Relation and fuzzy clustering for document categorization and analysis

Mei, Jian-Ping

2012

Mei, J.-P. (2012). Relation and fuzzy clustering for document categorization and analysis.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/48627>

<https://doi.org/10.32657/10356/48627>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**RELATION AND FUZZY CLUSTERING FOR DOCUMENT
CATEGORIZATION AND ANALYSIS**

MEI JIAN-PING

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

2012

RELATION AND FUZZY CLUSTERING FOR DOCUMENT CATEGORIZATION AND ANALYSIS

MEI JIAN-PING

MEI JIAN-PING

School of Electrical and Electronic Engineering

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

2012

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 my classmate Nguyen Duc Thang for their friendship and help in my research.

I also would like to thank Linh, Caoxiao, Xinyi and all the other FYP students who have helped me in the project.

Special thanks also to all the very helpful technicians: Ms How and Christina in Software Engineering Lab; Audrey in Media Technology Lab.

Last and most importantly, I wish to thank my family: my husband who stays with me through the joys and pains; and my parents, my grandparents, and my sister for their constant love and encouragement.

Contents

Acknowledgments	i
Summary	ii
List of Figures	viii
List of Tables	xi
1 Introduction	1
1.1 Overview	1
1.2 Motivations	3
1.3 Objectives	4
1.4 Contributions	5
1.5 Structure of the Thesis	6
2 Background	7
2.1 Overview	7
2.2 Document Representation and Similarity	8
2.2.1 Document Representation	9
2.2.2 Document Similarity	12
2.3 Data Formats	14
2.3.1 Vector-based Data and Proximity Data	14
2.3.2 Multi-type Relational Data	17
2.4 Existing Document Clustering Approaches	20
2.4.1 Hierarchical Clustering	21
2.4.2 Partitioning Clustering	22
2.4.3 Fuzzy Clustering	25
2.4.4 Probabilistic Model-based Methods	28

2.4.5	Graph Partitioning Methods	29
2.4.6	Nonnegative Matrix Factorization Methods	31
2.4.7	Co-clustering and Multi-type Relational Clustering	32
2.4.8	Semi-supervised Clustering	35
2.4.9	Discussion	36
2.5	Why Treat Document Data as Multi-type Relational Data? Why Fuzzy Clustering?	37
2.6	Evaluation	39
2.7	Summary	41
3	k-Partitions Clustering with Weighted Medoids for Relational Data	43
3.1	Overview	43
3.2	Existing Medoid-based Clustering Approaches	44
3.2.1	PAM: A k -Medoids Approach	44
3.2.2	FCMdd: A Fuzzy Relative of k -Medoids	45
3.2.3	Limitation of the One-Medoid-One-Cluster Strategy	46
3.3	The Proposed Fk-Parts for Relational Data Clustering	47
3.3.1	Formulation	47
3.3.2	Solution	51
3.3.3	Algorithm	56
3.4	Experimental Results	59
3.4.1	Weighted Medoids vs. Single Medoid	59
3.4.2	Fuzzy Membership vs. Prototype Weight	61
3.4.3	Country Categorization	62
3.4.4	Document Categorization	66
3.5	Summary	70
4	Relation Integrated Fuzzy c-Means	73
4.1	Overview	73
4.2	Existing Fuzzy Clustering Approaches	74
4.2.1	Hyperspherical fuzzy c-means (HFCM)	74
4.2.2	Fuzzy Relational Clustering	75

4.2.3	Constrained Fuzzy Clustering	76
4.3	Motivation	77
4.4	Proposed LinkFCM	78
4.4.1	Formulation	78
4.4.2	Solution	80
4.4.3	Choice of β	83
4.5	Constrained LinkFCM	84
4.6	Experiments	87
4.6.1	Artificial Data	87
4.6.2	Cora Data	90
4.7	Summary	99
5	Fuzzy Approach for Multi-type Relational Data Clustering	101
5.1	Overview	101
5.2	Motivation	103
5.3	Fuzzy Clustering for Multi-type relational Data	104
5.3.1	Problem Formulation	104
5.3.2	Solution	107
5.3.3	Algorithm	109
5.3.4	Parameter Setting	111
5.4	Star-Structure and Extended Star-Structure	112
5.4.1	FC-MR for Star-Structured Relational Data	112
5.4.2	FC-MR for Extended Star-Structured Relational Data	114
5.4.3	FC-MR for Homogeneous Relational Data	115
5.5	Related Work	115
5.6	Experimental Results	117
5.6.1	A Running Example	118
5.6.2	Experiments on Star-Structured Relational data	119
5.6.3	Experiment on Extended Star-Structured data	127
5.7	Summary	131

6	Application: Subtopic-based Multi-document Summarization	133
6.1	Overview	133
6.2	Background and Motivations	134
6.2.1	Background	134
6.2.2	Why Fk -Parts?	136
6.3	The Proposed Method: SumCR	137
6.3.1	System Overview	137
6.3.2	Feature, Score and Selection	138
6.4	Experimental Results	142
6.4.1	Systems Evaluated	142
6.4.2	Data Sets and Preprocessing	143
6.4.3	Evaluation Metric	144
6.4.4	Comparison of Results	144
6.4.5	Results with Different w_p	148
6.5	Summary	148
7	Conclusion and Future Work	151
7.1	Conclusion	151
7.2	Future Work	155
	Publication	158
	References	160

Summary

Data mining is a very important stage in the knowledge discovery process for automatic analysis of a large amount of data. Among various data mining applications with machine learning approaches, this thesis focuses on the investigations of using fuzzy clustering for automatic document categorization. Traditional document clustering approaches deal with pure text documents. However, data generated nowadays become more and more diverse in many applications. Other than text or content, other types of data may also be available for describing the document dataset from different angles and can be used for clustering the documents. Compared with the way of performing data transformation to combine or merge different types of data into one basic form and then applying a conventional clustering approach to produce the document clusters, directly clustering with multiple types of data has several merits. It keeps the loss of information to the least, is able to discover interrelated patterns, and produces more informative results.

Based on the solid foundation of objective-function based fuzzy clustering of vector-based data, we consider to represent document data as relational data, and propose models for the corresponding relational data clustering. Three approaches are proposed to handle the document clustering problem under different scenarios. We start with a basic situation where documents are clustered based on document-document relation and propose a k -partitions clustering approach with weighted medoids called Fk -Parts. This approach is a generalized k -medoids approach, where each cluster is represented by multiple representative objects with different weights. This new mechanism enhances the ability of Fk -Parts to describe clusters of various structures. After that, we consider situations where both the vector-based data encoding the text content of each document and the document-document relation are available. To

make use of both types of data, we propose the LinkFCM approach by incorporating relational data into the well known fuzzy c-means formulation. Based on the new objective function of LinkFCM, it clusters documents by using the content information of each individual document together with the relationships between documents. Finally we extend our study to handle more general situations where documents are clustered based on the relations between documents and multiple other types of objects. A fuzzy clustering FC-MR is proposed to simultaneously cluster documents and other related types of objects. When only the document-document relation is available, FC-MR is equivalent to *Fk*-Parts. Given document-word relation and document-document relation, LinkFCM treats each document as a vector of words, and only produces document clusters; while FC-MR treats *document* and *word* as two object types and produces document clusters as well as word clusters.

