q, V_p, w_p, V_q, w_q
    end for
  Until No move for all *n* documents
end procedure
Noted: ml_i ,M_i*=0,* if *i* is only involved in *cannot-link* constraints

Figure 4.8: Detailed Steps of PMVS Algorithm

cluster, just like we did in LMVS. Then, the documents in ***ConS*** are visited individually or in group without repeat in another random visiting order, depend on if they are involved in one or more *must-link* constraints. Moreover, it is noted some documents may be labelled as "missed" if they could not be assigned into any of the clusters due to the constraints and the

order of partitioning. Therefore, in the last step we try to re-assign each "missed" document by putting it back into one of the cluster, after all the documents currently inside the clusters have been visited. This is considered as the last step. Each of the "missed" documents is assigned to the cluster which obtains the maximum MVS value between the document and the cluster centroid among all the k clusters, and at the same time it avoids the violation of the constraints.

In the next a few paragraphs, detailed analysis is provided to explain the key differences between the PMVS and LMVS algorithm. Especially in PMVS, some new terms are introduced and discussions are given to see how to make good use of the pair-wise constraints with the help of these terms to achieve a quick and smooth convergence. In order to give a clear picture of the whole algorithm, the reason of giving 3 steps instead only one in each iteration is firstly discussed. It follows by the elaboration on the 2nd step due to further utilizing the constraints to guide the cluster search other than the viewpoint selection.

Visiting Order

In the previous section, we have discussed that a stable set of appropriate viewpoints is generally helpful in the MVS-based clustering approaches. However, PMVS Clustering still has the similar risk caused by a dynamic viewpoint set as in MVSC, although we believe the viewpoints selected from the ***ConS*** are more appropriate than an 'outsider' in MVSC. In general, the stability of a viewpoint chosen from a *cannot-link* document pair is considered lower than a viewpoint with fixed class label. This is because the scope of the documents served by a viewpoint in PMVS Clustering may suddenly change in the middle of an iteration, if this viewpoint jumps to another cluster, on the premise of satisfying its constraints. Other than that, the impact created by the move of a viewpoint in PMVS could be even greater than a viewpoint in the original MVSC, especially if this viewpoint is also involved in certain *must-link* constraints. In this case, it may give high chance to lead the PMVS Clustering converges at a bad local maximal if a random visiting order is applied to all the documents in the dataset. We believe this problem can be avoided by carefully arranging the visiting order. That is, the visit of ***FrS*** should be scheduled as the first step in an iteration before any viewpoints are visited, to ensure the move decisions on majority of the documents are not affected by the impact of a dynamic updated viewpoint set.

On the other hand, those "missed" documents should be always handled in the last step, as they only get the chance to find an available cluster, after certain constrained documents have been moved to certain new clusters during the 2nd step. Otherwise they have to remain as "missed" documents until next iteration.

Discussion on the Cannot-link Constraints

During the Refinement process, the *cannot-link* constraints simply tell the algorithm that a few particular clusters can be skipped from checking the improvement of moving a new

document in, which usually speeds up the clustering process. However, trusting all the *cannot-link* constraints in checking every move of the documents may prevent the algorithm from achieving the real optimization. For example, in PC-Sph*kmeans*, exchanging the assigned cluster index of the two documents involved in a *cannot-link* constraint is common and applicable if it leads to an improvement of the overall criterion function value. However, unfortunately we are not able to take this action in the PMVS Clustering, because we are not reassigning all the documents into an empty search space in the next iteration.

In principle, we hope the PMVS Clustering could be more robustness to converge at a good local maximum with the help of a well-designed algorithm. Therefore, we made some necessary improvement on the PMVS algorithm to still allow a document involved in one or more *cannot-link* constraints moving to any other clusters, if the other document involved in the constraint have not been visited in the same iteration. It is similar to the first document in a *cannot-link* constraint can be re-assigned to any of the clusters in *kmeans* clustering, if its "partner" has not been re-assigned yet in this iteration.

In this case, a new term called *checklist* is introduced to record the indices of the documents that have been visited in ***ConS*** during the 2nd step. Only those constraints associated with the documents recorded in *checklist* are taken effect at that moment, while the others are temporarily ignored. Meanwhile, another new term *igc* (stands for ignored cluster) is used to store the cluster indices. Every document in ***ConS*** has its own *igc*, which works closely with the *checklist*, tells the algorithm which clusters can be really ignored from checking the respective improvement when it is visited.

Suppose two *cannot-linked* documents d_i and d_j are located at cluster r_1 and r_2, respectively. d_i is visited before d_j in a particular iteration. There are a few possible scenarios for the move of them during the Refinement.

Scenario 1:

The improvement achieved by moving d_i out from cluster r_1 is negative, then d_i is decided to stay at cluster r_1, and r_1 is added into the *igc* for d_j. When d_j is visited, cluster r_1 will not be checked and so the improvement of moving it into cluster r_1 will not be calculated.

Scenario 2:

Among all the *k*-1 clusters, the maximum improvement is achieved by moving d_i to a third cluster r_3, then d_i is moved to cluster r_3 immediately. Meanwhile, d_i is added into the *checklist*, and r_3 is added into *igc* for d_j as well. Similarly, when d_j is visited, the algorithm will only check the improvement by moving d_j into the rest *k*-2 clusters. r_3 is no longer considered.

Scenario 3:

Among all the *k-1* clusters, the maximum improvement is achieved by moving d_i to cluster r_2 (denoted by IP_{max}), and the second maximum improvement is achieved in cluster r_3 (denoted by IP_{max2}). In this case, the cost of moving d_j out from cluster r_2 must be calculated

before moving d_i into cluster r_2. It would be fine if one of the other clusters is able to accept d_j in and leads a positive improvement, including cluster r_1. However, if not, we need to calculate the minimum cost of moving d_j out from cluster r_2, subtract it from IP_{max} to get the overall improvement, and compare it with IP_{max2}. If IP_{max2} is larger, then d_i will be moved to cluster r_3; otherwise d_i will be moved to cluster r_2 and d_j will be moved to the cluster which leads to the minimum cost immediately. After that, both d_i and d_j are added into the *checklist*.

Scenario 4:

Continued with scenario 3, if r_2 is the only cluster that leads to an improvement for d_i, then the same action is taken for d_i and d_j if the overall improvement is positive. Otherwise, both d_i and d_j stay in their current cluster.

Discussion on the Must-link Constraints

Other than utilizing the *cannot-link* constraints, the *must-link* constraints also need to be satisfied together, while a smooth convergence to a good local maximal can be achieved. Normally, a document will be moved from its current cluster to another if and only if this leads to an improvement to the value of I_R or I_V. However, for a document involved in one or more *must-link* constraints in PMVS Clustering, we should check the "group" improvement made by the move of this entire group of *must-link*ed documents, calculated by line 29 in Figure 4.8, instead of the improvement made based on the move of an individual document, calculated by line 22. The reason to do so can be justified as follows. Given a set of must-linked documents, when only one of them is visited and a move decision is made, then the rest of the documents need to be moved together to the same cluster immediately without checking whether an improvement is also guaranteed to each of the following moves, respectively. In other words, the decision making is also considered sensitive to the visiting order within this set of must-linked documents as the first visited document becomes critical. In such a situation, the value of I_R or I_V may sometimes decrease, and the algorithm may not converge smoothly. However, by checking the "improvement in group", the algorithm is guaranteed for a smooth convergence, and all the constraints can be strictly satisfied during the clustering process.

4.5.3 Complexity

During the optimization procedure, in each iteration, the main sources of computational cost of the MVS Clustering usually include two parts:

- Searching for the optimum cluster to move each individual document to
- Updating composite vectors as a result of such move

From previous section, we understand to make the move decision for every individual document, the decrement to the criterion functions by moving it out from its current cluster and the increment to the criterion functions by moving it into the rest of the *k*-1 clusters need

be calculated. Each of them requires $O(m)$, where m is the total number of terms/words. Therefore, the computational complexity of each move decision is $O(m \cdot k)$, and the total complexity for the whole dataset is $O((n-g) \cdot m \cdot k)$ for LMVS Clustering and $O(n \cdot m \cdot k)$ for PMVS clustering, respectively. For updating the affected composite vectors D_r by such moves, each D_r requires $O(m)$, so the total complexity is $O(m \cdot k)$.

It is noted that our clustering approach is partitional and incremental; therefore it is not necessary to compute the entire similarity matrix in each iteration. Moreover, since the textual datasets are usually very sparse, we consider the total number of nonzero entities *nz* in the matrix D is much smaller than $n \cdot m$. In this case, the real computational cost for searching the optimum cluster in LMVS and PMVS Clustering can be reduced to $O(\widetilde{nz} \cdot k)$ and $O(nz \cdot k)$ respectively, where $\widetilde{nz}$ denotes the total number of non-zero entries in all unlabeled documents.

In conclusion, the time complexity of LMVS and PMVS Clustering are $O(\widetilde{nz} \cdot k)$ and $O(nz \cdot k)$, respectively.

Last but not least, for PMVS Clustering, although the big O remains the same as the original un-supervised MVSC, the actual computational cost can be cut down when some multiple searching tasks for a few *must-link* documents can be combined into one, and the number of clusters to be checked may be reduced with the help of the *checklist* and *igc*.

4.6 Experimental Results

To verify the advantages of the proposed methods, evaluation of the clustering performance through the experimental study on a number of benchmark datasets is carried out. We compare LMVS-I_R, LMVS-I_V, PMVS-I_R and PMVS-I_V with the state-of-art semi-supervised clustering algorithms that using SVS or implicit measure, and transductive learning method.

4.6.1 Datasets

16 benchmark datasets are used for the examination of the clustering methods in this chapter. All of them are preprocessed by exactly standard procedures which we described in Section 4.4. 10 of them have already tested in Chapter 3. The 6 more extremely unbalanced datasets, *tr11*, *tr12*, *tr23*, *tr31* from CLUTO [24] and *re0*, *Reuters7* from *Reuters-21578* are added into the experimental study to further validate the effectiveness of the semi-supervised MVS-clustering approaches. The basic characteristics of these dataset are summarized in Table 4.2. A dataset is considered as extremely unbalanced if *Balance* is less than 0.1, therefore half of the testing datasets can be considered as extremely unbalanced in this chapter.

Table 4.2: Brief descriptions on datasets

Data	Source	*k*	*n*	*m*	Balance
CB	BankResearch[	2	2000	4791	1
SM	BankResearch	2	2000	5450	1
WPD6	BankResearch	6	600	2660	1
newsgroups	20newsgroups	20	2000	3703	1
Multi10	20newsgroups	10	500	2015	1
tr11	TREC	9	414	6424	0.045
tr12	TREC	8	313	5799	0.097
tr23	TREC	6	204	5831	0.066
tr31	TREC	7	927	10127	0.006
sports	TREC	7	8580	18324	0.036
k1b	WebACE	6	2340	13859	0.043
re0	Reuter]	13	1504	2886	0.018
Reuters7	Reuters	7	2500	4977	0.082
reuters3	Reuters	3	1076	2837	0.748
webkb4	WEBKB	4	4199	10921	0.364
classic	CACM/CISI/ CRAN/MED	4	7089	12009	0.323

4.6.2 Experimental Setting

To demonstrate how well the semi-supervised MVSCs performance can be achieved, we compare them with a few existing clustering/transductive methods. LMVS Clustering is compared with Constrained-Sph*kmeans* (C-Sph*kmeans*)[118] and weighted semi-supervised nonnegative matrix factorization with normalized cut weighting [130] (WNMF-NCW), two popular existing label-based semi-supervised approaches using SVS or implicate similarity measure, respectively. Moreover, we also compare with a robust graph-based transductive learning method called "Graph Transduction via Alternative Minimization" (GTAM) [152]. PMVS Clustering is compared with Pair-wise Constrained Spherical *kmeans* (PC-Sph*kmeans*) [106] and Penalized Matrix Factorization (PMF) [74]. The other two NMF-based clustering approaches appeared in Chapter 3 are not presented as we consider PMF is generally the best one for comparison in terms of *Accuracy* among the 3 NMF-based methods, through the extensive experimental study given in Chapter 3.

For each dataset, cluster number k is predefined equal to the number of true class c. τ_{max} is equal to 200, and the stopping threshold for C-Sph*kmeans*, PC-Sph*kmeans*, WNMF-NCW and PMF is set to 1E-5. The regulating factor α in I_R is fixed to 0.3, as one of the most appropriate values reported in [1]. As a graph-based approach, some data pre-processing effort has to be made for GTAM. The instructions given in the respective paper regarding the graph constructions and parameter tuning for textual dataset (*webkb4*) are strictly followed. The same graph construction suggested by [171], i.e.: linear kernels and cosine nearest neighbor graphs with Gaussian weights is used. The number of nearest neighbor is fixed at 200, sigma is set to the average edge length of the graph. The weighting factor μ is set to 99. On the other hand, we understand a better performance may be achieved by giving a different kernel or parameter selection on a different dataset. However, looking for the most suitable experimental protocol for every single dataset is out of the scope of this study.

It is noted in the label-based approaches, the initial centroid or label matrix in the respective algorithms is relevant to the selected labeled documents, which are randomly picked up from the whole dataset. In other words, the number of labeled documents in each of the known classes is indeed imbalanced. Especially for those extremely unbalanced datasets, it is quite common that no document will be picked up from those relatively smaller sized classes, and leads to the incomplete seeding situation. Ten independent experimental trials with different initial centroids will be considered as one test run, and the best trial in terms of the largest criterion function value obtained among the ten is chosen. The amount of selected labeled documents is proportional to the size of dataset *n*. In order to show both I_R and I_V are able to perform well with a very small number of labels compared with the size of dataset, we show the performance with only 1% to 3% documents with ground truth labels on twelve datasets which *n* is larger than 500, and 5% to 15% documents with ground truth labels in steps of 5% on the rest four datasets.

For GTAM, some manual adjustment is necessary to ensure at least one label can be provided for each of the classes. We randomly remove one labeled document from the class having the largest number of labels, and randomly pick up one document and label it in the empty class; so the total number of class label remains unchanged, and the overall distribution of the class labels is still random.

In the pair-wise constrained based approaches, the initial centroids and selected pair-wise constraints can be independent to each other; therefore, all algorithms use the same set of pair-wise constraints in each test run, which also consists of 10 independent trials given by 10 randomly initialized centroids. A different set of constraints is then applied on different test runs to study the impact from the pair-wise constraints. Furthermore, we provide two different systematic approaches to form the constraints. The first scenario is called “pair-up on subset”. We randomly label a few selected documents by referring the ground truth, pair up any two of the documents, get all the *must-link* constraints and randomly select equal number of *cannot-link* constraints. The reason of not using all the *cannot-link* constraints will be explained in the later sections. The second scenario is called “general random selection”. We randomly pick up two documents from the dataset, and give a *must-link* or *cannot-link* constraint by referring their ground truth labels. This could be repeated many times until we obtain exact the same number of pair-wise constraint (including *must-link* and *cannot-link*) to the total number of pair-wise constraints used in scenario 1. In general, the number of *cannot-link* constraints should be larger than the number of *must-link* constraints in scenario 2.