The performance of the three proposed approaches are evaluated using benchmark document datasets extracted from three real-world document collections: CLUTO, 20Newsgroups and Cora. Our experimental results show that compared with using single medoid, using multiple weighted medoids provides a more powerful way to characterize the underlying clusters, and this enables *Fk*-Parts to achieve better clustering results than existing medoid-based approaches. Compared with existing approaches using transformed data, it is usually more effective to make use of different types of data directly during clustering. Our experimental results also show that the proposed FC-MR algorithm produces clustering results with higher accuracy in less execution time than matrix factorization-based and spectral-based multi-type relational data clustering approaches.

Other than document categorization, we also investigate one of the important applications of the proposed *Fk*-Parts clustering approach for extractive-based text summarization. A novel subtopic-level feature is defined based on the sentence clusters generated with *Fk*-Parts. With this new feature, the summary produced tends to have a high relevance and broad coverage in content. Experimental results on benchmark DUC datasets show the good performance and great potential of our clustering-based summarization system SumCR.

List of Figures

2.1	The “a bag of words” model.	10
2.2	Vector representation of a document.	11
2.3	Two basic data formats: vector-based data and relational data	15
2.4	Graph representation of homogeneous relational data. Partitioning of a graph gives the clustering result of the dataset.	16
2.5	Bi-partite graph representation of bi-type relational data. Partitioning of a bi-partite graph gives the clustering result of the corresponding bi- type relational data. Different object-types are denoted with different shapes.	18
2.6	Structures of relational data: (a) a classical graph contains a relation between objects of the same type, and a bi-partite graph contains a relation between two types of objects; (b) A r -partite graph contains relations between objects of r different types.	18
2.7	General structure of r -type relational data with relations between dif- ferent types and relations between the same types.	19
2.8	Relational data with two special structures: star-structure and extended- star structure.	19
2.9	Document dataset with relations among multiple types of objects. Dif- ferent object types are denoted with different shapes.	21
3.2	Scatter plot of the X_{11} data. The red circle “o” represents the ideal center of a cluster.	60

3.3	Difference between (u) and prototype weight (v): (a) the scatter plot of the X_{600} data; (b) the contour plot of u values produced with FCMdd; (c) the contour plot of v values produced with Fk-Parts. Only u and v values of objects with respect to the cluster they belong to are plotted. The dot “•” represents the ideal center of a cluster.	63
3.4	Clustering results of four (dis)similarity-based partitioning approaches for document datasets in Table 3.5.	69
4.1	Illustration of a document dataset for which two matrices are extracted: the paper-word co-occurrence matrix and the paper-paper relation matrix. The relation between two papers may be defined in various ways to describe specific relationships between two papers, e.g., two papers are said to be related if one of them is cited by the other, or the paper-paper relation defines whether two papers are written by the same author. . .	78
4.2	A 2-D dataset with two semi-circle shaped clusters. (a) Distribution of the dataset; (b) The clustering results with FCMm.	88
4.3	Clustering result of LinkFCMm using data-based relational information for the 2-D dataset. (a) Links between each object and their 3-nearest neighbors; (b) clustering result of LinkFCMm.	89
4.4	Clustering results of LinkFCMm with user-provided relation for the 2-D dataset. (a), (c) and (e) plots the pairwise constraints with $y = \{20, 80, 200\}$, respectively, where black-solid lines indicate must-links and red-dash lines indicate cannot-links; (b), (d) and (f) show the corresponding clustering results of LinkFCMm.	91
4.5	A sample record of extracted information of a paper in the Cora data. Each record usually consists of three parts: the header part, which includes URL, Title, Author(s) and some other related information like Address, Affiliation and Email; the content of abstract; a list of formatted references.	92
4.6	Clustering accuracy of LinkFCMm and LinkFCMq with different numbers of pairwise constraints.	98

5.1	A toy problem of tri-type relational data clustering. The data consists of 30 objects, where $ \mathcal{X}_1 = 10$, $ \mathcal{X}_2 = 12$, and $ \mathcal{X}_3 = 8$ and two relations: $\mathcal{X}_1 \sim \mathcal{X}_2$ and $\mathcal{X}_1 \sim \mathcal{X}_3$. The corresponding relation matrices $\mathbf{R}_{12}$, $\mathbf{R}_{13}$ are plotted. Each entry in a relation matrix is plotted as a dot, and a larger dot indicates a stronger relationship.	118
5.2	Fuzzy memberships of documents with respect to C_1 of TM1 by HFCM and FC-MR. The (horizontal) x-axis denotes the document id ($i = 1, \dots, 500$) and the (vertical) y-axis shows the value of the membership u_{i1} . In (a), small circles label objects that are clustered incorrectly. . .	123
5.3	Comparison of time efficiency of the four approaches: iTopicModel, LinkFCM, NMF-MR and FC-MR. The time required by each of the approaches on eight datasets are displayed in log(seconds).	131
6.1	An overview of the proposed SumCR summarizer.	139
7.1	Comparison of input and output of the three proposed approaches: Fk -Parts, LinkFCM and FC-MR.	154
7.2	Connections between the proposed approaches and existing (fuzzy) clustering approaches.	155

List of Tables

3.1	Results of FCMdd and Fk-Parts for the X_{11} dataset	60
3.2	Countries Data [1]: Dissimilarity Matrix ($d_{ij} \in D$)	65
3.3	Results of FCMdd for the Countries Data	65
3.4	Results of Fk-Parts and PAM for the Countries Data	66
3.5	Summary of eight document datasets (n : # of documents, s : # of words, k : # of classes, $\bar{n}_c$: average number of documents per class, and <i>Balance</i> : the size ratio of the smallest class to the largest one)	67
3.6	Performance comparison of the six relational clustering approaches for the eight datasets given in Table 3.5	71
3.7	Significance test of improvement of Fk-Parts compared with SNMF and UPGMA	71
4.1	Summary of Cora Datasets. n : # of papers, s : # of words, k : # of subfields, $\bar{l}$: average number of citations per paper.	93
4.2	Performance comparison of fuzzy approaches using Cora data	95
4.3	Significance test of improvement of LinkFCM compared to HFCM and Fk-parts. The results of HFCM and Fk-Parts are based on word+link	96
4.4	Comparison of <i>Accuracy</i> of semi-supervised learning for Cora data	97
5.1	Updating of rankings and memberships in FC-MR in iteration	120
5.2	Topic structures of the three 20Newsgroup datasets: TM1, TM2 and TM3	121
5.3	Contingency table of three fuzzy approaches for TM1	123
5.4	Contingency table of three fuzzy approaches for TM2	123
5.5	Contingency table of three fuzzy approaches for TM3	123

5.6	Performance comparison of three multi-type relational data clustering approaches for 20Newsgroups data	125
5.7	Top 10 words in each cluster of the three newsgroup datasets generated by FC-MR	125
5.8	Comparison of <i>Accuracy</i> of four approaches iTopicMode, LinkFCM, NMF-MR and FC-MR, using two groups of document datasets	130
5.9	Comparison of <i>NMI</i> of four approaches iTopicMode, LinkFCM, NMF-MR and FC-MR, using two groups of document datasets	130
6.1	Summarization systems	143
6.2	Summary of the summarization tasks and datasets	143
6.3	Comparison of SumCR, MEAD and SNMF	145
6.4	Improvement of SumCR with respect to MEAD and SNMF	145
6.5	More comparisons of SumCR with state-of-the-art approaches	146
6.6	Results of SumCR-G on DUC2004 with $w_e = 1$ and various w_p	149
6.7	Results of SumCR-Q on DUC2005 with $w_e = w_s = 1$ and various w_p	149
6.8	Results of SumCR-Q on DUC2006 with $w_e = w_s = 1$ and various w_p	149

Chapter 1

Introduction

1.1 Overview

With the development of computer and Internet technologies, a huge amount of data has been generated every day at an ever increasing speed by organizations and individual users of the World Wide Web (Web). Data Mining emerges to be an important research topic due to the practical need to solve the information overload problem. Clustering has been used as a fundamental yet important data mining tool in many real-world data analysis tasks for exploring underlying structures of a dataset. For a given dataset, a clustering approach groups objects in the dataset into subgroups called clusters so that objects in the same cluster are expected to be more similar than those in different clusters. This cluster-represented data provides a simpler description of the original dataset but without loss of much information. Clustering has been an active subject in several research communities such as statistics, pattern recognition and machine learning. More recently, clustering plays an important role in data mining applications, information retrieval, bioinformatics and many others. From the machine learning point of view, clustering is an unsupervised learning method as it processes based on the nature properties of the data without additional supervision. This makes clustering different from classification, which labels each of the objects as one or more pre-defined categories. A clustering approach always implicitly or explicitly involves a similarity measure and a clustering criterion, which directly contribute to the effectiveness of the clustering approach.

Due to the existence of large amounts of document data, document clustering arises to be a need in applications that rely on document analysis, such as Web mining and information retrieval. According to the official Google blog in July, 2008, the Web already contains one trillion unique URLs and the number of individual Web pages is growing by several billion pages per day. Other than online data, organization owned databases also contain large amount of document data. Document clustering or automatic document categorization may be applied as a pre-step to facilitate further mining procedures, e.g. information retrieval by focusing on a subset rather than the whole data collection [2], [3]. It is also used for effective representation, i.e., providing data in an organized structure so that the user can browse the content of the relevant information more efficiently, e.g., grouping the results returned by a search engine in response to a query [4, 5].

Up till now, various clustering approaches have been proposed for solving different problems. A clustering algorithm is usually formulated by taking into consideration the requirements of the particular task and the nature of the data to be handled. Current document clustering approaches mainly cluster documents based on the text data contained in each of the documents. (Text) document clustering itself is a well-established research topic in text mining, which is a research area around applying data mining approaches to text possibly together with techniques from other fields such as database, information retrieval, machine learning, and natural language processing (NLP). Before proceeding to the actual clustering process, representing the given document dataset into a proper mathematical form which can be handled by a clustering algorithm is always the problem that needs to be addressed first. A classic yet still widely used document representation model is the vector space model (VSM) or “a bag of words” model. In this model, each document is represented as a vector with each valid term (word) as a dimension. A large number of existing document clustering approaches are designed based on this model for document representation, and cosine measure is used to calculate the similarity between two documents. Despite the simplicity, directly clustering documents with the VSM representation is difficult since the vector space formed is sparse and high-dimensional. Moreover, this type of representation only preserves statistical information of the original document dataset while other useful information is lost.

Techniques from the NLP community provide more sophisticated text representation models and text similarity measures that may be used to improve the document clustering result. In some cases, the text contained in each document is short, and thus provides insufficient information for the clustering purpose. Nevertheless, other kinds of data other than text may exist. To improve the document clustering results, new clustering methods should be developed to effectively make use of multiple types of data together in document clustering.

1.2 Motivations

Despite extensive studies on document clustering, to effectively cluster and handle various kinds of document data still remains as a very challenging task. Some of the main challenges to develop an effective clustering engine for documents are discussed as follows:

- Diversity of the data. Other than the text content of each document, other useful information may be available. A typical example is Web document, where other than the text, links among documents on the Web provide very useful information on the structure of the Web document data. Another example is the research paper data. For each paper recorded in a database, together with the content of each paper, the reference list, the author, and the journal or conference the paper is published are usually also recorded. A fundamental while critical concept in clustering is the measure of similarity between two documents. It may be insufficient to conclude that two documents are not related to the same topic simply because they contain no word in common. The association of these two documents may be described by other types of data, e.g., the links to the same Web page, or the two are reported by the same author, or the two are published to solve the same problem. Since different types of data possibly describe the document dataset from different perspectives, and some times information contained in the text data may not be sufficient enough for the purpose of clustering, it is very important to be able to effectively make use of multiple types of data together for document clustering.

- **Overlap.** In many clustering algorithms, such as k-means and hierarchical clustering, one document is only assigned to one cluster. However, in real cases, one document might span multiple topics; in other words, more than one topic could be described in one document. For this reason, it is more natural for a clustering algorithm to allow each document to belong to multiple topics with various degrees so that overlap between clusters is able to be captured.
- **Interpretability of clustering results.** As an automated process, the output of the clustering system should be able to be understood by human experts or subsequent systems for further processing, validation and evaluation. Clusters of a simple 2-D or 3-D dataset can usually be interpreted by human beings. However, when the complexity of the dataset increases, clustering results become more difficult to be interpreted, even for human experts. It is helpful if descriptive information on each of the produced document clusters can be obtained together with the document clusters at the end of the clustering process.
- **Scalability.** On the one hand, a clustering algorithm with good effectiveness usually requires high computational cost; on the other hand, we are facing an ever-increasing huge volume of data. Therefore for clustering to be practically useful, clustering algorithms need to maintain low computational complexities or be scalable to large datasets.

1.3 Objectives

The goal of this study is to improve clustering analysis of documents by finding a suitable mathematical representation of the document data, and developing effective while scalable clustering algorithms based on the corresponding data representations. We treat the document data as relational data, which may involve multiple relations including pairwise relation of documents and relations between document and other object types. Each relation typically encodes one type of data or one view of the dataset. This relation-based model provides a more flexible and effective way than the traditional vector-based model for representing multiple groups of data which describe the document dataset from different perspectives. Based on the solid foundation

of objective-function based fuzzy clustering of vector-based data and the alternating optimization algorithm, our proposed fuzzy models for relational data clustering aim to achieve improvement in clustering accuracy while maintain simplicity in time complexity.

1.4 Contributions

In the process of achieving our objective, we have made several contributions to the field of clustering as follows:

- Provide a comprehensive review of the field of document clustering by discussing important properties of major representative approaches of clustering and their recent developments.
- Propose a new k -partitions clustering approach Fk -Parts which characterizes each cluster using multiple weighted medoids and works on pairwise document relation.
- Propose a new fuzzy clustering approach LinkFCM to cluster documents in a dataset based on both the vector representation of each document and the relationships between pairs of documents.
- Formulate a fuzzy clustering approach for multi-type relational data called FC-MR. This approach is a ranking-based multi-way clustering, which simultaneously clusters documents and other related types of objects.
- Apply the Fk -Parts clustering approach to sentence-extractive based text summarization. Based on the sentence clustering result of Fk -Parts, a new subtopic-level feature is defined to help to extract a summary for a group of documents with its content has high relevance and a broad coverage.