After all, the results reported in this chapter on each dataset by every method is the average of 10 test runs, if no specific other statement.

4.6.3 LMVS Clustering Results

This section gives the performance evaluation on a group of experiments for LMVS Clustering. The results in *NMI*, *Accuracy* and *F_Score* of four label-based clustering approaches and one transductive learning approach are presented in Table 4.3-4.5, respectively. The clustering performance given by original MVSC-I_R, MVSC-I_V, *spherical kmeans* [38] and original NMF-NCW [67] are tabulated in the row with 0% label for each dataset. The value in bold and underlined is the best result, while the value in bold only is the second best for each specific prior knowledge level.

LMVS v.s. MVSC:

Overall, we are able to draw consistent conclusions by observing results in the three tables. Firstly, we surely confirm that, LMVS Clustering (denoted by LMVS only for short in this section) outperforms MVSC on almost all the datasets by effectively incorporating the class labels into the clustering process. Secondly, consistent improvements, shown in the three evaluation metrics can be achieved when more available labeled documents are provided. Thirdly, a very small number of labels, in other words, appropriate viewpoints is required in this LMVS model. Significant performance improvement is easily achieved by providing only 1% labeled documents in 10 out of the first 12 datasets, except *WPD6* and *tr31*, compared with the original MVSC. And for the rest four, LMVS also outperforms MVSC with only 5% labels. This phenomenon does not only show the potential of our proposed method, but also indicates that the overall quality of the viewpoints is a more important factor for an effective MVS measure, rather than the quantity, as long as a "minimum" requirement on the quantity can be satisfied. If not, we realize LMVS sometimes may suffer from slightly setback on certain datasets, when the absolute number of the appropriate viewpoints is too small to provide an accurate similarity assessment, such as giving 1% labels for *WPD6* and *tr31*. In that case, with on average only one available viewpoint from each cluster during the clustering process, it leads to a poor clustering performance on both datasets using I_V, compared to making use of all the outsiders as viewpoints in MVSC. Meanwhile, the performance improves rapidly for *tr31* when another 1% labels is available to the user. Clustering results on some more datasets at higher prior knowledge level can be found in [172].

LMVS v.s. Others Clustering Approaches:

The strength of LMVS can be also shown when it is compared to other semi-supervised clustering approaches. Firstly, among the four approaches listed in Table 4.3-4.5, overall the top two are either both I_R and I_V or one of them. Except I_R gives comparable results to C-

Table 4.3: Clustering results of label-based methods in *NMI*

label rate	**Data**	LMVS-I_R	LMVS-I_V	C-Sphkmeans	WNMF-NCW	GTAM	**Data**	LMVS-I_R	LMVS-I_V	C-Sphkmeans	WNMF-NCW	GTAM
0	classic	57.4	**64.4**	**57.7**	59.3	-	k1b	**70.1**	67.1	64.9	**72.8**	-
1%		58.6	**67.8**	**64.3**	62.5	59.4		**74.4**	72.3	71.5	**73.2**	57.4
2%		58.6	**68.4**	64.6	**66.1**	64.2		**76.9**	74.3	74.2	**75.0**	62.5
3%		57.8	**69.0**	64.9	**66.8**	64.8		**77.3**	**75.8**	75.5	75.4	65.8
0	re0	39.9	**40.2**	**40.2**	38.4	-	news	**79.7**	**79.5**	74.8	73.2	-
1%		**42.1**	40.6	**41.4**	40.5	38.6	groups	**78.0**	**78.8**	75.6	75.8	66.6
2%		**43.4**	**42.6**	42.6	42.2	41.9		**80.2**	**81.3**	78.3	79.0	70.4
3%		43.7	**45.2**	**44.2**	43.3	43.2		**81.7**	**82.1**	79.9	80.2	72.6
0	tr31	**61.3**	**65.8**	58.6	54.5	-	WPD6	45.4	46.6	43.3	**47.2**	-
1%		**63.4**	**63.1**	59.2	55.8	56.6		45.8	43.1	46.3	**49.8**	**53.2**
2%		**68.1**	**69.6**	62.4	57.2	60.5		47.3	**51.2**	47.8	**50.4**	**54.1**
3%		**69.0**	**71.2**	66.1	62.5	59.7		**51.5**	**52.3**	47.8	51.2	**54.0**
0	sports	**66.9**	**71.9**	63.3	65.6	-	CB	2.3	2.3	**2.8**	**8.0**	-
1%		**72.3**	**74.8**	70.1	71.2	70.2		**18.0**	**20.7**	11.4	13.5	17.5
2%		**75.1**	**76.2**	69.8	71.6	70.9		**23.6**	**26.6**	12.8	17.2	22.4
3%		**75.6**	**76.8**	68.6	71.7	71.2		**31.0**	**32.9**	20.4	20.6	27.0
0	reuters3	35.3	**55.5**	54.6	**54.9**	-	SM	0.9	0.9	**13.3**	**4.5**	-
1%		49.0	**58.4**	54.6	54.7	**65.6**		40.0	**41.6**	30.1	18.4	**80.3**
2%		56.8	**64.8**	55.2	56.7	**69.4**		**55.7**	51.4	35.7	24.6	**92.1**
3%		56.8	**65.9**	55.2	58.4	**74.8**		65.3	**65.6**	63.2	26.5	**99.2**
0	Reuters7	**63.3**	**63.2**	61.2	58.4	-	webkb4	27.7	28.1	**37.1**	**38.3**	-
1%		**63.3**	**64.9**	61.6	61.0	62.4		**42.1**	**42.3**	40.6	40.4	38.2
2%		**67.1**	**67.7**	64.3	64.7	64.9		**44.2**	**43.5**	41.7	41.5	39.0
3%		**67.7**	**69.0**	63.7	66.5	67.4		**46.5**	**44.7**	42.8	43.2	39.3
0	tr11	**71.2**	**67.4**	61.1	64.1	-	tr23	**43.2**	**43.4**	41.3	36.8	-
5%		**73.4**	**71.4**	70.3	70.8	68.0		**46.8**	**47.0**	41.3	40.7	49.8
10%		**76.9**	**75.8**	74.1	73.6	73.5		**47.9**	**50.0**	43.0	43.2	50.7
15%		**77.9**	**77.6**	76.5	76.9	73.0		**53.5**	**52.7**	45.2	44.8	52.4
0	tr12	**68.6**	**68.6**	65.4	67.1	-	Multi10	**58.5**	58.0	35.2	**59.0**	-
5%		**73.6**	**71.6**	66.8	70.2	65.1		**63.3**	**65.6**	48.1	61.4	60.2
10%		**76.7**	**76.7**	70.5	72.9	70.5		**67.8**	**69.1**	58.7	63.8	66.6
15%		**78.4**	**78.5**	73.3	73.6	72.4		**72.6**	**72.8**	63.4	66.0	71.4

Table 4.4: Clustering results of label-based methods in *Accuracy*

label rate	**Data**	LMVS-I_R	LMVS-I_V	C-Sphkmeans	WNMF-NCW	GTAM	**Data**	LMVS-I_R	LMVS-I_V	C-Sphkmeans	WNMF-NCW	GTAM
0	classic	**64.8**	**72.4**	61.8	64.4	-	k1b	**82.5**	76.1	63.9	**86.0**	-
1%		**71.9**	**82.8**	71.7	68.2	65.7		**86.3**	81.4	80.1	**86.8**	65.9
2%		71.9	**83.0**	**72.1**	71.6	68.1		**88.9**	86.6	83.5	**87.2**	71.2
3%		**72.9**	**84.1**	72.5	72.4	71.8		**90.2**	**89.6**	87.0	89.4	73.0
0	re0	**39.9**	**40.0**	36.3	37.2	-	news	**71.4**	**73.9**	68.7	67.6	-
1%		42.0	**42.3**	40.0	40.2	**44.9**	groups	**74.4**	**73.6**	70.2	70.8	66.6
2%		43.7	43.6	**44.8**	43.6	**46.1**		**77.3**	**77.8**	76.1	72.6	68.8
3%		43.4	**47.9**	46.2	46.0	**46.4**		**81.4**	**81.2**	79.7	75.5	71.0
0	tr31	**66.3**	**70.0**	59.8	61.3	-	WPD6	55.3	**58.7**	53.7	**60.2**	-
1%		**70.5**	**72.4**	61.2	64.6	63.7		58.1	57.7	56.9	**62.4**	**68.6**
2%		**75.4**	**74.5**	65.8	66.9	67.9		59.9	**67.4**	60.6	64.0	**70.5**
3%		**75.6**	**77.5**	71.7	69.1	66.7		66.6	**68.4**	59.9	65.6	**70.9**
0	sports	**71.7**	**75.2**	69.7	70.2	-	CB	51.2	51.3	**58.2**	**64.1**	-
1%		**81.4**	**86.6**	72.2	77.4	75.1		**72.5**	**73.7**	68.0	68.3	67.8
2%		**84.8**	**86.8**	72.6	79.7	78.0		**77.0**	**77.4**	69.4	71.7	73.6
3%		**86.0**	**87.2**	74.4	80.8	78.4		**81.0**	**81.4**	75.3	75.1	79.3
0	reuters3	58.9	**70.4**	**75.5**	70.1	-	SM	53.2	52.3	**67.6**	**62.4**	-
1%		73.3	**82.3**	76.0	76.3	**88.1**		**83.2**	**82.3**	80.6	70.0	**95.3**
2%		79.2	**87.4**	76.4	76.5	**88.6**		**89.0**	**86.9**	82.8	73.2	**96.8**
3%		82.1	**88.1**	76.7	81.6	**90.0**		**90.9**	**91.7**	86.8	75.9	**99.8**
0	Reuters7	**70.4**	**71.1**	63.5	65.7	-	webkb4	51.8	52.8	**58.6**	**61.0**	-
1%		**72.6**	**76.7**	70.7	69.4	71.2		**70.9**	**71.6**	68.9	65.3	63.7
2%		**78.4**	**82.0**	72.7	74.6	75.8		**73.2**	**73.2**	71.1	69.4	64.6
3%		**78.8**	**82.7**	73.3	75.2	78.0		**76.1**	**74.5**	72.6	72.2	67.8
0	tr11	**71.2**	**66.0**	60.6	64.9	-	tr23	**48.9**	**46.1**	42.9	43.2	-
5%		**75.3**	**73.6**	72.2	70.0	71.5		**52.4**	**52.4**	46.1	46.7	51.4
10%		**81.7**	**79.6**	77.3	74.8	77.3		**55.7**	**58.2**	52.0	49.0	55.6
15%		**82.4**	**83.3**	81.9	76.3	77.7		**59.5**	**61.4**	54.3	52.1	55.3
0	tr12	**70.0**	**70.6**	64.8	66.3	-	Multi10	63.7	**65.6**	47.5	**64.7**	-
5%		**77.2**	**76.4**	70.8	73.5	66.8		**73.1**	**75.2**	63.6	70.3	68.0
10%		**82.6**	**80.6**	78.6	76.4	76.3		**78.2**	**80.9**	73.7	73.1	76.2
15%		**84.4**	**84.1**	78.9	78.6	77.3		**82.9**	**83.7**	78.0	75.8	81.8

Table 4.5: Clustering results of label-based methods in *F_Score*

label rate	**Data**	LMVS-I_R	LMVS-I_V	C-Sphkmeans	WNMF-NCW	GTAM	**Data**	LMVS-I_R	LMVS-I_V	C-Sphkmeans	WNMF-NCW	GTAM
0	classic	65.8	**73.4**	**68.7**	64.7	-	k1b	**87.3**	77.5	72.9	**87.2**	-
1%		70.5	**82.4**	**70.5**	67.4	66.8		**88.9**	85.3	83.6	**88.6**	72.2
2%		69.3	**81.8**	**70.9**	69.8	68.3		**90.7**	88.1	87.8	**88.4**	78.4
3%		68.7	**84.2**	**71.4**	71.3	70.2		**92.1**	89.4	89.1	**89.6**	82.6
0	re0	**46.0**	45.8	42.1	40.7	-	news	**76.4**	**77.6**	72.8	67.8	-
1%		48.7	**48.9**	47.8	47.4	**51.7**	groups	**77.4**	**77.2**	70.4	71.0	62.5
2%		50.5	**50.8**	**51.3**	50.2	50.8		**79.5**	**80.5**	76.5	73.1	67.5
3%		50.2	**54.8**	52.3	51.5	**53.0**		**81.8**	**82.4**	79.6	76.0	71.2
0	tr31	**72.8**	**78.0**	67.9	69.7	-	WPD6	**63.2**	**64.4**	61.3	60.7	-
1%		**76.3**	**73.5**	69.9	69.6	71.0		63.7	62.3	63.7	**65.3**	**71.8**
2%		**80.4**	**81.5**	73.8	70.9	74.7		65.6	**69.9**	66.5	65.1	**72.6**
3%		**83.7**	**82.3**	77.9	73.3	74.0		69.9	**71.1**	65.8	66.1	**71.5**
0	sports	**80.3**	**80.4**	70.2	68.7	-	CB	**66.2**	**66.2**	59.7	64.5	-
1%		**86.9**	**89.3**	78.2	84.0	81.1		**71.9**	**73.0**	67.6	68.0	68.6
2%		**87.3**	**90.2**	80.9	84.9	82.3		**76.9**	**76.9**	69.0	71.1	73.5
3%		**87.1**	**90.8**	82.3	85.2	84.0		**81.0**	**81.2**	75.2	76.2	79.4
0	reuters3	63.7	**75.2**	**74.3**	71.7	-	SM	**65.1**	64.1	**68.9**	61.9	-
1%		74.8	**81.8**	74.8	75.6	**88.3**		**82.9**	81.9	80.6	69.6	**96.2**
2%		80.4	**87.4**	75.3	77.0	**88.9**		**88.9**	86.7	82.8	73.0	**97.8**
3%		84.3	**88.1**	75.6	76.4	**90.2**		90.9	**91.6**	86.8	75.6	**99.8**
0	Reuters7	**77.4**	**77.5**	71.8	72.7	-	webkb4	56.8	56.8	**64.2**	**65.8**	-
1%		**78.6**	**80.7**	77.0	76.5	73.6		**72.6**	**72.8**	71.0	70.4	66.3
2%		**82.6**	**83.6**	78.1	74.6	77.2		**74.7**	**74.2**	71.1	72.0	67.4
3%		**83.0**	**85.5**	78.2	75.2	80.9		**77.1**	**75.5**	74.1	74.0	70.4
0	tr11	**74.9**	**72.8**	71.0	65.4	-	tr23	**56.0**	**55.3**	52.3	42.9	-
5%		**78.9**	**77.6**	72.2	71.3	75.3		**63.0**	**59.1**	52.8	50.4	62.1
10%		**83.6**	**82.6**	77.5	75.0	79.7		**62.9**	**64.8**	57.7	54.2	61.5
15%		**84.4**	**84.6**	81.9	78.7	80.1		**65.7**	**68.1**	60.3	54.2	63.5
0	tr12	**74.3**	**75.8**	71.5	68.3	-	Multi10	63.7	**65.6**	62.4	**66.3**	-
5%		**79.6**	**77.9**	73.9	75.6	72.1		**74.2**	**75.6**	63.4	67.3	69.6
10%		**83.0**	**81.2**	79.1	76.4	76.9		**79.0**	**80.8**	73.4	74.3	76.5
15%		**84.7**	**84.0**	80.0	77.7	78.4		**83.2**	**83.7**	77.9	76.1	81.8

Sph*kmeans* on *classic*, I_V gives comparable results to WNMF-NCW on *k1b* and some comparable results given in the low labeling level, LMVS obviously outperforms all the rest datasets. Secondly, with same limited amount of class labels, LMVS is able to lead to significant improvement, while C-Sph*kmeans* or WNMF-NCW may not. Sometimes, it achieves the best performance, even though the original MVSC did worse than either Sph*kmeans* or NMF-NCW, such as *CB, SM, webkb4.*

LMVS v.s. GTAM:

LMVS outperforms GTAM on 12 datasets significantly. Meanwhile, it is important to point out, most of the 12 datasets are either extremely unbalanced such as *tr31, k1b*, or have a relatively large *k* such as *newsgroups*. In general, these two kinds of datasets are more challenging to be handled compared with those relatively simple datasets which have only 2-3 clusters or equal balanced cluster sizes, e.g.: *SM* or *reuters3*. On the other aspect, GTAM achieves better results on 3 datasets, *WPD6*, *reuters3* and *SM*, and give comparable results on *re0*. In addition, from the experimental setting, we realize that LMVS is a simpler and more feasible approach without so many users' interference before the actual clustering process, and less demanding on the distribution on the labels compared with GTAM. In Section 4.3.2, we have briefly discussed why LMVS is able to handle most of the incomplete seeding

problem, compared with the existing semi-supervised clustering approaches. Meanwhile, GTAM directly fails if incomplete seeding occurs.

Impact on Incomplete Seeding

In this section, we would like to verify this strength of LMVS for handling the incomplete seeding issue through the experimental studies on certain benchmark datasets. Four datasets with relatively large *k* are selected from the 16 datasets for this purpose. The class size of *newgroups* and *Multi10* is equal balanced, while the class size of *tr11* and *tr12* is extremely unbalanced. In total, 2% random labels are used for *newsgroups*, and 10% random labels are used for the rest three datasets for all the experiments.

The impact to the clustering performance under incomplete seeding in *Accuracy* is presented in Figure 4.9. It is obvious that the performance degradation on two LMVS Clustering approaches is very insignificant compared with C-Sph*kmeans* and WNMF-NCW, as the cluster search for every document is benefited from a group of class labels through a more effective LMVS measure. Meanwhile, the documents assigned in clusters with no labels may be 'helpless' during the clustering process in C-Sph*kmeans* and WNMF-NCW; therefore, the overall clustering performance rapidly drops down. It implies that LMVS may be a good solution for categorizing those extremely unbalanced datasets.

4.6.4 PMVS Clustering Results

The performance of PMVS Clustering (denoted by PMVS only for short in this section) and necessary discussions are presented in this section. The results in *NMI*, *Accuracy and F_Score* on twelve datasets are shown in Table 4.6-4.8, and Table 4.9-4.11, corresponding to 2 different scenarios on the selection of pair-wise constraints. Four datasets: *re0, tr31, WPD6* and *newsgroups* are excluded from the experimental study here, as the constraints generated by scenario 1 may not always satisfy the requirement of carrying PMVS when only 1% or 2% labeled documents are provided.

Scenario 1

In this scenario, the equal numbers of *must-link* and *cannot-link* constraints are incorporated into the clustering process, and they are distributed over a small subset of the dataset only. First of all, comparing with the original MVSC results tabulated in Section 4.6.3, the results of PMVS presented in Table 4.6-4.8 still achieves significant improvement with only 1% constraints for 6 out of the 8 datasets, and shows continuous and consistent improvement when more constraints can be provided. It is not surprised that: for these relatively simple and large datasets, e.g.: *sports*, *CB*, *k1b*, PMVS is able to achieve comparable results to LMVS. This could be explained by recalling the mechanism of PMVS algorithm. When all the *must-link* constraints among the documents in ***ConS*** for each of the

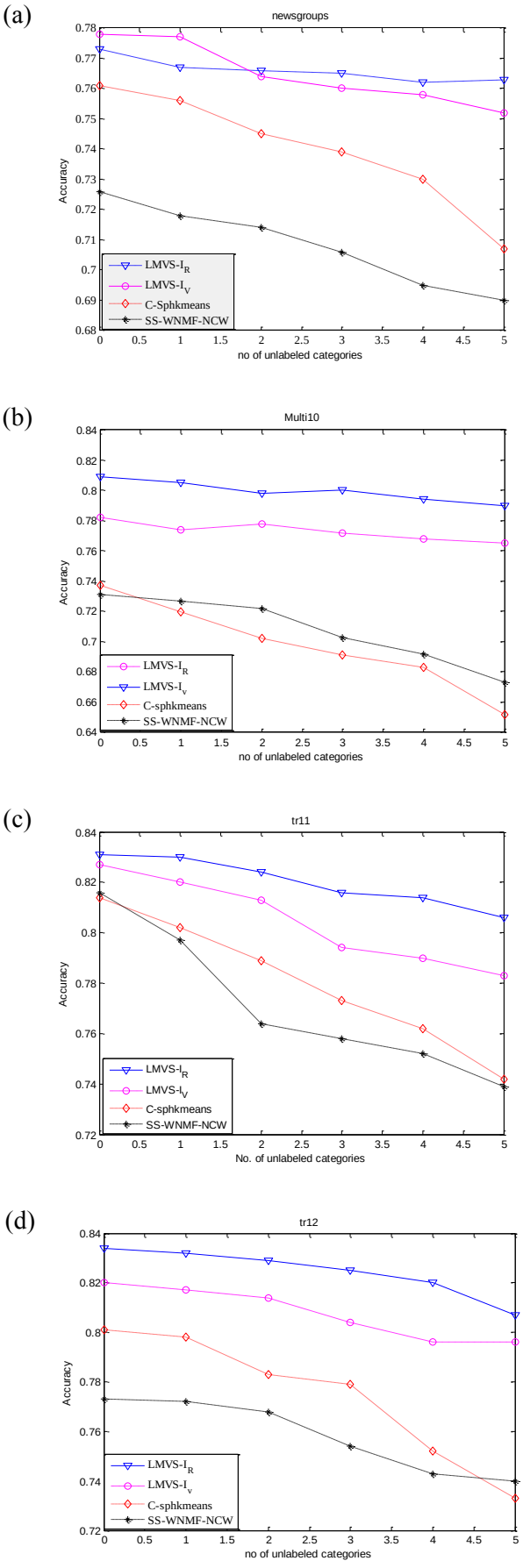

Figure 4.9: Performance comparisons in *Accuracy* under incomplete seeding

clusters are available, there will be only one move decision made for each cluster during the 2nd step of an iteration, as all these documents are now combined as an individual. Moreover, as these documents are strongly bonded together, they usually remain staying at the respective clusters, and get little chance to jump into another cluster during the Refinement process. Due to this characteristic, it could be clear to us now why we did not use the full *cannot-link* constraints that can be generated from the labeled documents to run the simulation. If we do so, PMVS would be almost equivalent to the LMVS. However, by selecting less number of *cannot-link* constraints, we are able to observe the impact of further reducing the number of appropriate viewpoints based on an approximate clustering process by LMVS, and the conclusion made on most of the datasets is that, as long as we are able to obtain a few appropriate viewpoints from the knowledge, even if we do not fully make use of the class labels, the performance on MVS-based clustering will be not affected much by the reduction of the number of viewpoints, which again fits in the inference we made in Section 4.4. Therefore, combining together with the observation from the results of LMVS, we found our inference does not only supported by the validity tests for the similarity matrices, but also successfully approved by the simulations on benchmark dataset.

Secondly, we could see PMVS also outperforms PC-sph*kmeans* and PMF on most of the datasets under Scenario 1. Only the result of I_R on *reuters3* and *classic* are slightly outperformed by PC-Sph*kmeans*. Meanwhile, on the other hand, PMVS may also suffer from the same setback problem as LMVS, when the number of *cannot-link* constraints is very small. It is noted the number of *must-link* constraints decides the total number of constraints. For example, in practical, given only 1% labels for *WPD6* may be not able to generate any *must-link* constraint, or for *Multi10* and *tr12* which hold a large *k*, the number of *must-link* constraints and *cannot-link* constraints may be also too small to provide sufficient multiple appropriate viewpoints for an accurate multi-viewpoint-based similarity assessment. Meanwhile, we also see that the performance rises rapidly and become competitive when just a few more constraints can be provided. Therefore, by avoiding the extreme case, our method is still considered powerful.

Scenario 2

In scenario 2, we randomly select the exact same number of total pair-wise constraints as we do in scenario1, but the distribution of the document is no longer restricted within a subset of the dataset. The conclusion we could draw from Table 4.9-4.11 looks a bit more complicated, though PMVS-I_R and I_V overall still lead higher performance compared to PC-sph*kmeans* and PMF on most of the datasets. Here we mainly focus on discussing the impact to the performance of PMVS using two different scenarios. In general, the improvement led by the increment of another 1% or 5% pair-wise constraints from one prior knowledge level

Table 4.6*:* Clustering results of Constrain-based methods using Scenario 1in *NMI*

Label rate	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF
1%	classic	56.9	**66.8**	**64.4**	44.3	k1b	**72.4**	71.3	**74.0**	71.5
2%		56.8	**67.5**	**64.7**	44.3		**74.1**	72.1	**74.7**	72.8
3%		58.1	**68.7**	**65.0**	44.5		**76.2**	**74.5**	74.3	73.4
1%	sports	**71.0**	**74,1**	67.2	70.2	reuters3	42.6	**53.7**	**54.3**	46.8
2%		**74.3**	**74.8**	67.3	70.6		52.8	**60.2**	**54.9**	47.7
3%		**74.7**	**75.9**	69.2	71.0		**55.6**	**63.6**	55.3	50.3
1%	SM	**37.0**	**41.5**	30.9	0.7	CB	**17.9**	**20.4**	5.1	11.4
2%		**54.3**	**51.4**	37.5	2.5		**23.4**	**26.7**	11.2	12.1
3%		**61.9**	**63.3**	47.3	14.4		**31.0**	**32.3**	14.7	12.5
1%	Reuters7	61.3	**63.2**	**62.5**	59.1	webkb4	**41.6**	**41.5**	39.3	33.6
2%		**63.6**	**64.7**	63.4	59.6		**43.0**	**42.7**	39.9	33.8
3%		**65.0**	**65.2**	63.5	60.3		**46.4**	**45.1**	41.2	38.1
5%	tr11	**72.6**	70.4	**71.9**	62.5	tr23	**43.2**	**43.8**	42.8	28.2
10%		**76.9**	**75.8**	73.4	63.1		**45.9**	**48.2**	44.7	32.3
15%		**77.2**	**76.9**	75.5	64.7		**51.8**	**51.4**	48.3	33.5
5%	tr12	**67.5**	67.0	**68.8**	64.3	Multi10	**60.4**	**61.9**	49.0	58.9
10%		**73.8**	**74.0**	72.0	58.9		**65.8**	**65.7**	55.8	63.1
15%		**76.2**	**76.5**	73.9	62.7		**70.6**	**71.4**	60.2	63.2

Table 4.7*:* Clustering results of Constrain-based methods using Scenario 1in *Accuracy*

Label rate	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF
1%	classic	68.8	**81.5**	**71.6**	63.4	k1b	**83.6**	79.7	80.6	**82.8**
2%		68.9	**82.3**	**71.9**	63.7		**86.1**	84.0	83.5	**86.0**
3%		69.7	**83.6**	**72.1**	63.6		**88.2**	87.5	82.2	**89.6**
1%	sports	82.0	82.4	66.5	70.3	reuters3	67.5	**77.6**	**75.6**	65.1
2%		87.4	86.8	69.5	72.7		75.3	**82.1**	**76.4**	68.9
3%		90.0	90.2	71.2	74.9		**80.2**	**85.6**	76.6	73.2
1%	SM	**79.3**	**83.2**	80.3	52.4	CB	**68.9**	**71.5**	60.8	69.4
2%		**83.1**	**85.0**	83.6	53.0		**75.2**	**76.3**	69.1	69.5
3%		**86.0**	**87.2**	86.5	52.6		**80.4**	**80.8**	71.1	69.8
1%	Reuters7	**71.2**	**73.2**	69.0	66.1	webkb4	**70.4**	**71.0**	66.7	59.4
2%		**75.3**	**78.1**	71.5	71.5		**72.8**	**72.6**	68.5	61.6
3%		**76.4**	**80.9**	72.9	73.6		**75.0**	**74.2**	70.7	66.3
5%	tr11	**70.6**	69.2	**73.7**	64.0	tr23	**48.7**	**49.4**	47.0	41.0
10%		**80.6**	**80.4**	78.6	68.2		**52.7**	**54.3**	50.4	43.7
15%		**82.4**	**82.1**	77.5	70.5		**56.5**	**58.2**	56.5	47.1
5%	tr12	72.1	**73.9**	**75.9**	71.0	Multi10	**69.4**	**68.2**	64.3	65.9
10%		**82.0**	**81.2**	81.0	68.9		**75.4**	**76.7**	70.5	70.4
15%		**83.7**	83.2	**84.7**	73.1		**80.2**	**81.8**	73.8	70.1