1.5 Structure of the Thesis

In the next chapter, we provide a literature review of the field of clustering, including data representation formats, representative clustering approaches and evaluation metrics. In Chapter 3 to Chapter 5, we present the detailed formulations and experimental studies of the three proposed approaches namely *Fk*-Parts, LinkFCM and FC-MR, respectively. In Chapter 6, we provides an application of *Fk*-Parts in automatic multi-document summarization. The conclusion of this thesis and some future research directions are given in Chapter 7.

Chapter 2

Background

2.1 Overview

Due to the importance of clustering in various data analysis tasks, such as document categorization in text mining [6–8], gene expression data analysis in bioinformatics [9, 10], market basket analysis [11, 12], spatial database applications [13, 14] and Web applications [15–17], considerable efforts have been made on development of clustering algorithms by researchers from multiple disciplines. Common open issues in clustering including the number of clusters, robustness, and scalability, together with additional challenges in clustering of some specific types of data, such as the problem of high-dimensionality for text data, gene expression data and transaction data, and the difficulty of capturing arbitrary shaped clusters for spacial data, have been extensively investigated in the literature. 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 document analysis.

Specifically, the review of background knowledge of document clustering in this chapter contains the following four components:

- Document representation and similarity. Documents are encoded into a model attempting to retain information as much as possible under a certain computation limitation. The clustering and similarity calculation are then performed based on the document representation model. Similarity measure is a building block in clustering which alone can improve the final performance even with the

same clustering algorithm. A good similarity measure captures discriminative information between documents that helps the clustering algorithms find underlying structures while a poor similarity measure conveys little useful information and may mislead the clustering process.

- **Data formats.** Before any clustering algorithm can be applied, the document data needs to be represented into a mathematical form based on which clustering is performed. Most of the clustering approaches cluster documents based on vector-represented data (or object data), or pairwise proximity data (or relational data). For a document dataset which is described by several types of data, it is more naturally to be treated as multi-type relational data.
- **Clustering algorithms.** This is the core technique in data clustering. Typically, the previous data preprocessing and representation are eventually used to serve the clustering algorithm. We will review clustering approaches in categories. Clustering approaches in each category are formulated under the same or similar mathematical framework. We compare different clustering methods through discussion the strong and weak points of each type of clustering methods. Based on the review study of these existing clustering approaches, we then discuss the need of the proposed approaches for document clustering.
- **Evaluation.** This is a necessary component in the research of clustering since we need to validate the effectiveness of an algorithm and compare the performance of different algorithms in data clustering. We review several widely used external evaluation metrics which measure the agreement between the clusters produced by an automatic clustering algorithm and the ground-truth of classes.

2.2 Document Representation and Similarity

Document representation which directly determines the transformed data structure and information from the original document collection is an important step for text-based information retrieval. For document clustering, some important concepts, such as document similarity, are possibly defined based on the document representation

model. The most widely used document representation model is the vector space model (VSM) [18], in which each document is represented as a vector with each valid term (word) as a dimension. Although frequently adopted in many text mining researches, much helpful information except frequency is not considered in VSM. The phrase-based model (PBM) uses phrases or sequences of words as units, with which word order information is reserved. Next, we address them in details.

2.2.1 Document Representation

According to [19], the construction of the vector space model of documents consists of two basic steps. First, to define a term as the granularity of the text unit and use it as a feature. This unit can be a single word, n-grams, noun phrase, etc. Second, to establish a weighting scheme which measures the importance of each term. In the original vector space model, or “bag of words” model, each valid word is served as a term. For a document set containing n documents and s unique words, a document-term co-occurrence matrix $\mathbf{X}_{n \times s}$ then is used to represent the whole document dataset. Figure 2.1 illustrates the “bag of words” representation of a document and Figure 2.2 illustrates the vector-represented document. To decide the importance of each term in a document, several term weighting schemes have been proposed, including binary term frequency, where a term is weighted according to whether it is occurred in the document, simple term frequency, where a term has a high weight if it appears in the document frequently, and Term Frequency-Inverse Document Frequency (tf-idf) [20], where the weight of a term in a document is decided based on the occurrence frequency of this term in this document as well as the total frequency of this term appeared in all the documents. Since only rely on term frequency in the related document can give some frequently appeared words high weights which in fact contain little discriminative information. To avoid this situation, inverse document frequency is multiplied with term frequency together to determine the final weight of each term in a document. The tf-idf value for a term p in document i is calculated as

$$x_{ip} = tf_{ip} \cdot \log \frac{n}{idf_{ip}} \quad (2.1)$$

where tf_{ip} and idf_{ip} represent the frequency of the p th term appeared in the i th document and the number of documents containing the p th term, respectively. The

Figure 2.1: The “a bag of words” model.

effect of using tf-idf is that a term p occurring frequently in a document i but rarely in the rest of the collection has 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 small subset of words in the whole word dictionary. This makes the document-term matrix very sparse as most entries are zero. This high dimensional and sparse space can be very noisy and difficult for normal clustering algorithms to perform effectively, because the high dimensionality cause the ‘dimension curse’ problem, and meanwhile it is difficult to find a proper similarity measure defined in such a high dimensional space. To alleviate this problem, similarity can be calculated in a transformed space where the dimension is reduced through feature selection or feature extraction processes. To improve the information contained in the representation model as well as to reduce the computation requirement, one key problem is to determine a small number of features and their weights which are used to characterize documents.

Although VSM is simple and widely used, it only captures the frequency information of each term while other information which is probably useful for document clustering is ignored. Various improved representation models have been proposed

Figure 2.2: Vector representation of a document.

to enrich the information in the representation by incorporating word sequence information, semantic meanings of words or background knowledge. Phrase-based models such as Suffix Tree Clustering (STC) [4] and Document Index Graph (DIG) [7] describe documents by phrases to count in order information by which the words occur in a document. Since for some cases, though the vector representations of two text units are identical, the meanings of them are obviously different when the words occurred in different orders. A phrase-based model encodes in word order information to improve the representation of a document, hence to improve the document similarity and finally the clustering accuracy. Another motivation of a phrase-based model is that phrases can be used to describe or label a document cluster, while only separate terms are used in the vector space model to interpret a cluster. It is generally agreed that phrases are more descriptive than individual words. Benefitted from the development of Natural Language Processing (NLP) techniques and the availability of large thesaurus, more sophisticated information from the original document collection may be obtained. In [21], linguistic-based methods together with statistical-based ones are combined to select a small number of words as features based on both syntactic position and frequency in the document. In [22], terms and phrases are weighted according to how they contribute to the semantic meaning of the sentences rather than just the frequency. And recently, using external ontology, such as WordNet [23] and Wikipedia [24], to enrich document representation by semantic meanings has been

shown to be effective. In [24], each document is mapped into one or more concepts and categories in Wikipedia so that the similarity between two documents is not only based on the content of each document, e.g, the words contained, but also the concepts and categories they are related according to Wikipedia.