Table 4.8: Clustering results of Constrain-based methods using Scenario 1in *F_Score*

label rate	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans s	PMF
1%	classic	68.5	**80.9**	**70.3**	69.4	k1b	**85.8**	83.9	**85.2**	84.5
2%		69.0	**81.8**	**70.7**	69.8		**87.9**	85.1	87.1	**87.1**
3%		69.2	**83.2**	**71.0**	69.7		**89.8**	87.2	86.1	**90.2**
1%	SM	**81.7**	**84.0**	80.1	53.1	reuters3	58.9	**70.4**	**74.3**	66.0
2%		**84.7**	**87.8**	83.6	58.8		73.3	**82.3**	**75.2**	67.7
3%		**90.2**	**91.7**	86.6	59.5		75.2	**86.4**	**76.6**	73.3
1%	sports	**85.9**	**86.3**	75.8	75.6	CB	**82.1**	**88.1**	61.8	69.3
2%		**86.0**	**87.2**	78.3	80.4		**76.5**	**76.6**	67.5	69.1
3%		**86.9**	**88.8**	79.6	81.5		**80.6**	**80.9**	70.6	69.5
1%	Reuters7	**73.9**	**76.6**	75.7	67.9	webkb4	**71.3**	**72.0**	69.1	63.4
2%		**78.8**	**80.6**	77.2	72.0		**74.3**	**74.2**	70.5	65.0
3%		**81.4**	**83.0**	78.1	74.2		**76.7**	**76.2**	72.2	68.4
5%	tr11	**74.4**	72.1	**74.1**	69.0	tr23	**63.0**	**59.1**	55.9	45.5
10%		78.4	**78.4**	**81.6**	71.9		**62.9**	**64.8**	57.6	49.2
15%		**81.4**	**82.6**	80.0	74.4		**65.7**	**68.1**	62.6	51.9
5%	tr12	**75.6**	73.9	**76.4**	70.9	Multi10	**70.0**	**72.8**	64.0	67.7
10%		**81.1**	**81.4**	80.8	68.9		**73.7**	**75.3**	70.9	73.3
15%		83.0	**84.2**	**83.6**	73.1		**80.8**	**81.4**	73.9	73.5

Table 4.9: Clustering results of Constrain-based methods using Scenario 2 in *NMI*

label rate	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF
1%	classic	**65.4**	**68.2**	55.6	42.8	k1b	**72.2**	**70.7**	65.1	71.1
2%		67.3	**72.9**	68.4	**68.0**		**75.8**	**74.6**	67.8	74.2
3%		75.8	**88.4**	77.9	**92.7**		**78.9**	**80.4**	73.8	76.8
1%	SM	**63.5**	**65.7**	36.5	50.5	reuters3	47.0	**60.6**	50.6	**52.6**
2%		**100**	**90.6**	84.8	79.2		**62.3**	**67.6**	48.3	58.8
3%		**100**	**100**	95.4	88.3		**68.9**	**72.7**	55.2	65.8
1%	sports	**68.5**	**73.0**	66.1	43.9	CB	**29.7**	**32.6**	6.6	15.2
2%		**76.7**	**79.5**	74.1	71.2		**64.4**	**66.0**	30.4	29.9
3%		**86.3**	**87.2**	80.6	84.3		**96.0**	**96.8**	64.5	50.2
1%	Reuters7	**62.3**	**63.1**	57.6	58.6	webkb4	**39.9**	**40.5**	37.0	33.0
2%		**66.7**	**67.9**	58.2	61.4		**57.0**	**56.3**	47.1	49.4
3%		**73.8**	**75.2**	59.0	66.6		**64.2**	**66.8**	60.2	62.6
5%	tr11	**71.1**	**70.5**	65.9	66.5	tr23	**43.7**	**43.4**	37.2	32.4
10%		**81.8**	**79.3**	75.5	72.4		**54.6**	**56.9**	47.0	47.6
15%		**87.9**	**86.1**	81.7	79.6		**65.4**	**67.5**	62.6	61.3
5%	tr12	**71.4**	**70.0**	64.0	64.3	Multi10	**70.6**	**72.5**	41.4	62.4
10%		**78.9**	75.8	**76.8**	59.0		**78.4**	**80.0**	67.9	76.4
15%		**88.4**	**87.6**	87.3	62.7		**88.2**	**90.5**	87.4	84.0

Table 4.10: Clustering results of Constrain-based methods using Scenario 2 in *Accuracy*

label rate	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF
1%	classic	**72.5**	**75.8**	61.4	59.8	k1b	**75,6**	73.9	67.6	**74.8**
2%		84.8	**88.1**	**85.4**	78.0		**80.3**	**77.2**	69.1	76.1
3%		89.4	**96.3**	87.9	**98.0**		**83.6**	**85.4**	77.3	81.4
1%	SM	**91.6**	**92.3**	81.6	86.4	reuters3	72.3	**77.1**	**73.5**	69.9
2%		**100**	**99.6**	97.5	96.5		**87.2**	**90.0**	74.7	76.4
3%		**100**	**100**	99.3	98.3		**92.7**	**94.2**	81.4	87.3
1%	sports	**77.8**	**81.6**	64.9	54.7	CB	**80.8**	**81.8**	62.5	71.0
2%		**82.5**	**84.7**	68.2	78.2		**96.0**	**96.4**	80.3	80.5
3%		**88.8**	**91.4**	69.2	88.3		**98.6**	**99.2**	90.2	88.4
1%	Reuters7	**72.3**	**75.2**	62.1	68.9	webkb4	**62.8**	**64.7**	60.0	62.5
2%		**75.5**	**78.4**	61.6	71.8		**82.0**	**81.6**	75.6	75.2
3%		**82.4**	**84.7**	64.0	76.0		**85.4**	**88.2**	80.8	80.1
5%	tr11	**68.4**	**68.0**	64.1	67.4	tr23	**50.6**	**49.8**	43.9	42.8
10%		**82.9**	78.8	**80.4**	76.8		**60.3**	**63.4**	52.1	59.5
15%		**94.3**	**94.7**	92.0	86.9		**77.8**	**79.2**	63.5	73.8
5%	tr12	**75.4**	**73.5**	67.7	67.9	Multi10	**79.0**	**80.7**	51.9	70.2
10%		**81.6**	77.8	**79.6**	65.4		**84.3**	**86.4**	78.3	82.1
15%		**92.0**	**91.5**	89.4	68.7		**94.2**	**95.8**	92.5	88.4

Table 4.11: Clustering results of Constrain-based methods using Scenario 2 in *F_Score*

label rate	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF	**Data**	PMVS-I_R	PMVS-I_V	PC-Sphkmeans	PMF
1%	classic	68.5	**75.8**	**69.7**	62.8	k1b	**77.0**	**76.4**	73.9	76.3
2%		**86.9**	**89.8**	86.6	83.0		**84.2**	**82.6**	76.3	81.3
3%		91.6	**97.8**	94.3	**98.0**		**86.4**	**89.4**	83.0	86.0
1%	SM	**92.8**	**93.2**	81.5	81.5	reuters3	73.3	**78.0**	**74.0**	71.6
2%		**100**	**99.6**	97.6	97.6		**87.7**	**90.0**	73.4	76.3
3%		**100**	**100**	99.5	99.2		**93.0**	**95.1**	81.4	88.8
1%	sports	**83.4**	**84.6**	74.1	62.6	CB	**80.8**	**82.4**	63.3	70.8
2%		**85.5**	**88.0**	77.3	82.8		**96.0**	**96.4**	81.2	80.5
3%		90.7	**93.0**	78.7	**91.2**		**99.6**	**99.8**	93.2	88.3
1%	Reuters7	**75.3**	**76.2**	69.7	73.2	webkb4	**67.0**	**68,3**	65.2	64.8
2%		**79.5**	**80.4**	70.1	74.6		**84.7**	**82.0**	75.6	76.6
3%		**85.4**	**85.7**	72.3	79.2		**90.4**	**89.5**	84.8	81.6
5%	tr11	**73.6**	**73.2**	70.6	72.2	tr23	**63.0**	**59.1**	50.9	48.5
10%		**85.0**	80.6	**83.8**	79.2		**67.9**	**71.8**	60.4	65.6
15%		**94.4**	**95.1**	93.5	88.5		**84.1**	**85.1**	71.7	78.1
5%	tr12	**79.6**	**77.7**	72.0	71.0	Multi10	**80.7**	**82.0**	53.6	70.8
10%		**83,3**	80.2	**81.9**	68.9		**85.1**	**86.5**	78.3	83.3
15%		**93.6**	**93.2**	90.2	73.1		**95.4**	**96.0**	92.8	88.5

to its next level in scenario 2 is much higher than that in scenario 1, since the 'random selection' mechanism is generally able to cover much more number of documents in the dataset, compared to pre-select a small subset of the labeled documents for generating the pair-wise constraints. In this case, more *cannot-link* constraints lead to more appropriate viewpoints,and these viewpoints are shared by every document in the same cluster. Especially for those large datasets, e.g.: *sports*, *CB*, the number of constraints is relatively large enough to broadly cover most documents in the dataset; therefore, it is reasonable that the clustering performance under scenario 2 can be better than the performance under scenario 1 in most cases.

4.6.5 Actual Run Time

Besides the accuracy of two semi-supervised MVS-based Clustering, we now further evaluate the performance of them in terms of time efficiency. It is noted that, two MVS-based approaches and two Sph*kmeans*-based approaches are implemented in Java; therefore, the actual run time in millisecond for LMVS Clustering, PMVS Clustering under scenario 1 and the corresponding sph*kmeans* clustering on 12 datasets is reported in Table 4.12. Meanwhile, WNMF-NCW, GTAM and PMF are implemented in Matlab. Therefore the actual run time cost could not be directly compared with the MVS-based Clustering and is not provided in this section. Similarly to the previous sections, the value in bold and underlined is the best result, while the value in bold only is the second best for each specific prior knowledge level.

First of all, the runtime cost of the MVS based-clustering carried based on two different criterion functions I_R and I_V is compared. From the table, we could see I_V is always a better choice than I_R in term of time efficiency for both LMVS and PMVS Clustering. Then, comparing with C-Sph*kmeans*, LMVS-I_V is faster on 11 of 12 datasets, except *k1b*, and the runtime cost saving on most of the datasets is at least 40%. Similarly, comparing with PC-Sp*kmeans*, PMVS−I_V is also faster on all the 8 relatively large datasets, and the runtime cost saving is at least 50%.

Regarding the relationship between the runtime cost and the amount of prior knowledge, it is a bit difficult to draw a common conclusion. In general, the actual runtime cost can be reduced on most of the datasets when more knowledge is incorporated into the clustering process. However, both LMVS and PMVS Clustering converge slower than the original un-supervised MVSC on many datasets. The reason is LMVS and PMVS Clustering may need more iterations in the Refinement process to find a better cluster presentation of the datasets which corresponds to a better local maximal on the criterion function value, with the help of the prior knowledge; meanwhile, original MVSC may converge at a poor clustering result by less iterations due to a bad initialization. This reason may also explain why the runtime cost does not decrease with the increment of the available knowledge on some of the datasets.

Table 4.12: Actual Run Time of MVS Clustering and Sph*kmeans* in milliseconds

label rate	Data	LMVS-I_R	LMVS-I_V	C-Sph*kmeans*	PMVS-I_R	PMVS-I_V	PC-Sph*kmeans*
0	classic	**1951**	**1564**	8452	**1951**	**1564**	8452
1%		**4220**	**2161**	4434	**4432**	**2316**	13401
2%		**4820**	**2134**	4880	**5078**	**2293**	16307
3%		**5005**	**2143**	5055	**5356**	**2246**	21108
0	sports	**12203**	**10835**	24368	**12203**	**10835**	24368
1%		**31878**	**18649**	26615	**37661**	**18856**	73365
2%		**31760**	**19495**	35120	**35732**	**20261**	83385
3%		**30626**	**21034**	36796	**36347**	**22354**	92951
0	SM	**1023**	**780**	2032	**1023**	**780**	2032
1%		**933**	**698**	1612	**960**	**730**	1966
2%		**872**	**624**	1503	**902**	**714**	1957
3%		**831**	**630**	1673	**859**	**678**	1823
0	Ruters7	**1386**	**888**	1976	**1386**	**888**	1976
1%		**2139**	**1430**	2460	**2486**	**1432**	4816
2%		2288	**1233**	**1357**	**2365**	**1408**	8599
3%		2168	**1182**	**1392**	**2298**	**1276**	9914
0	k1b	**3309**	**2337**	3896	**3309**	**2337**	3896
1%		5960	**3400**	**3282**	5824	**3732**	**4618**
2%		4952	**3236**	**2777**	**5072**	**3508**	6095
3%		4472	**2798**	**2788**	**4735**	**2964**	5850
0	reuters3	**353**	**205**	867	**353**	**205**	867
5%		**349**	**296**	490	**336**	**302**	1077
10%		**313**	**215**	453	**352**	**246**	1088
15%		**308**	**205**	562	**307**	**232**	1001
0	CB	**1532**	**1204**	2573	**1532**	**1204**	2573
1%		**714**	**707**	1843	**834**	**793**	2654
2%		**651**	**617**	1756	**645**	**689**	1964
3%		**607**	**583**	1533	**635**	**660**	1786
0	webkb4	**3261**	**2367**	6410	**3261**	**2367**	6410
1%		**7633**	**3397**	6862	**8564**	**3879**	16329
2%		**6522**	**3309**	9249	**7548**	**3762**	18383
3%		**5787**	**3346**	13890	**6420**	**3415**	20230
0	tr11	2272	**849**	**1062**	2272	**849**	**1062**
5%		2240	**1393**	**1511**	2542	**1402**	**939**
10%		1302	**1152**	**1206**	1964	**1388**	**831**
15%		1156	**938**	**1127**	1567	**920**	**913**
0	tr12	**498**	**526**	831	**498**	526	**478**
5%		**1004**	**788**	1167	1256	**830**	**537**
10%		**819**	**612**	908	938	**680**	**496**
15%		**654**	**603**	891	814	**661**	**529**
0	tr23	**265**	**270**	284	**265**	**270**	284
5%		**569**	**447**	1203	584	**482**	**361**
10%		**519**	**441**	939	530	**461**	**446**
15%		**505**	**418**	876	508	**386**	**493**
0	Multi10	336	**268**	**298**	336	**268**	**298**
5%		**542**	**374**	555	587	**396**	**329**
10%		**414**	**310**	461	443	**342**	**284**
15%		**353**	**256**	382	398	**282**	**261**

4.6.6 Significance Test

To further examine the significance of the performance gain, each of the MVS-based approaches was paired up with those compared semi-supervised clustering algorithms for a paired t-test [173] with α=5%.

Given two paired sets **X** and **Y** of *Z* measured values, the null hypothesis of the test is that the differences between **X** and **Y** come from a population with mean 0. The alternative hypothesis is that the paired sets differ from each other in a significant way. In our experiment, these tests were done based on the evaluation values obtained on the nine datasets. The typical 5% significance level was used. If the t-test returns a *p*-value smaller than 0.05, we reject the null hypothesis and say that the difference of the performance between two algorithms is significant. In other words, we can say that the dominance of the LMVS or PMVS Clustering to the other clustering algorithms is significant. Otherwise, the null hypothesis is true and the comparison is considered insignificant. We test on the clustering results with 2% labels or equivalent constraints on the datasets having more than 500 documents and 10% labels or equivalent constraints on the datasets having 500 or less documents. The outcome of the test is shown in Table 4.13. The symbol ">>" indicates the algorithm in the row performs the algorithm in the column significantly better at the prior knowledge level, while ">" indicates an insignificant comparison. The values right below the symbols is the *p*-value of the *t*-test.

From the results of the *t*-test, the advantage of LMVS Clustering over WNMF-NCW and PMVS Clustering over PMF is obvious, as the performances are both statistically significant. A few special cases happened on the test between PMVS-I_R and PC-Sph*kmeans*. PMVS-I_R under scenario1 is not significantly better than PC-Sph*kmeans* based on *NMI* measure. Moreover, for the clustering carried on the other datasets by I_R, a relatively large *p*-value which is close to the threshold value is obtained if it is paired-up with PC-Sph*kmeans*, although it can be considered as statistically significant. Possible reasons could be found by referring the respective result tables. We could see that the results of PC-Sph*kmeans* in *NMI*

Table 4.13: Significance Test

		C-Sph*kmeans*	WNMF-NCW		PC-Sph*kmeans*	PMF	PC-Sph*kmeans*	PMF
NMI	LMVS-I_R	>>(>>)	>>	PMVS-I_R	>(>>)	>>	>>	>>
		0.036(3.6e-3)	1.1e-3		0.062(5.4e-3)	8.5e-3	2.7e-3	7.4e-4
	LMVS-I_V	>>	>>	PMVS-I_V	>>	>>	>>	>>
		3.3e-3	5.5e-3		2.3e-3	6.7e-4	5.3e-4	2.1e-4
Accuracy	LMVS-I_R	>>	>>	PMVS-I_R	>>(>>)	>>	>>	>>
		8.2e-3	1.8e-3		0.021(1.6e-3)	5.4e-3	0.024	4.6e-3
	LMVS-I_V	>>	>>	PMVS-I_V	>>	>>	>>	>>
		5.6e-4	2.1e-4		3.1e-3	2.2e-3	4.2e-3	1.8e-3
FScore	LMVS-I_R	>>	>>	PMVS-I_R	>>(>>)	>>	>>	>>
		6.5e-3	1.2e-3		0.035(1.3e-3)	2.8e-3	9.4e-3	2.1e-3
	LMVS-I_V	>>	>>	PMVS-I_V	>>	>>	>>	>>
		2.8e-3	6.2e-4		2.8e-3	1.2e-3	3.2e-3	8.1e-4

on *classic* and *reuters3* are both higher than that on I_R, although the results in *Accuracy* or *F_Socre* on *classic* are considered comparable. In this case, we also report the *p*-values in the bracket only based on the performance of all the rest datasets in *NMI* measure, where *classic* and *reuters3* were removed as "outliers". Under this circumstance, a much smaller *p*-value is obtained and LMVS-I_R or PMVS-I_R is confirmed to outperform C-Sph*kmeans* or PC-Sph*kmeans* significantly.

4.7 Conclusions and Future Work

In this chapter, we present how a novel and effective semi-supervised clustering framework (SS-MVSC) can be developed by directly incorporating some available prior knowledge into the MVS manner and making it immediately applicable to the clustering tasks, instead of making use of the knowledge to enhance the similarity measure through an independent DML algorithm before the actual clustering process is carried out in the traditional ways.

The key success factor of this work is to obtain multiple appropriate viewpoints from the dataset with the help of prior knowledge for a more effective similarity measure between two documents. Two similarity measures have been formulated accordingly, by utilizing two different types of prior knowledge available to user: class labels and pair-wise constraints, named by LMVS and PMVS, respectively. In LMVS measure, only the labeled documents can be served as the appropriate viewpoints; while in PMVS measure, the appropriate viewpoints can be obtained from the pair-wise *cannot-link* constraints between a few documents first, and then shared with every document in the same cluster. Two new clustering criterion functions, I_R and I_V have been formulated from each measure accordingly. Theoretical analysis on the these functions and empirical study through validity tests have also been conducted to ensure the new clustering approachess are able to make good use of the prior knowledge for similarity measure, other than guiding the cluster search. Meanwhile, a few potential issues in the original MVSC due to misleading of the inappropriate viewpoints are successfully addressed. Lastly, extensive empirical studies conducted on a number of benchmark textual datasets with various amount of the prior knowledge level under different evaluation metrics, demonstrate the great potential and merits of the proposed semi-supervised MVS clustering framework.

As summarized above, it would also be interesting to explore other novel methods to find out the more appropriate viewpoints, other than making use of class labels or pair-wise constraints, e.g.: active selection, ranking or develop more sophisticated label construction techniques. In addition, we may consider that whether the MVS measure can be also effectively applied to other types of clustering, such as hierarchical clustering, fuzzy clustering, and therefore formulate other new clustering criterion functions to deal with other in real world applications or problems.

Chapter 5

Applications: Semi-Supervised Clustering for Sentiment Data Analysis

5.1 Sentiment Analysis

In recent years, web content generated by individual users on the Internet in the form of weblogs (Facebook, Twitter), discussion forums and online review sites increases extremely rapid. Millions of people log into their Facebook, Twitter account or other particular review sites, wish to make a public voice to share their opinions to share information, or to post their feeling or personal experience on a particular product or service through the Internet. These pieces of information are then viewed by millions of the other social network users through the Internet. At the moment of getting useful advice from these comments, we realized the important part of this phenomenon is that these comments actually present both huge opportunities and challenges to both sides: not only benefit the consumers, but also make sense to the supplier (company). To be more specific, for supplier, this user-generated content provides a rich source of implicit consumer feedback, like *tripadvisor*, *Amazon*. Tracking the comments written on these sites enables companies to look into how the customers evaluate their products or service, which provides useful insight on how to make improvement or market products better. For consumers, the feedback and opinions from diverse sources guide them, to aid in making more realistic decisions based on the recommendations. These decisions could range from which new mobile phone to buy, which movie to watch, which hotel to stay, and so on.

On the other side, the consequent challenge for both roles is to be able to analyze the vast quantity of available information such as: thousands of comments, for example: how to quickly collect and summarize the meaningful insights therein based on texts but not a simple score? The process of analyzing textual information, extracting meaningful patterns and discovering opinions, from which to support appropriate and fact-based decision making, is then called Sentiment Analysis [174].

The research on sentiment analysis remains a challenge task. This is because handling the unprecedented scale of unstructured user-generated web content requires new methodologies which are able to incorporate into natural language processing and machine learning techniques.

Data mining is also a useful technique for Sentiment Analysis. It refers as “Opinion Mining” when handling the sentimental data . For example, Twitter Sentiment - a result of a Stanford classroom project - is a tool that allows people to discover the feeling about a product, brand or topic by collecting and classifying tweets. For example, if you are going to purchase an iPhone, and wonder what other people think about this product, you can have Twitter Sentiment find it out for you, as shown in Fig. 5.1. Or you want to take a look of the evaluation on a popular movie; IMDB is a good website to check the scores and detailed feedbacks from others. Two screen-captures on a remarkable and a terrible movie as examples are given in Fig.5.2 and 5.3, accordingly.

One key component of this process is to estimate the sentiment expressed in the user’s comments around selected topics of interest or a more specific characteristic in a service

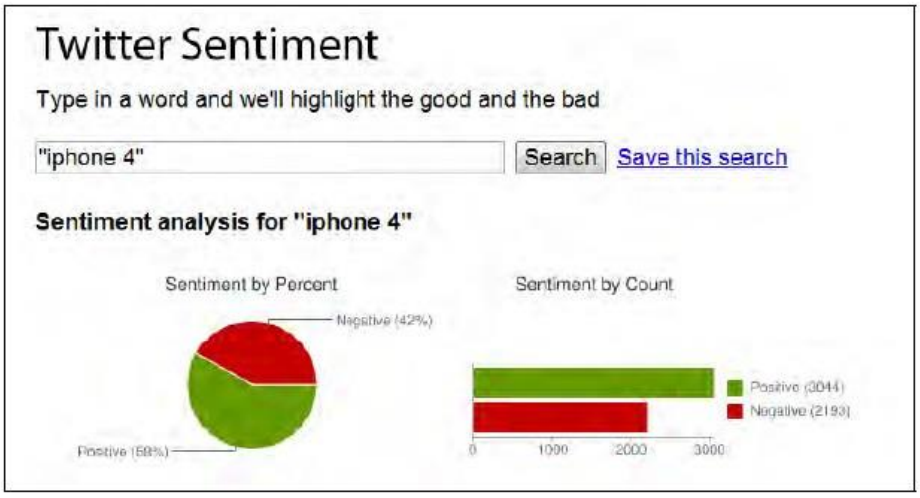

Figure 5.1: Twitter Sentiment from a Stanford academic project

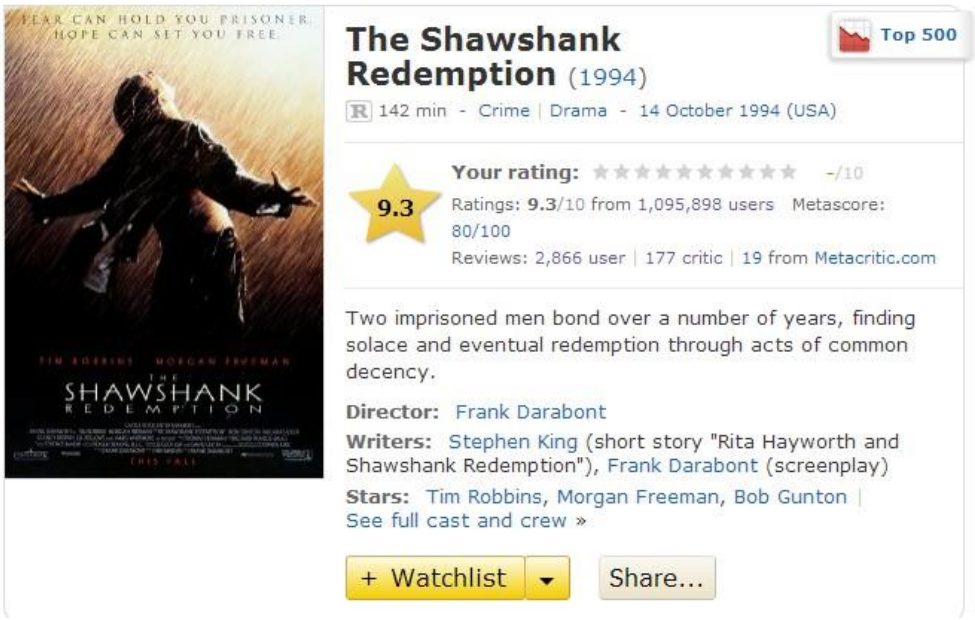

Figure 5.2: One of the top 100 movies scored by IMDB

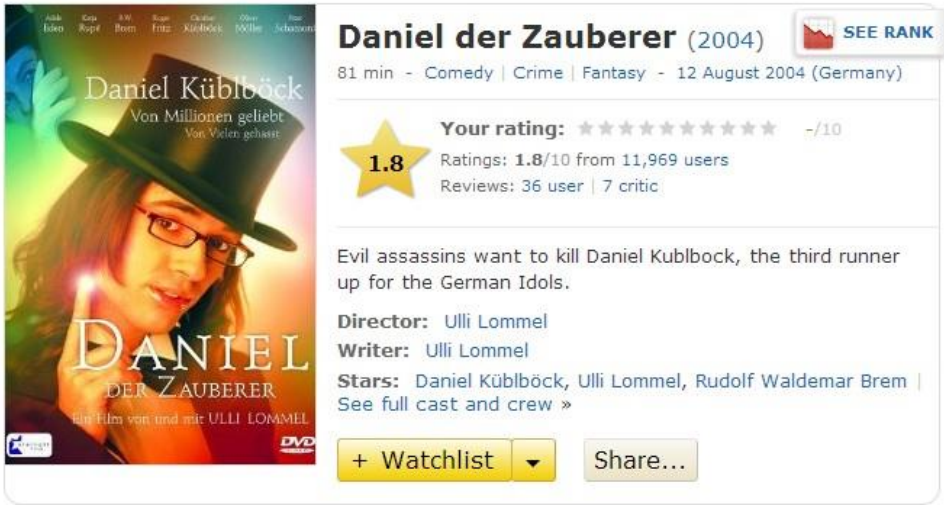

Figure 5.3: One of the bottom 100 movies scored by IMDB

rather than the overall score. The emerging area of sentiment analysis focuses on the task of automatically identifying if a piece of text delivers a positive or negative opinion towards a subject matter. However, it is mentioned in [175], "detecting the sentiment expressed in documents tend to be an extremely difficult task, and the performance of sentiment classifiers can vary a great deal depending on the domain of application". Recently, researchers aim to build a smart and robust system that provides insights across a broad list of different products and topics of interest for sentiment analysis and such a system needs to be as flexible as possible in order to adapt to new domains of the products or services with minimal supervision**.**

In this chapter, we would like to see how the proposed DSS-HFCR can be applied to Sentiment Analysis, and how well it can perform compared with other state-of-the-art techniques under the same condition.

5.2 DSS-HFCR for Sentiment Text Corpus

5.2.1 Sentiment Word Labeling for Movie Review

movies_reviews is a very popular dataset in sentiment analysis literature [176]. It consists of 1000 positive and 1000 negative movie reviews drawn from the IMDB archive of the *rec.arts.movies.reviews* newsgroups. The raw dataset is pre-processed by rainbow and the vocabulary size is kept to 5000, by selecting the top 5000 words having the highest mutual information gain. To incorporate useful label information in the word domain, a deep study on the *movie_reviews* dataset itself is compulsory. We scan the vocabulary list very carefully, and find out all the sentiment words, and categorize them into either the "positive" or the "negative" group. Some of the selected words have been listed in Table 5.1. The number in the bracket indicates how many documents contain this word in the dataset. However, it is noted a 'positive' word may also appear in those 'negative' reviews, while a 'negative' words

Table 5.1: Selected Sentimental Words in *movie_reviews*

Word Category	Document Category	Representative Sentimental Words
positive	positive	{*perfect(182), nicely(64), excellent(110), effective(123), wonderful(126), smooth(21), powerful(104), sincere(20), strengths(14), finest(33), lovingly(14),innocence(33), best(489), intelligent(98), memorable(27), endearing(33) , remarkable(46), strong(148)*}
	negative	{*perfect(75), nicely(29), excellent(35), effective(46), wonderful(0), smooth(5), powerful(48), sincere(4), lovingly(0), strengths(2), finest(5), innocence(0), best(365), intelligent(55), memorable(101), endearing(10), remarkable(18), strong(148)*}
negative	positive	{*bad(259),boring(48),fails(37),incompetent(4),lifeless(7),nonsense(8), painful(16),poorly(14),ridiculous(21),stupid(39),tedious(10),waste(21), worst(44)* }
	negative	{*bad(515),boring(169),fails(96),incompetent(16),lifeless(34),nonsense(29), painful(41),poorly(73),ridiculous(108),stupid(155),tedious(33),waste(108), worst(194)* }

also appear frequently in those 'positive' reviews. For example, by referring to the ground truth, we realize while *bad* appears in 515 negative reviews, it also appears in 259 positive reviews, maybe just behind a "not" to present a positive comment. However, this information is definitely unknown to the user for a real-life categorization task. Those counting given in the bracket in Table 5.1 are just for demonstration purpose to ask the users pay a lot of attention to pick up other sentiment words like *bad*, *strong*, and avoid giving an 'positive' or 'negative' label to them. In other words, the initial membership of those words should not be simply set to 1 and 0.