2.2.2 Document Similarity

Proximity measures including similarity and dissimilarity are critical building blocks in clustering. In the simplest models, a measure of (dis)similarity can serve as a criterion of validity. More generally, (dis)similarity measure plays a critical role in most of the clustering algorithms. The same (dis)similarity measure with different algorithms or different (dis)similarity measures with the same algorithm, can yield different results for the same dataset. Document similarity can be defined based on the document representation model, like vector space model (VSM) and phrase-based model (PBM), or can be calculated directly without representing the documents into a specific model. We only review several widely used document similarity measures as well as some recently developed ones.

When VSM is used for document representation, 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

$$S_{cos} = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \quad (2.2)$$

where $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^s$ represent two documents, $\langle, \rangle$ is the inner product of two document vectors, and $\|\cdot\|$ is the l_2 norm or Euclidean norm of a vector, i.e.,

$$\begin{aligned} \langle \mathbf{x}_i, \mathbf{x}_j \rangle &= \sum_{p=1}^s (x_{ip} \cdot x_{jp}), \\ \|\mathbf{x}_i\| &= \sqrt{\sum_{p=1}^s x_{ip}^2} \end{aligned} \quad (2.3)$$

There are other measures such as Jaccard measure may be used to measure document similarity based on the "bag of words" model. With Jaccard measure, two documents are regarded to be more similar if they share more common words.

Document similarity may also be calculated based on other representation models in terms of other types of information. The suffix tree model (STM) reserves the complete word order information but no frequency information is explicitly recorded. Although the order information provides more accurate local information which can be helpful to improve the accuracy of a similarity measure, it is not sufficient enough for only using order information to capture the content of a document when two documents share few common phrases but actually similar with each other in content. Rather than using a single type of information, recently proposed similarity measures for documents attempt to combine both the frequency and the order information [7, 19, 25]. Significant improvements in clustering results are shown in these studies when hybrid similarities are used. 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:

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

with

$$\varphi_{ST} = \frac{|E^i \cap E^j|}{|E^i \cup E^j|} \quad (2.5)$$

where E^i and E^j are edges in the suffix tree related to two documents respectively. This is an explicit way to combine two types of information based on two document representation models. A different way to incorporate the frequency into the STM-based similarity is given in [25]. Each document is first encoded into STM by which the words order information is reserved, and then the resulting nodes in the suffix tree are mapped into VSM to further count in the frequency information. In the second step, nodes in the constructed suffix tree are treated as terms in VSM, and documents are represented as vectors with these terms as dimensions. The same tf-idf weighting scheme is used to weight the terms and finally cosine is used to calculate the document similarity. Although these two hybrid similarity measures are both based on the STM and use cosine measure for the VSM-based representation, the combinations of two types of information in these two approaches are conducted on two different levels. Another hybrid similarity measure proposed in [7] is similar to Eq. (2.4) except that the order information is obtained based on the DIG document

model that considers several factors of the matched phrases, such as significance level, length, and frequency. A consistent conclusion made in these studies is that better performance in clustering is achieved by using a hybrid similarity to consider both order information and frequency, compared with using a similarity to only consider 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 [26], the results returned by a search engine are represented with features derived from shared links. In [27], 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 [28], 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.

2.3 Data Formats

A document dataset is finally represented in a mathematical form which can be handled by automatic clustering algorithms. Most of the existing clustering approaches cluster documents based on vector-based data or proximity data. We first discuss these two basic data formats, and then discuss multi-type relational data, which in our view is a more suitable data format for encoding document datasets which are described by multiple types of data from different angles.

2.3.1 Vector-based Data and Proximity Data

Many clustering approaches such as the well known k-means clustering, cluster documents by representing each document into an s -dimensional vector, where each dimension or feature may be a single word, a phrase, or a derived attribute. A document

Figure 2.3: Two basic data formats: vector-based data and relational data

dataset with n documents then is represented as a document-feature matrix $\mathbf{X}_{n \times s}$ where each x_{ip} is the weight of the p th feature in the i th document. Other than clustering documents with vector-represented data, some clustering methods such as graph-based method and kernel-based method produce clusters of objects only based on the pairwise document proximities. This type of data is called proximity data or relational data. An $n \times n$ proximity matrix $\mathbf{R}$ may be used to record the pairwise relationships of documents. Figure 2.3 shows the types of data formats in terms of an object-feature matrix and object-object matrix, respectively. For the proximity matrix, different approaches may require it to have different properties. Some clustering approaches such as kernel-based clustering require it to be semi-positive definite; many approaches just require it to be non-negative and symmetric; and some others may be able to handle a proximity matrix with a more general form which can be asymmetric or even with negative values.

Compared with representing documents as vectors of numerical features, the document-document proximity data is more generic. In some cases, it is ineffective or difficult to define a set of numerical features to properly describe the dataset while pairwise relationships may be more naturally or easily obtained. The proximity matrix may be calculated from vector-based data with some similarity metrics, e.g., cosine similarity, or defined based on any other proper document representation models such as suffix tree, or directly derived from data with relational information, such as links among documents.

Figure 2.4: Graph representation of homogeneous relational data. Partitioning of a graph gives the clustering result of the dataset.

Since the pairwise proximity data contains a relation between objects of the same type, e.g., document-document relation, it is said to be homogeneous relational data. Based on the proximity matrix or affinity matrix, a graph can be constructed. Each vertex of the graph corresponds to a document in the dataset and the weight on an edge of the graph is the proximity or association between two documents. Therefore, the proximity-based clustering problem may be treated as the problem of partitioning of the corresponding graph. Consider a small dataset containing four documents, and the non-zero entries of the proximity matrix are $r_{12} = 0.8$, $r_{14} = 0.6$, $r_{23} = 0.5$, $r_{34} = 0.7$. Figure 2.4 illustrates the graph representation of this proximity data and the possible partitioning of the graph assume the number of clusters is 2.

However, in some real-world applications, the data involves relations among several types of objects or entities. It could be possible to apply some data preprocessing to transform such kind of data into one of the two basic data formats, namely vector-based or homogeneous relational data. However, the transformation process requires additional computations and information is lost. Next, we discuss the multi-type relational data format for handling such kind of cases.

2.3.2 Multi-type Relational Data

Multi-type relational data consists of relations among multiple types of objects. Such kind of data is represented by a set of matrices where each matrix represents a relation between two types of objects [29–31]. Some other models have also been used to represent multi-type relational data, e.g., in [32], a relation graph is used to represent the multi-type relational data where each relation that is represented by a high-dimensional tensor can involve multiple types of objects. Here we only discuss the way of representing multi-type relational data as a set of pairwise relations, i.e., a relation is recorded by a matrix or a two-dimensional array.

As shown in Figure 2.4, homogeneous relational data corresponds to a standard graph which contains a single type of vertices; multi-type relational data which involves r types of objects may be represented as a r -partite graph. Here a r -partite graph is a graph whose vertices can be partitioned into r disjoint sets where no two vertices within the same set are connected. Figure 2.5 illustrates a bi-partite graph and its possible partitioning. Instead of showing all the detailed connections on object level, we may want to illustrate the structure of relational data on object-type level. It is shown from Figure 2.6a that a classic graph contains a relation between the same type of objects, and a bi-partite graph contains a relation between two types of objects. Similarly, the structure of a r -partite graph is shown in Figure 2.6b. Since there is no relation between the same type of objects on a r -partite graph, approaches based on this model are unable to consider the link information between the same type of objects. More generally, relational data should contain both relation between the same type of objects, and relation between different types of objects as shown in Figure 2.7.