5.2.2 Experimental Settings

Other than the OSS-NMF [134] mentioned in Chapter 3, another non-negative matrix tri-factorization based approach [131] (SS-MFLK), a document-word bi-partite graph based co-regularization approach [138] (DWCR) with lexical knowledge for sentiment analysis and semi-supervised latent Dirichlet allocation [83], Green Function [177] are selected for the comparison purpose. Moreover, through these comparisons, we also indirectly compared with DA [178], SVM and Transductive SVM [179] which is given in[138]. Among these approaches, the clustering performance when only the lexical knowledge can be incorporated into the clustering process is also reported in OSS-NMF and MFLK. Since we are unable to implement all of the compared algorithms by our own, some of the results presented in later sections are directly cited from the respective paper. To guarantee a fair comparison, we strictly follow the same experimental settings given by the compared algorithms. The stopping threshold is set to 1e-5, and maximum number of iteration is equal to 200. As we just discussed in Section 5.2.1, the initial membership assigned in the 'positive' cluster for those 'positive' words like *effective*, *smooth* and the initial membership assigned in the 'negative' cluster for those 'negative' words like *bad*, *upset* are all set to 0.75. Meanwhile, if

the user is confident enough for some of the words, such as *lovingly*, *waste*; the initial membership for this kind of words can be directly set to 1 in the positive or negative cluster, respectively, even though they may still appear in those reviews giving an overall opposite sentiment opinion.

5.2.3 Results and Discussions

Firstly, we investigate how the clustering performance is affected when only the lexical knowledge from the word domain is incorporated. We pick up all the sentiment words can be labeled and randomly select some of them to observe the performance on *movie_review*. The ratio of the words in use gradually increased from 20% to 100%, and the performance in *Accuracy* measure is presented in Table 5.2. Among the three algorithms, it is obviously our proposed SS-HFCR-W outperforms ONTMF-W, and gives comparable results with SS-MFLK-W. Unlike SS-MFLK-W which only shows a rapid 'rise up' when the lexical knowledge is fully used, the performance of SS-HFCR-W is improved much smoother against the increment of the labeled words. After that, we also test how good the performance is when the prior knowledge from both document and word domain is working together during the clustering process, and help each other iteratively to retrieve a better cluster representation. We now make use of the full lexical knowledge and gradually increase the ratio of the labeled documents. The performance comparisons among 6 different clustering algorithms are presented in Table 5.3. We find out that the performance of DSS-HFCR can be significantly improved based on what we have achieved in Table 5.2. Meanwhile, DSS-HFCR significantly outperforms four algorithms when the label documents is less than or equal 30%, except DWCR. When the ratio of the label documents continuously increases up to 50%, the performance of DSS-HFCR is still much better than 3 approaches, and comparable to SS-LDA. Next, we also show the performance comparison in *Accuracy* between DSS-HFCR and DWCR when the ratio of labeled documents increases from 2.5% to 20% in steps of 2.5%, to further verify the potential advantages of using DSS-HFCR for the *movie_review*. The results given in Table 5.5 shows the performance of DSS-HFCR is always slightly better than DWCR.

5.3.4 Analysis on Sentiment Words

From the results tabulated above, it is claimed that only incorporating the lexical knowledge does not help the clustering much. There are several possible reasons about it. Firstly, since the amount of the words can be labeled is usually a very small portion compared to the size of the whole vocabulary set. Therefore, assisting by only a few labeled words does not give strong influence to the clustering process. Secondly, selecting the "positive" or "negative" words by a lexical classifier has some natural drawbacks. The lexical classifier will label a

Table 5.2: *Accuracy* with increasing fraction of sentiment words labeled on *movie_review*

Algorithm	0	20%	40%	60%	80%	100%
SS-HFCR-W	**59.2**	**61.4**	**65.3**	**66.2**	**66.8**	68.2
OTNMF-W	56.5	58.1	60.0	58.4	58.8	61.6
SS-MFLK-W	50.6	56.2	58.9	62.0	66.4	**73.9**

Table 5.3: *Accuracy* with increasing number of labeled documents on *movie_review*

Label rate	**DSS-HFCR**	**OSS-NMF**	**SSMFLK**	**DWCR**	**SS-LDA**	**Green Function**
10%	**76.2**	64.6	60.0	74.2	69.4	55.1
20%	**79.3**	65.0	62.7	77.6	75.5	65.2
30%	**83.6**	73.1	67.0	-	79.6	65.5
40%	**85.4**	75.8	73.1	-	83.7	68.3
50%	**87.1**	78.9	80.1	-	86.7	77.5

Table 5.4: *Accuracy* with increasing number of labeled documents on *movie_review*

Algorithm	2.5%	5%	7.5%	10%	12.5%	15%	17.5%	20%
DSS-HFCR	**72.6**	**73.8**	**74.3**	**76.2**	74.2	**77.7**	**79.0**	**79.3**
DWCR	70.4	71.1	73.0	74.2	**75.0**	76.3	76.4	77.6

document as the positive one if there are more positive lexicon terms than negative lexicon terms in those documents, regardless about the frequency value. Moreover, it is not to distinguish the "properties" of a selected word. To understanding the sentiment meaning of a word it really carries in the document, it also requires context-based analysis. Therefore, sometimes it is not easy to decide the initial membership of a selected word just using binary values.

The fuzzy co-clustering allows a continuous membership value between 0 and 1 to be assigned on a sentimental word, to indicate whether it is more on the "positive" side or "negative" side. This is also one of the reasons that why we prefer to using a more time consuming way: the human judgment to prepare the lexical knowledge.

The final word memberships obtained by DSS-HFCR on some of the selected sentimental words in the respective clusters are now presented in Table 5.5 and 5.6, respectively. It implies that a good clustering result on the document domain always comes together with a convincing clustering results on the word domain.

To further elaborate how the sentiment words work, we look into the other sentiment words selected in the *movie_review*. This set constitutes the terms that have changed most dramatically in sentiment: *lone, origin, basic, show, doubt, revolut, pretti, know, reason, captur, complet, complex, talent, upset, secur, debat, critic, plain*. This analysis is able to help us to understand the domain-specificity of certain words with sentiment, which may be not possible to encode into a single general-purpose lexicon, but by context. For example, words such as *capture, revolution,* and *complex* can be associated with positive experiences in descriptions of movies, though they may be considered negative in other fields. Pang et al.

Table 5.5: The final membership on selected sentiment words with (0.75/0.25) initial membership

Word Type	Membership						
Positive	best	effective	pretty	meaningful	revolut	strong	talent
	0.902	0.897	0.804	0.938	0.893	0.925	0.687
Negative	bad	doubt	painful	upset	worse	weakness	worse
	0.843	0.756	0.904	0.758	0.912	0.856	0.798

Table 5.6: The final membership on selected sentiment words with (1/0) initial membership

Word Type	Membership							
Positive	excellent	finest	innocent	memorable	sincere	smooth	strength	wonderful
	0.933	0.943	0.864	0.995	0.928	0.934	0.910	0.996
Negative	boring	dull	lifeless	incompetent	fails	stupid	waste	ridiculous
	0.969	0.989	0.978	0.966	0.975	0.994	0.988	0.990

[32] observed *the move_review* data in very detailed level, and point out that "the down-weighting of positive lexicon terms, such as *talent* for MOVIES is also consistent with the 'thwarted expectation' narratives ".

5.3 Conclusion

In this chapter, we studied how to apply the proposed DSS-HFCR to the web sentiment analysis, when some lexical knowledge and document domain knowledge are available to the user. To be more specific, the lexical knowledge usually comes from a few sentiment words which can be treated as the key information clearly representing people's mind. The experiments show that a better sentimental analysis result on the *movie_review* dataset can be achieved by DSS-HFCR compared to some existing dual-knowledge based clustering/learning approaches.

Chapter 6

Conclusions

6.1 Summary of Research

The research work presented in this thesis emphasizes on the development of novel semi-supervised clustering approaches for categorizing large scaled, high-dimensional textual data collection. Other than proposing new concepts and frameworks, developing effective and efficient algorithms, we also demonstrate that the proposed work is applicable to solve real-life problems. In this chapter, we summarize the work in the following paragraphs.

In Chapter 2, we have conducted a literature survey of the important background knowledge in the field of text clustering. It includes a variety of existing clustering algorithms and systems, together with their applications in various fields, especially in the domain of semi-supervised clustering. We have also discussed some critical problems that researchers have encountered when dealing with high-dimensional textual data. How to address these problems remains as the big challenges in the past decades, and lead us to the motivation of the research work going for semi-supervised clustering in this thesis.

In Chapter 3, we proposed three semi-supervised clustering approaches under the fuzzy co-clustering framework by incorporating a very small amount of prior knowledge into the clustering process. The knowledge can be a small group of class labels or pair-wise constraints from the document and word domain. Two partitioning-ranking based approaches SS-FCL and SS-FCC are developed by incorporating a few labeled documents or a few pair-wise constraints from only the document domain, respectively. Other than the categorization results for the documents, the output of these two approaches also includes a group of work ranking clusters, which is beneficial for other data mining techniques, such as text summarization. Then, a heuristic dual-partitioning based approach DSS-HFCR is also developed to make fully use of the available knowledge from both document and word domain. Extensive experimental studies have been conducted over a number of benchmark textual datasets. The strength of the proposed methods in terms of improved accuracies, time efficiency and stability is successfully verified through the performance comparison with a few popular label-based/constrained-based semi-supervised clustering approaches.

In Chapter 4, we reported the semi-supervised clustering framework using multi-viewpoint based similarity measure. Using cosine similarity as the basis, we study how to obtain multiple appropriate viewpoints with the help of two types of prior knowledge: class labels or *cannot-link* constraints to formulate a more informative and effective similarity measure and immediately apply it to the clustering task. In other words, the effort normally required by applying an independent distance metric learning algorithm before the clustering algorithm itself for such a goal can be avoided. Two measures are proposed under this novel framework. In LMVS measure, only the labeled documents are used as the appropriate viewpoints; while in PMVS measure, the appropriate viewpoints are obtained through the pair-wise *cannot-link* constraints between a few document pairs and a sharing process. The strength of the measures is shown through some validity tests. Then, two MVS-based clustering approaches, named by LMVS & PMVS Clustering, are proposed respectively. Theoretical analysis on the clustering criterion functions is also provided to ensure the new clustering approachess are able to make good use of the prior knowledge for similarity enhancement, in addition to the cluster search-guiding purpose. Meanwhile, some existing issues raised from the un-supervised MVSC, such as the misleading effect during the clustering process caused by the inappropriate viewpoints are successfully addressed. Lastly, extensive experimental studies conducted on a number of benchmark textual datasets with various amount of the prior knowledge level under different evaluation metrics, demonstrate the great potential and merits of the proposed semi-supervised MVS clustering framework.

Finally, in chapter 5, we apply DSS-HFCR to web sentiment data analysis. Through the experimental study on a popular *movie_revew* dataset, we show DSS-HFCR is able to effectively utilize the lexical knowledge which comes from a few sentiment words labelled by human judgment together with the available prior kowledge from the document domain, and signficantly improve the perfomance of the sentimental data categorization.

Other than that, our studies should be applicable on other types of high-dimensional data other than the text corpus, such as gene microarray data in bioinformatics field. Future extension of our studies to these types of data and application domains will definitely provide more insights into other facets of the proposed techniques. Nevertheless, a long and tough journey of research is about exploring new knowledge. We hope that the research presented in this thesis is able to address a few challenging problems encountered in the high dimensional data clustering field, and also helpful to inspire someone else who is also working in this area. In such way it brings us a few steps closer to more effective and efficient clustering in the end.

6.2 Future Work

The research achievements reported in this thesis is about developing new semi-supervised clustering techniques for high dimensional textual data categorization. We have also specified

a few potential directions that could be continued for future exploration in the end of every main chapter. In this section, we are going to systematically summarize them and emphasis the respective significance again.

Integration of Domain Knowledge

Our studies in this thesis focus on the semi-supervised clustering, i.e. the techniques between un-supervised learning and the supervised learning. In general, it carries the advantage of the un-supervised learning which is able to handle huge amount of data, and also avoid requesting the training process which is usually very expensive in other supervised clustering. Taking both the effectiveness and the cost into consideration, semi-supervised clustering aims for effectively improve the performance with only a very limited demanding on the prior knowledge. Therefore, the way of incorporating the knowledge becomes the critical impact factor to the success of a semi-supervise clustering method. Through our thesis, we mainly make use of class labels or pair-wise constraints, other forms of the domain knowledge or supervision, such as triple constraints, label propagation; manifold may be the future work for our approaches. Other than that, looking for multiple source of the knowledge is also a very important direction, such as study on how the knowledge from the word domain could be maximally utilized.

Number of the Clusters

As we all understand, many research studies on data clustering are carried out based on some strong assumptions. Like in the generative algorithm, the assumption made is usually related to a particular statistical model. For our cases, just like most partitioning clustering approaches, the number of clusters k needs to be specified in all the proposed approaches. Meanwhile, there have been quite a number of proposed works that aims to find the natural number of data clusters automatically [180]. However, we can expect this to be a very challenging task due to the high computational cost and specific understanding to the application itself in finding k. Therefore, this kind of algorithms is usually not generic enough to use. The fact is that for any collection of data, there is always more than one way to perceive and divide into groups. For example, given the same set of documents, it can be categorized into different sub-topics, depending on different angle of views, different understanding of different people. In the end, there is not yet any method that can guarantee to claim about the correct number for every dataset. To our knowledge, the objective formulation of fuzzy co-clustering model has a common characteristic, it enforces the number of document cluster and word cluster must be equal. However, this may lead to a potential limitation to the real applications, especially when the knowledge from the word domain is available but the number of the word category in the reference is not equal to the number of document cluster k. We appreciate the future research effort made on releasing the fuzzy co-

clustering from this restriction. This will make the current proposed methods much more flexible and applicable to the real-world problems.

Online Algorithms

So far, we only developed algorithms in a batch mode for a given dataset. To handle incremental data or online data, online versions of the proposed algorithms need to be further developed. More efforts can be made on the dynamic clustering of web data [181, 182], such as news, social networks as well.

Other Applications

In this thesis, we have applied the proposed clustering approaches for web textual document categorization and sentiment data analysis. In the future, other application fields may be explored, including text mining in the social network, gene expression, categorization of the protein structure, image pattern recognition and so on. We believe the effectiveness of the proposed methods on the textual documents can be also benefited these applications, for example: correctly grouping the gene simultaneously with grouping the DNA slice using the FCC model.

Author's Publication

Journal

1. Y. Yan, Y. Wang, L. Chen, "Fuzzy Clustering with Multi-Viewpoint based Similarity Measure", Under Preparation to IEEE Transactions on Fuzzy Systems

2. Y. Yan, L. Chen, "Constrain-based Clustering with Multi-Viewpoint based Similarity Measure", Under Preparation to IEEE Transactions on SMC.

3. Y. Yan, L. Chen, "Multi-Viewpoint based Similarity Measure and Criterion Functions for Semi-Supervised Document Clustering", Submitted to IEEE Transactions on Knowledge and Data Engineering.

4. Y. Yan, L. Chen, and W. C. Tjhi, "Semi-supervised fuzzy co-clustering algorithm for document categorization," *Knowledge and Information Systems*, vol. 34, pp. 55-74, 2013.

5. Y. Yan, L. Chen, and W. C. Tjhi, "Fuzzy semi-supervised co-clustering for text documents," *Fuzzy Sets and Systems*, vol. 215, pp. 74-89, 2013.

Conference

1. Y. Yan, L. Chen and C, Chan, "MVS based Clustering," accepted by *9th International Conference on Information, Communication and Signal Processing*, 2013.

2. Y. Yan, L. Chen, and D. T. Nguyen, "Semi-Supervised Clustering with Multi-Viewpoint based Similarity Measure," International Jointed Conference on Neural Network, 2012.

3. Y. Yan and L. Chen, "Label-based semi-supervised fuzzy co-clustering for document categorzation," in *8th International Conference on Information, Communication and Signal Processing*, 2011.

4. Y. Yan and L. Chen, "Hyperspherical possibilistic fuzzy c-means for high-dimensional data clustering," in *7th International Conference on Information, Communication and Signal Processing*, 2009.

Bibliography

[1] D. T. Nguyen, L. Chen, and C. K. Chan, "Clustering with Multi-Viewpoint Based Similarity Measure," *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, 2011.

[2] U. M. Fayyad, *Advances in Knowledge Discovery and Data Mining AAAI/MIT*, 199

[3] C. J. v. Rijsbergen, "Information Retrieval, 2nd Edition.," 1979.

[4] T. D. Wu, "Symptom clustering and syndromic knowledge in diagnostic problem solving," in *Proceedings - Annual Symposium on Computer Applications in Medical Care*, 1989, pp. 45-49.

[5] S. Lee, "Efficient multistage approach for unsupervised image classification," in *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2005, pp. 3852-3855.

[6] M. Filippone, F. Masulli, S. Rovetta, S. Mitra, and H. Banka, "Possibilistic approach to biclustering: An application to oligonucleotide microarray data analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 4210 LNBI, ed, 2006, pp. 312-322.

[7] M. Paola, H. James, S. Gerold, and W. Eric, "P. Merlo/J. Henderson/G. Schneider/E. Wehrli: Learning Document Similarity Using Natural Language Processing," 2003.

[8] G. Salton and M. J. McGill, "Introduction to Modern Information Retrieval, ch," *Text analysis and automatic indexing*, pp. 59-71,201-211, 1983.

[9] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, pp. 513-523, 1988.

[10] Y. Zhang, N. Zincir-Heywood, and E. Milios, "Term-based clustering and summarization of Web page collections," 2004, pp. 60-74.

[11] F. Beil, M. Ester, and X. Xu, "Frequent term-based text clustering," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 436-442.

[12] Y. Miao, V. Kešelj, and E. Milios, "Document clustering using character N-grams: A comparative evaluation with term-based and word-based clustering," 2005, pp. 357-358.

[13] M. Shafiei, S. Wang, R. Zhang, E. Milios, B. Tang, J. Tougas, *et al.*, "Document representation and dimension reduction for text clustering," 2007, pp. 770-779.

[14] V. W. Feng and G. Hirst, "Text-level discourse parsing with rich linguistic features," 2012, pp. 60-68.

[15] S. Joty, G. Carenini, and R. T. Ng, "A novel discriminative framework for sentence-level discourse analysis," 2012, pp. 904-915.

[16] "http://wordnet.princeton.edu/.".

[17] D. R. Recupero, "A new unsupervised method for document clustering by using WordNet lexical and conceptual relations," *Information Retrieval,* vol. 10, pp. 563-579, 2007.

[18] P. Worawitphinyo, X. Gao, and S. Jabeen, "Improving suffix tree clustering with new ranking and similarity measures," vol. 7121 LNAI, ed, 2011, pp. 55-68.

[19] H. Chim and X. Deng, "Efficient phrase-based document similarity for clustering," *IEEE Transactions on Knowledge and Data Engineering,* vol. 20, pp. 1217-1229, 2008.

[20] N. O. Andrews and E. A. Fox, "Recent developments in document clustering," *Department of Computer Science Virginia Tech,* 2007.

[21] O. Zamir and O. Etzioni, "Web document clustering: A feasibility demonstration," *SIGIR Forum (ACM Special Interest Group on Information Retrieval),* pp. 46-54, 1998.

[22] S. M. Zu Eissen, B. Stein, and M. Potthast, "The suffix tree document model revisited," *Proceedings of the 5th International Conference on Knowledge Management,* pp. 596-603, 2005.

[23] K. M. Hammouda and M. S. Kamel, "Efficient phrase-based document indexing for web document clustering," *IEEE Transactions on Knowledge and Data Engineering,* vol. 16, pp. 1279-1296, 2004.

[24] G. Karypis, "CLUTO a clustering tollkit," *Department of Computer Science* vol. Uni. of Minnesota, Tech. Rep 2003.

[25] A. Strehl, J.Ghosh, and R.Mooney, "Impact of similarity measures on web-page clustering," *In Proc. of the 17th National Conf. on Artif. Intell: ,* vol. Workshop of Artif. Intell. for Web Search, pp. 58-64, 2000.

[26] M.Pelillo, "What is a cluster? Perspectives from game theory," *in Proc. of NIPS Workshop on Clutering Theory,* 2009.

[27] P. Fraundorf, "Thermal roots of correlation-based complexity," *Complexity,* vol. 13, pp. 18-26, 2008.

[28] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum, "Hierarchical topic models and the nested Chinese restaurant process," in *Advances in Neural Information Processing Systems,* 2004.

[29] M. Halkidi, B. Nguyen, I. Varlamis, and M. Vazirgiannis, "THESUS: Organizing web document collections based on link semantics," *VLDB Journal*, vol. 12, pp. 320-332, 2003.

[30] X. He, H. Zha, C. H. Q. Ding, and H. D. Simon, "Web document clustering using hyperlink structures," *Computational Statistics and Data Analysis*, vol. 41, pp. 19-45, 2002.

[31] C. Bouveyron and C. Brunet-Saumard, "Model-based clustering of high-dimensional data: A review," *Computational Statistics and Data Analysis*, vol. 71, pp. 52-78, 2014.

[32] H. P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, 2009.

[33] F. Klawonn, "What can fuzzy cluster analysis contribute to clustering of high-dimensional data?," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 8256 LNAI, ed, 2013, pp. 1-14.

[34] I. Kojadinovic, "Agglomerative hierarchical clustering of continuous variables based on mutual information," *Computational Statistics and Data Analysis*, vol. 46, pp. 269-294, 2004.

[35] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis," *John Wiley & Sons*, 1990.

[36] Y. Zhao and G. Karypis, "Hierarchical clustering algorithms for document datasets," *Data Mining and Knowledge Discovery*, vol. 10, pp. 141-168, 2005.

[37] X. Wu, V. Kumar, Q. J. Ross, J. Ghosh, Q. Yang, H. Motoda, *et al.*, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, pp. 1-37, 2008.

[38] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, pp. 143-175, 2001.

[39] R. Kashef and M. S. Kamel, "Enhanced bisecting k-means clustering using intermediate cooperation," *Pattern Recognition*, vol. 42, pp. 2557-2569, 2009.

[40] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means, spectral clustering and normalized cuts," in *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 551-556.

[41] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton, "Competitive learning algorithms for vector quantization," *Neural Networks*, vol. 3, pp. 277-290, 1990.

[42] L. Jing, M. K. Ng, J. Xu, and J. Z. Huang, "Subspace clustering of text documents with feature weighting k-means algorithm," in *Lecture Notes in Computer Science*

(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) vol. 3518 LNAI, ed, 2005, pp. 802-812.

[43] L. King-Ip and R. Kondadadi, "A similarity-based soft clustering algorithm for documents," in *Database Systems for Advanced Applications, 2001. Proceedings. Seventh International Conference on*, 2001, pp. 40-47.

[44] Y. Zhao and G. Karypis, "Soft clustering criterion functions for partitional document clustering: A summary of results," in *International Conference on Information and Knowledge Management, Proceedings*, 2004, pp. 246-247.

[45] K. Yu, S. Yu, and V. Tresp, "Soft clustering on graphs," in *Advances in Neural Information Processing Systems*, 2005, pp. 1553-1560.

[46] S. Mitra, H. Banka, and W. Pedrycz, "Rough-fuzzy collaborative clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, pp. 795-805, 2006.

[47] E. M. Mahdi Shafiei, "Model-based Overlapping Co-Clustering," *Proceedings of the Fourth Workshop on Text Mining, Sixth SIAM International Conference on Data Mining, Bethesda, Maryland, April 20-22*, 2006.

[48] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition - Part II," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, pp. 786-801, 1999.

[49] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition - Part I," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, pp. 778-785, 1999.

[50] S. Rajathi, N. Shajunisha, and S. S. Caroline, "Correlative analysis of soft clustering algorithms," in *Advanced Computing (ICoAC), 2013 Fifth International Conference on*, 2013, pp. 360-365.

[51] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, 2005.

[52] M. E. S. Mendes and L. Sacks, "Evaluating fuzzy clustering for relevance-based information access," in *IEEE International Conference on Fuzzy Systems*, 2003, pp. 648-653.

[53] Jianping Mei and L. Chen, "Fuzzy clustering with weighted medoids for relational data," *Pattern Recognition*, 2010.

[54] J. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms," *Plenum Press New York*, 1981.

[55] W. C. Tjhi and L. Chen, "A heuristic-based fuzzy co-clustering algorithm for categorization of high-dimensional data," *Fuzzy Sets and Systems*, vol. 159, pp. 371-389, 2008.

[56] S. Miyamoto and K. Umayahara, "Fuzzy clustering by quadratic regularization," in *IEEE International Conference on Fuzzy Systems*, 1998, pp. 1394-1399.

[57] R.Krishnapuram; and J.Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.,*, vol. 4, pp. 393-396, 1993.

[58] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 13, pp. 517-530, 2005.

[59] Y. Yan and L. Chen, "Hyperspherical possibilistic fuzzy c-means for high-dimensional data clustering," in *ICICS 2009 - Conference Proceedings of the 7th International Conference on Information, Communications and Signal Processing*, 2009.

[60] C. H. Oh, K. Honda, and H. Ichihashi, "Fuzzy clustering for categorical multivariate data," in *Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS*, 2001, pp. 2154-2159.

[61] M. C. V. Nascimento and A. C. P. L. F. De Carvalho, "Spectral methods for graph clustering - A survey," *European Journal of Operational Research*, vol. 211, pp. 221-231, 2011.

[62] A. Dasgupta, J. Hopcroft, R. Kannan, and P. Mitra, "Spectral clustering by recursive partitioning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 4168 LNCS, ed, 2006, pp. 256-267.

[63] T. Y. Choe and C. I. Park, "A k-way graph partitioning algorithm based on clustering by eigenvector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 3037, ed, 2004, pp. 598-601.

[64] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, pp. 1074-1085, 1992.

[65] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cult algorithm for graph partitioning and data clustering," 2001, pp. 107-114.

[66] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 269-274.

[67] W. Xu, X. Liu, and Y. Gong, "Document Clustering Based On Non-negative Matrix Factorization," *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, pp. 267-273, 2003.

[68] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 126-135.

[69] Z. Li, X. Wu, and H. Peng, "Nonnegative Matrix Factorization on Orthogonal Subspace," *Pattern Recognition Letters*, vol. 31, pp. 905-911, 2010.

[70] Christian Thurau, Kristian Kersting, and M. Wahabzada, "Convex non-negative matrix factorization for massive datasets," *Knowledge and Information Systems*, 2010.

[71] C. Ding, T. Li, and M. Jordan, "Convex and semi-nonnegative matrix factorizations for clustering and low-dimension representation," *Technical Report*, vol. LBNL-60428, 2006.

[72] T. Li and C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," 2007, pp. 362-371.

[73] T. Li, "Non-negative Matrix Factorizations for Clustering: A Survey."

[74] F. Wang, T. Li, and C. Zhang, "Semi-supervised clustering via matrix factorization," in *SIAM international conference on data mining* 2008, pp. 1-12.

[75] Y. Chen, L. Wang, and M. Dong, "Non-Negative matrix factorization for semisupervised heterogeneous data coclustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1459-1474, 2010.

[76] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using vonmises-fisher distributions," *Journal of Machine Learning Research*, vol. 6, pp. 1345-1382, 2005.

[77] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1-38, 1977.

[78] S. Zhong and J. Ghosh, "Generative model-based document clustering: A comparative study," *Knowledge and Information Systems*, vol. 8, pp. 374-384, 2005.

[79] "http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation."

[80] "http://en.wikipedia.org/wiki/Probabilistic_latent_semantic_analysis."

[81] http://en.wikipedia.org/wiki/Bayesian_inference.

[82] Z. Guo, S. Zhu, Y. Chi, Z. Zhang, and Y. Gong, "A latent topic model for linked documents," in *Proceedings - 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009*, 2009, pp. 720-721.

[83] D. Wang, M. Thint, and A. Al-Rubaie, "Semi-supervised latent Dirichlet allocation and its application for document classification," in *Proceedings of the 2012 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, WI-IAT 2012*, 2012, pp. 306-310.