Multi-type relational data may form different structures depending on the availability of relations. A star-structure is a special case where relations only exist between the central type and several attribute types as illustrated in Figure 2.8a. The central type is always the target type that is interested. We will refer it as extended star structure when the central-type to central-type relation is further combined with a standard star-structure as illustrated in Figure 2.8b.

We have discussed earlier that a document dataset can be represented as a document-feature matrix or a document-document matrix. Although these two data formats are

Figure 2.5: Bi-partite graph representation of bi-type relational data. Partitioning of a bi-partite graph gives the clustering result of the corresponding bi-type relational data. Different object-types are denoted with different shapes.

Figure 2.6: Structures of relational data: (a) a classical graph contains a relation between objects of the same type, and a bi-partite graph contains a relation between two types of objects; (b) A r -partite graph contains relations between objects of r different types.

Figure 2.7: General structure of r -type relational data with relations between different types and relations between the same types.

(a) Star-structure

(b) Extended star-structure

Figure 2.8: Relational data with two special structures: star-structure and extended-star structure.

generally suitable for text document data, it may not be sufficient enough for those document datasets where multiple types of data are used for the clustering process. To be able to encode all types of data, we may represent such a document dataset as multi-type relational data, where each relation describes one type of data. For example, for a document dataset as illustrated in Figure 2.9, three types of data are available: the links among documents, the words each document contains, and the persons who wrote or published the documents. We may consider this document dataset as a three-type relational data with three relations, i.e., *document*, *word* and *author* these three object types together with relations between document and the other two types, i.e. document-word relation and document-author relation, as well as the document-document relation. Since each relation corresponds to a matrix, a multi-type relational data is associated with a set of relation matrices. For the data in Figure 2.9, we may use three matrices $\mathbf{R}_{11}$, $\mathbf{R}_{12}$ and $\mathbf{R}_{13}$ to record the document-document, document-word and document-author relations, respectively. Although in the vector space model document data is also represented as a document-word matrix, words in this representation model are features and only documents are objects; while here both documents and words are objects, and the document-word matrix records the relation between these two types of objects.

2.4 Existing Document Clustering Approaches

Once the target data is represented into a mathematic form, a clustering algorithm is applied to produce the final clusters. In this section, we review different existing clustering approaches in the literature and discuss their unique properties. Instead of exhaustive covering various types of clustering methods, we examine most widely used and recently developed methods for document clustering. Some clustering approaches, such as density-based clustering, grid-based clustering, are not covered as they are not well suited for dealing with document data though they are popular for handling some other types of data, e.g., spacial data with arbitrary shaped clusters.

Figure 2.9: Document dataset with relations among multiple types of objects. Different object types are denoted with different shapes.

2.4.1 Hierarchical Clustering

A hierarchical method creates a hierarchical decomposition of a given dataset called dendrogram. From the dendrogram, partitioning with different granularity can be selected by cutting it from different levels, the higher level the smaller number of clusters with larger sizes. According to the way that the dendrogram forms, hierarchical clustering can be further grouped into bottom-up agglomerative approach and top-down divisive approach [1, 33]. Agglomerative hierarchical clustering starts with each single object as a cluster and then gradually merges clusters until all objects are in the same 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 clusters into smaller ones until all of the clusters only contain one object. The proximity measure of two subsets of objects generalized from the proximity between individual objects is used to decide the operation of merging or splitting. This kind of derived proximity is called linkage metric. Major inter-cluster linkage metrics include single link [34], complete link [35] and average link or Unweighted Pair Group Method with Arithmetic (UPGMA). Given two clusters $\mathcal{C}_p$ and $\mathcal{C}_q$, all these three linkages define the distance of two clusters based on the distances between all pairs of x_i and x_j with

$x_i \in \mathcal{C}_p$ and $x_j \in \mathcal{C}_q$. The minimum, maximum and average are used by single-link, complete-link and average-link, respectively. Linkage metric significantly affects hierarchical clustering methods since it reflects the particular definition of closeness of two clusters.

Hierarchical clustering has been studied for several decades and it is still widely used as an important clustering approach today. Classic linkage-based hierarchical clustering does not require the data to be represented as vectors as only pairwise (dis)similarities are used. Organizing a set of documents into a hierarchical structure is very useful in many applications for conveniently browsing the content of the document set. However, hierarchical clustering approaches, such as complete and UPGMA linkage based ones, suffer from time complexity which is typically $O(N^2 \log N)$. Other than heavy computations needed for large datasets, the disadvantage of hierarchical clustering is the vagueness of termination criterion and, most hierarchical clustering approaches do not revisit clusters once constructed. In [36], the performances of hierarchical clustering and partitioning clustering with different criterion functions and merging schemes have been tested for document clustering. The experimental results in this study show that comparing with hierarchical clustering algorithms, partitioning ones are more favorable for large document clustering regarding to higher clustering accuracy and lower computational requirements.

2.4.2 Partitioning Clustering

Unlike hierarchical clustering which produces a series of nested clusters, partitioning clustering generates a flat partitioning of the target dataset. Given a set of n objects, partitioning clustering groups these n objects into k partitions, where each partition is a cluster. Typically $k \ll n$, and each cluster at least contains one object. The clustering is hard if each object is only grouped into one cluster. The problem of hard k -partitions clustering can be defined as below:

Definition 2.1 *Given a dataset $\mathcal{X}$ of n objects, a hard k -partitions clustering is to partition $\mathcal{X}$ into k disjoint subsets, i.e., $\mathcal{X} = \mathcal{C}_1 \cup \mathcal{C}_2, \dots \cup \mathcal{C}_k$, $\forall c, \mathcal{C}_c \neq \emptyset$, and $\forall c \neq f, \mathcal{C}_c \cap \mathcal{C}_f = \emptyset$, where each subset $\mathcal{C}_c$ corresponds to a cluster.*

A critical issue of partitioning approaches is what kind of a partitioning is a good one? In other words, the quality measure of a partitioning has to be well-defined. A general criterion of a good partitioning is that objects in the same clusters are similar or related to each other whereas objects in different clusters are very different. As discussed in Chapter 2.2.2, similarity measure is another important factor in clustering.

Since to find a global optimal partitioning would require exhaustive enumeration of all the possible partitioning, heuristic methods are usually used instead in most applications. The two most well known heuristics are k-means and k-medoids. These two approaches are “center-based” approaches, where each cluster is represented by a centroid in k-means or a medoid in k-medoids. A centroid is the mean of objects in the same cluster, and a medoid is a representative object selected from the given dataset. Both of k-means and k-medoids try to find a partitioning that the total distance or dissimilarity from all objects to their associated cluster centers, i.e, the centroids or medoids, is minimized. Since the centroid is defined in a vector space, k-means takes vector-based data as input; while k-medoids only requires pairwise dissimilarities since the k medoids are selected from the given dataset. Here we only discuss the k-means algorithm and some of its important extensions. K-medoids will be reviewed in the next Chapter as one relevant work to our first proposed approach.