[84] Y. Lu, S. Okada, and K. Nitta, "Semi-supervised latent Dirichlet allocation for multi-label text classification," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 7906 LNAI, ed, 2013, pp. 351-360.

[85] K. Kummamuru, A. Dhawale, and R. Krishnapuram, "Fuzzy co-clustering of documents and keywords," in *IEEE International Conference on Fuzzy Systems*, 2003, pp. 772-777.

[86] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," *Proc. 9th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD 03)*, pp. 89-98, 2003.

[87] W. C. Tjhi and L. Chen, "Possibilistic fuzzy co-clustering of large document collections," *Pattern Recognition*, vol. 40, pp. 3452-3466, 2007.

[88] Q. Gu and J. Zhou, "Co-Clustering on Manifolds," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2009.*, 2009.

[89] J. Li and T. Li, "HCC: A hierarchical co-clustering algorithm," in *SIGIR 2010 Proceedings - 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2010, pp. 861-862.

[90] Y. Zhao, J. Y. Xu, G. Wang, L. Chen, B. Wang, and G. Yu, "Maximal subspace co-regulated gene clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, 2008.

[91] F. Shang, L. C. Jiao, and F. Wang, "Graph dual regularization non-negative matrix factorization for co-clustering," *Pattern Recognition*, vol. 45, pp. 2237-2250, 2012.

[92] M. M. Shafiei and E. E. Muios, "Latent dirichlet co-clustering," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2006, pp. 542-551.

[93] Q. Fu and A. Banerjee, "Bayesian overlapping subspace clustering," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2009, pp. 776-781.

[94] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, pp. 24-45, 2004.

[95] M. Charrad and M. Ben Ahmed, "Simultaneous clustering: A survey," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 6744 LNCS, ed, 2011, pp. 370-375.

[96] R. Bekkerman, R. El-Yaniv, Y. Winter, and N. Tishby, "On feature distributional clustering for text categorization," *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, pp. 146-153, 2001.

[97] S. Vempala and G. Wang, "The benefit of spectral projection for document clustering," *Proceedings of the 3rd Annual Workshop on Clustering High Dimensional Data and its Applications, SIAM International Conference on Data Mining*, 2005.

[98] B. Gao, T. Y. Liu, X. Zheng, Q. S. Cheng, and W. Y. Ma, "Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering," 2005, pp. 41-50.

[99] D. Greene and P. Cunningham, "Spectral co-clustering for dynamic bipartite graphs," in *CEUR Workshop Proceedings*, 2010, pp. 29-40.

[100] L. A. F. Park, C. A. Leckie, K. Ramamohanarao, and J. C. Bezdek, "Adapting spectral co-clustering to documents and terms using Latent Semantic Analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 5866 LNAI, ed, 2009, pp. 301-311.

[101] H. Frigui and O. Nasraoui, "Simultaneous categorization of text documents and identification of cluster-dependent keywords," in *IEEE International Conference on Fuzzy Systems*, 2002, pp. 1108-1113.

[102] Y. Chen, M. Rege, M. Dong, and J. Hua, "Non-negative matrix factorization for semi-supervised data clustering," *Knowledge and Information Systems*, vol. 17, pp. 355-379, 2008.

[103] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 59-68, 2004.

[104] N. Grira, M. Crucianu, and N. Boujemaa, "Active semi-supervised fuzzy clustering," *Pattern Recognition*, vol. 41, pp. 1851-1861, 2008.

[105] K. Li, Z. Cao, L. Cao, and R. Zhao, "A novel semi-supervised fuzzy c-means clustering method," in *2009 Chinese Control and Decision Conference, CCDC 2009*, 2009, pp. 3761-3765.

[106] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 577-584, 2001.

[107] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning Distance Functions using Equivalence Relations," *Proceedings, Twentieth International Conference on Machine Learning*, vol. 1, pp. 11-18, 2003.

[108] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," *Advances in Neural Information Processing Systems*, vol. 15, pp. 505-512, 2003.

[109] H. Cevikalp and R. Paredes, "Semi-supervised distance metric learning for visual object classification," in *VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications*, 2009, pp. 315-322.

[110] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, pp. 81-88, 2004.

[111] M. Soleymani Baghshah and S. Bagheri Shouraki, "Kernel-based metric learning for semi-supervised clustering," *Neurocomputing*.

[112] I. W. Tsang, P. M. Cheung, and J. T. Kwok, "Kernel relevant component analysis for distance metric learning," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2, pp. 954-959, 2005.

[113] B. Yan and C. Domeniconi, "An adaptive kernel method for semi-supervised clustering," *17th European Conference on Machine Learning*, pp. 18-22, 2006.

[114] D. Y. Yeung and H. Chang, "A kernel approach for semisupervised metric learning," *IEEE Transactions on Neural Networks*, vol. 18, pp. 141-149, 2007.

[115] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: A kernel approach," in *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 457-464.

[116] K. Li, C. Zhang, and C. Zheng, "Semi-supervised kernel clustering algorithm based on seed set," in *Proceedings - 2009 Asia-Pacific Conference on Information Processing, APCIP 2009*, 2009, pp. 169-172.

[117] X. Yin, S. Chen, E. Hu, and D. Zhang, "Semi-supervised clustering with metric learning: An adaptive kernel method," *Pattern Recognition*, vol. 43, pp. 1320-1333, 2010.

[118] S. Basu, A. Banerjee, and R. Mooney, "Semi-supervised clustering by seeding," *Proceedings of the 19th International Conference on Machine Learning*, pp. 19-26, 2002.

[119] Jianying Hu, Moninder Singh, and A. Mojsilovic, "Categorization Using Semi-Supervised Clustering," *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on Pattern Recognition*, pp. 1-4, 2009.

[120] Jing Gao, Pang-Ning Tan, and H. Cheng, "Semi-supervised clustering with partial background information," *SDM'06: proceedings of the sixth SIAM international conference on data mining. SIAM, Bethesda, MD, USA*, 2006.

[121] M. Charikar, V. Guruswami, and A. Wirth, "Clustering with qualitative information," in *Annual Symposium on Foundations of Computer Science - Proceedings*, 2003, pp. 524-533.

[122] Germain Forestier, C ́edric Wemmert , and P. Ganc ̧arski, "Semi-supervised collaborative clustering with partial background knowledge," *IEEE International Conference on Data Mining Workshops*, 2008.

[123] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Annual ACM Conference on Computational Learning Theory*, 1998, pp. 92-100.

[124] ENDO Yasunori, HAMASUNA Yukihiro, YAMASHIRO Makito, and M. Sadaaki, "On Semi-Supervised Fuzzy c-Means Clustering," *FUZZ-IEEE 2009, Korea*, 2009.

[125] N. Grira, M. Crucianu, and N. Boujemaa, "Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration," *IEEE International Conference on Fuzzy Systems (Fuzz'IEEE 2005)*, 2005.

[126] Y. Song, S. Pan, and S. Liu, "Constrainted Co-clustering for Textual Documents," *Association for the Advancement of Artificial intelligence*, 2010.

[127] V. Sindhwani, J. Hu, and A. Mojsilovic, "Regularized Co-Clustering with Dual Supervision," *In Proceedings of NIPS*, 2008.

[128] X. Shi, W. Fan, and P. S. Yu, "Efficient semi-supervised spectral co-clustering with constraints," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2010, pp. 1043-1048.

[129] S. Zhong, "Semi-supervised model-based document clustering: A comparative study," *Interational Journal of Machine Learning*, 2006.

[130] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *IEEE Signal Processing Letters*, vol. 17, pp. 4-7, 2010.

[131] Tao Li, Yi Zhang, and V. Sindhwani, "A non-negative Matrix Tri-factorization Approach to Sentiment Classification with Lexical Prior Knowledge," *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, 2009.

[132] Witold Pedrycz, Vincenzo Loia, and S. Senatore, "P-FCM: a proximity-based fuzzy clustering," *Fuzzy Sets and Systems*, vol. 148, pp. 21-41, 2004.

[133] Witold Pedrycz, Vincenzo Loia, and S. Senatore, "Fuzzy Clustering with Viewpoints," 2009.

[134] Huifang Ma, Weizhong Zhao, Qian Tan, and Z. Shi, "Orthogonal Nonegative Matrix Tri-factorization for Semi-supervised Document Co-clustering," *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 189-200, 2010.

[135] F. Huang, Y. Yang, T. Li, J. Zhang, T. Rutayisire, and A. Mahmood, "Semi-supervised hierarchical co-clustering," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 7414 LNAI, ed, 2012, pp. 310-319.

[136] Y. Song, S. Pan, S. Liu, F. Wei, M. X. Zhou, and W. Qian, "Constrained text coclustering with supervised and unsupervised constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, pp. 1227-1239, 2013.

[137] T. Li, C. Ding, Y. Zhang, and B. Shao, "Knowledge transformation from word space to document space," 2008, pp. 187-194.

[138] V. Sindhwani, "Document-word co-regularization for semi-supervised sentiment analysis," *In Proceedings of IEEE ICDM*, 2008.

[139] M. R. Ackermann, J. Blömer, and C. Sohler, "Clustering for metric and nonmetric distance measures," *ACM Transactions on Algorithms*, vol. 6, 2010.

[140] D. Klein, S. D. Kamvar, and C. D. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering," *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 307-314, 2002.

[141] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 39-48, 2003.

[142] L. Wu, S. C. H. Hoi, R. Jin, J. Zhu, and N. Yu, "Learning Bregman distance functions for semi-supervised clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, pp. 478-491, 2012.

[143] Part, Y. Chen, L. Wang, and M. Dong, "Semi-supervised document clustering with simultaneous text representation and categorization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 5781 LNAI, ed, 2009, pp. 211-226.

[144] T. Joachims, "Transductive Learning via Spectral Graph Partitioning," in *Proceedings, Twentieth International Conference on Machine Learning*, 2003, pp. 290-297.

[145] D. Zhou and C. J. C. Burges, "Spectral clustering and transductive learning with multiple views," in *ACM International Conference Proceeding Series*, 2007, pp. 1159-1166.

[146] A. Erdem and M. Pelillo, "Graph transduction as a noncooperative game," *Neural Computation*, vol. 24, pp. 700-723, 2012.

[147] X. Z. Jaydeep De, Li Cheng, "transduction on Directed Graphs via Absorbing Random Walks," *arXia preprint arXiv:*, vol. 1402.4566, 2014.

[148] W. Liu, J. Wang, and S. F. Chang, "Robust and scalable graph-based semisupervised learning," *Proceedings of the IEEE*, vol. 100, pp. 2624-2638, 2012.

[149] T. Iwata and K. Duh, "Bidirectional semi-supervised learning with graphs," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial*

Intelligence and Lecture Notes in Bioinformatics) vol. 7524 LNAI, ed, 2012, pp. 293-306.

[150] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, 2004.

[151] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," in *Proceedings, Twentieth International Conference on Machine Learning*, 2003, pp. 912-919.

[152] J. Wang, T. Jebara, and S. F. Chang, "Graph transduction via alternating minimization," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 1144-1151.

[153] J. Wang, Y. G. Jiang, and S. F. Chang, "Label diagnosis through self tuning for web image search," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009, pp. 1390-1397.

[154] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, pp. 316-323, 1999.

[155] R. Baeza-Yates and B. Ribeiro-Neto, "Modem information retrieval," *ACM Press/Addison-Wesley*, 1999.

[156] B. Larsen and C. Aone, "Fast and e®ective text mining using linear-time document clustering," *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999.

[157] Y. Yan and L. Chen, "Label-based semi-supervised fuzzy co-clustering for document categoraization," in *ICICS 2011 - 8th International Conference on Information, Communications and Signal Processing*, 2011.

[158] Y. Yan, L. Chen, and W. C. Tjhi, "Semi-supervised fuzzy co-clustering algorithm for document categorization," *Knowledge and Information Systems*, vol. 34, pp. 55-74, 2013.

[159] E. H. Ruspini, "A new approach to clustering," *Information and Control*, vol. 15, pp. 22-32, 1969.

[160] Y. Yan, L. Chen, and W. C. Tjhi, "Fuzzy semi-supervised co-clustering for text documents," *Fuzzy Sets and Systems*, vol. 215, pp. 74-89, 2013.

[161] "http://qwone.com/~jason/20Newsgroups/."

[162] "http://www.daviddlewis.com/resources/testcollections/reuters21578/."

[163] "http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/."

[164] L. Jing, M. K. Ng, and J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 1026-1041, 2007.

[165] M.P.Sinka and D.W.Corne, "a large benchmark dataset for web document clustering, soft computing systems " *design, management and application Vol.87 of frontiers in artificial intelligence and applications.*, pp. 881-890, 2002.

[166] A. Hotho, S. Staab, and G. Stumme, "Text clustering based on background knowledge," *Technical Report*, p. 36, 2003.

[167] X. Ji, W. Xu, and S. Zhu, "Document clustering with prior knowledge," in *Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 405-412.

[168] C. F. Gao, X. J. Wu, and S. S. Zhang, "An improved semi-supervised fuzzy clustering algorithm," *Kongzhi yu Juece/Control and Decision*, vol. 25, pp. 115-120.

[169] Sicheng Xiong, Javad Azimi, and x. Z.Fern, "Active Learning of Constraints for Semi-Supervised Clustering," *Knowledge and Data Engineering, IEEE Transactions on* vol. PP, 2013.

[170] Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz, "Sensor network localization, Euclidean distance matrix completions, and graph realization," *Optimization and Engineering*, vol. 11, pp. 45-66, 2010.

[171] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: From transductive to semi-supervised learning," in *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 825-832.

[172] Y. Yan, L. Chen, and D. T. Nguyen, "Semi-supervised clustering with multi-viewpoint based similarity measure," 2012.

[173] T. M. Mitchell, "Machine Learning. ," *McGraw-Hill*, 1997.

[174] "http://en.wikipedia.org/wiki/Sentiment_analysis."

[175] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, pp. 1-135, 2008.

[176] "http://www.cs.cornell.edu/People/pabo/movie-review-data/."

[177] C. Ding, H. D. Simon, R. Jin, and T. Li, "A learning framework using Green's function and kernel regularization with application to recommender system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 260-269.

[178] V. S. a. S. Keerthi, "Large scale semi-supervised linear svms," *In SIGIR, 2006*, 2006.

[179] T. Joachims, "Transductive inference for text classification using support vector machines," *International Conference on Machine Learning*, 1999.

[180] A. Strehl and J. Ghosh, "Cluster ensembles - A knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583-617, 2003.

[181] G. Peters and R. Weber, "Dynamic clustering with soft computing," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 2, pp. 226-236, 2012.

[182] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. L. F. De Carvalho, and J. Gama, "Data stream clustering: A survey," *ACM Computing Surveys,* vol. 46, 2013.