The k-means algorithm looks for local solutions to minimize the following criterion [37]:

$$J_{k-means} = \sum_{c=1}^k \sum_{\mathbf{x}_i \in \mathcal{C}_c} \|\mathbf{x}_i - \delta_c\|^2 \quad (2.6)$$

where δ_c is the centroid of cluster c calculated as the mean of all objects in that cluster, i.e.,

$$\delta_c = \frac{1}{|\mathcal{C}_c|} \sum_{\mathbf{x}_i \in \mathcal{C}_c} \mathbf{x}_i \quad (2.7)$$

Let

$$d_{ic}^2 = \|\mathbf{x}_i - \delta_c\|^2 \quad (2.8)$$

to denote the distance between the centroid of cluster c and an object in this cluster $\mathbf{x}_i$. Then Eq. (2.6) becomes

$$J_{k\text{-means}} = \sum_{c=1}^k \sum_{\mathbf{x}_i \in \mathcal{C}_c} d_{ic}^2 \quad (2.9)$$

The iterative relocation algorithm as in Algorithm 1 [38, 39] is used in k-means to find the local solution for this objective. The iteration starts with an initial guess of the partitioning, and then iteratively updates the centroids and cluster assignments to improve the quality of the clusters. The centroid of a cluster is the mean of objects that are currently assigned in this cluster and each object is reassigned to a cluster if the distance between this object and the centroid of this cluster is the smallest.

Algorithm 1: K-means

input : data matrix $\mathbf{X}$, and the number of clusters k
output: clusters $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$ and centroids $\delta_1, \delta_2, \dots, \delta_k$
Initialize partitioning $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$;
repeat
 for $c = 1, 2, \dots, k$ **do**
 | update $\delta_c = \sum_{\mathbf{x}_i \in \mathcal{C}_c} \mathbf{x}_i$
 end
 for $i = 1, 2, \dots, n$ **do**
 | assign $x_i \rightarrow \mathcal{C}_f$ with $f = \arg_c \min d_{ic}^2$ and $d_{ic}^2 = \|\mathbf{x}_i - \delta_c\|^2$
 end
until *no change in the cluster assignments*;

There are many extensions and variants of k-means with modifications in the objective function or algorithm. The k-means algorithm in Algorithm 1 is sensitive to initialization and only converges to local solutions. Other than forming k clusters at the same time, bisecting k-means produces two clusters first and then partitions one of the existing clusters into another two clusters until k clusters are formed. Algorithm 1 is called batch k-means, where all objects to be clustered are given at once, and in each iteration, the centroids are updated after all the objects having been assigned. In online or incremental k-means, such as competitive learning [40] and self-Organizing Maps [41], objects are coming one by one and only the centroid of cluster that the coming object is assigned to is updated. Other than modifying the algorithm, efforts have

also been made on developing modified objective functions. The spherical k-means proposed by Dhillon and Modha [42] is a cosine-based k-means type approach for handling document data. In the original k-means, Euclidean norm is used to measure the distance between each object to the centroids as in Eq. (2.8). As discussed early, cosine measure is more effective than Euclidean norm for document similarity measure, therefore, a cosine-based k-means is more favorable for documents. In spherical k-means, each document is represented as a vector with a unit length. All document vectors are on a high-dimensional unit sphere, and the cosine similarity between two documents becomes the angle between the two vectors.

The formulation of k-means and its many variants such as spherical k-means implicitly assumes that the underlying clusters are hyper-spherical or hyper-ellipsoidal. Thus these approaches fail to detect clusters in datasets with nonlinearly separable boundaries. Kernel k-means is proposed to handle data with non-linearly separable clusters. In kernel k-means (KKmeans) [43, 44], objects are first mapped into a high or infinite dimensional space, i.e., $\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i)$, and then k-mean clustering is performed in this mapped space called feature space. Mapping to some high dimensional space before clustering increases the probability to make those clusters which are non-linearly separable in the input space becomes linearly separable in the feature space, so that it tends to be easier to find those clusters. The key trick is that the non-linear mapping needs not to be performed explicitly. Inner products between all pairs of objects in the feature space can be calculated with a kernel function κ in the original data space, i.e.,

$$\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j) \quad (2.10)$$

The matrix $\mathbf{K}$ with $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ is called kernel matrix. According to Mercer's Theorem, $\mathbf{K}$ is a positive semi-definite matrix, and any positive semi-definite matrix could be treated as a kernel matrix formed by evaluating some mercer kernel on all pairs of objects in the input space.

2.4.3 Fuzzy Clustering

In a hard clustering or crisp clustering, like k-means, each object is only assigned to one cluster and the cluster assignment of each object takes a binary value to represent

whether or not an object belongs to a cluster. Principles from the fuzzy set theory are applied in fuzzy clustering to allow each object to belong to more than one clusters with different degrees [45–47]. The degree of belonging of object i with respect to cluster c is called a membership and denoted as u_{ic} . This membership represented cluster assignment is a continuous variable that takes values between 0 and 1. Fuzzy membership is typically required to be nonnegative and the sum of fuzzy memberships of each object to all the clusters is 1, i.e.,

$$\forall c, i, u_{ic} \geq 0 \quad \text{and} \quad \forall i, \sum_{c=1}^k u_{ic} = 1 \quad (2.11)$$

This constraint is called the “probabilistic constraint” [48]. With this constraint, the fuzzy memberships are said to be “relative”, i.e., the value of the membership of an object with respect to a cluster is dependent on the the relationship between this object and other clusters.

Fuzzy representation of cluster assignments is able to describe overlaps between clusters. This enables fuzzy clustering to capture the data structure more realistically, because most real-world datasets naturally contain some degree of overlaps between categories, e.g., one document may span multiple topics which naturally requires it to be assigned to all the related topics possibly with different degrees. Therefore, fuzzy clustering is considered to be more precise and informative than those crisp counterparts. Furthermore, taking continuously values of memberships makes the objective function smoother. Hard partitioning clustering algorithms move among a discrete lattice of possible objective values until reaching a local optimum where a move to a neighboring value dose not improve the function value. Relaxing the binary assignments to memberships enables the algorithm move along a continuous space which makes the optimization process more robust and consistent [1].

As a fuzzy extension of k-means, fuzzy c-means [47] is one of the most well known fuzzy clustering approaches. In the original version of fuzzy c-means, an exponent m called fuzzifier is introduced to produce the fuzziness of memberships. The objective of fuzzy c-means proposed in [47] is to minimize the following criterion:

$$J_{FCM} = \sum_{c=1}^k \sum_{i=1}^n u_{ic}^m d_{ic}^2 \quad (2.12)$$

where d_{ic}^2 is the distance between object $\mathbf{x}_i$ and the centroid of cluster c denoted as $\delta_c \in \mathbb{R}^s$. The parameter $m > 1$ in Eq. (2.12) is the fuzzifier which controls the level of fuzziness of memberships, i.e., the larger m is, the fuzzier the memberships tend to be. J_{FCM} defined in Eq. (2.12) calculates the total distance from all objects to their (softly) associated cluster centroids. d_{ic}^2 is typically calculated with the Euclidean norm like in k-means as Eq. (2.8).

Let $\mathbf{u}_c = [u_{1c}, u_{2c}, \dots, u_{nc}]^T$ represent the memberships of all objects in cluster c . Then the fuzzy c-means clustering formulated above is to finding optimal solutions of $\mathbf{U}_{n \times k} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k]$ and $\mathbf{\Delta}_{s \times k} = [\delta_1, \delta_2, \dots, \delta_k]$ to minimize J_{FCM} . Since it is difficult to solve the above problem directly, iterative algorithm is always used to find local solutions of $\mathbf{U}$ and $\mathbf{\Delta}$ through successively updating $\mathbf{U}$ and $\mathbf{\Delta}$ in an alternating manner. With the method of Lagrange multiplier, the following update formulas are derived [47]:

$$\delta_c = \frac{\sum_{i=1}^n u_{ic}^m \mathbf{x}_i}{\sum_{i=1}^n u_{ic}^m} \quad (2.13)$$

$$u_{ic} = \frac{\left(\frac{1}{d_{ic}^2}\right)^{\frac{1}{m-1}}}{\sum_{f=1}^k \left(\frac{1}{d_{if}^2}\right)^{\frac{1}{m-1}}} \quad (2.14)$$

Instead of using fuzzifier m , the technique of regularization is another frequently used way to produce a fuzzy partitioning of a given dataset. More specifically, the objective function of regularization based fuzzy c-means is formulated by integrating an additional term, referred as the regularization term or penalty term, into the objective function of k-means. Several variants of fuzzy c-means with different regularization terms, such as entropy regularization and quadratic regularization, have been developed [49, 50]. The objective function of quadratic regularized fuzzy c-means here referred as FCMq is in the form as below

$$J_{FCMq} = \sum_{c=1}^k \sum_{i=1}^n u_{ic} d_{ic}^2 + \frac{\eta}{2} \sum_{c=1}^k \sum_{i=1}^n u_{ic}^2 \quad (2.15)$$

where d_{ic}^2 is calculated by Eq. (2.8), and parameter η is the weight of the regularization term, which plays a similar role as the fuzzifier m in Eq. (2.12). The larger η is, the

more the regularization term contributes and the fuzzier the memberships tend to be. Update equations of δ_c and u_{ic} for FCMq are derived as [50]:

$$\delta_c = \frac{\sum_{i=1}^n u_{ic} \mathbf{x}_i}{\sum_{i=1}^n u_{ic}} \quad (2.16)$$

$$u_{ic} = \frac{1}{|\mathcal{K}^+|} + \frac{1}{\eta} \left(\frac{1}{|\mathcal{K}^+|} \sum_{f \in \mathcal{K}^+} d_{if}^2 - d_{ic}^2 \right) \quad (2.17)$$

where $\mathcal{K}^+ = \{f : u_{if} > 0\}$. More detailed discussions on how to calculate $\mathcal{K}^+$ can be found in [50].

The same as k-means, both fuzzifier-based and quadratic regularization-based fuzzy c-means approaches use Euclidean norm as distance measure. A cosine-distance based fuzzy c-means is proposed in [51] for document clustering, which could be considered as a fuzzy version of the spherical k-means. We will review this approach in more details in Chapter 4.

Although introducing of fuzzy membership makes fuzzy c-means more effective than k-means, like k-means, fuzzy c-means is known to be sensitive to noise. A possible solution to this problem is to relax the summation constraint in Eq. (2.11) to give the so-called possibilistic membership [52]. Other than enhanced robustness, possibilistic clustering can be used to detect the clusters one by one without specifying the number of clusters with the help of parameter selection procedures. With these merits, possibilistic clustering has been applied to many tasks, e.g., a possibilistic *c*-spherical shells algorithm has been used to identify ring patterns in [53], and a graded possibilistic approach has been proposed for interactive post processing in the segmentation of MRI images [54]. To handle clusters with non-linearly separable boundaries, a kernel-based possibilistic clustering has been proposed in [55]. However, it is known that possibilistic approach frequently produces identical clusters [56]. Recent studies in [57, 58] show that combinations of fuzzy and possibilistic approaches are able to enhance the robustness of clustering algorithms and achieve better results.

2.4.4 Probabilistic Model-based Methods

This type of clustering tries to fit the data into some probabilistic models. The most widely used probabilistic generative model-based clustering is the mixture models. The

mixture models method is based on the assumption that each cluster is mathematically represented by a parametric model, and the data is generated from a mixture of the k models. Each model also called component is a probabilistic distribution. The goal is to decide the parameters of each component as well as the mixing coefficients of these components to make the models fit the data best through the maximization of the likelihood function [59]. The Expectation-Maximization (EM) algorithm [60] has been applied to iteratively estimate the parameters. Gaussian mixture models (GMMs) is a popular generative model-based approach, where each component is a Gaussian distribution. For the similar reasons as spherical k-means, in [61], Banerjee et al proposed a mixture of vonMises-Fisher models approach for text clustering, where the vonMises-Fisher distribution could be seen as a spherical analog of Gaussian.

Other than applying conventional mixture models approaches for document clustering, topic modeling approaches become more popular for text mining. Topic modeling is based upon the idea that documents are mixtures of topics, where a topic is a probability distribution over words [62]. A topic model is a generative model for documents: it specifies a simple probabilistic procedure by which documents can be generated. The posterior is a topical decomposition of the documents. Since the topic model is over words not over documents as in classic mixture models, topic modeling can be used for more text mining tasks, such as statistical inference. Popular topic modeling approaches include probabilistic latent semantic indexing (PLSI) [63], latent Dirichlet allocation (LDA) [64] and its Bayesian variants [65]. Topic modeling approaches assume that documents are independent and each document is “a bag of words”. Since links among Web documents provide useful information for Web document analysis, recently proposed models-based approaches such as [16, 17, 66] establish topic models of documents by considering both content (words) and structure (links) information.

2.4.5 Graph Partitioning Methods

In discussion of relational data in 2.3, we have mentioned that document clustering may also be formulated as a graph partitioning problem based on the proximity matrix. When a graph is constructed from the proximity matrix or affinity matrix with each

document as a vertex and the proximity of two documents as the weight of the edge that connects these two vertices, the graph partitioning problem is to cut the graph into k disjoint parts so that some criterion is optimized. Since finding the exact way to cut is an NP hard problem, some heuristic algorithms are used to find approximate solutions. Recently, linear algebraic based method called spectral clustering has been extensively studied for graph partitioning. A spectral clustering approach first performs eigendecomposition of the affinity matrix based on which k eigenvectors are selected, where k equals to the number of clusters. Finally, k-means is typically used to cluster vectors in the k dimensional space constructed by those eigenvectors. The difference in constructing the affinity matrix and selecting eigenvectors results in different graph partitioning approaches, such as ratio cut [67], normalized cut [68, 69], and min-max cut [70]. Given a graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{R})$, where $\mathcal{V}$ is a set of vertices, $\mathcal{E}$ is a set of edges and matrix $\mathbf{R}$ records the weights of all the edges, the objective of normalized cut is to find a partitioning of $\mathcal{V}$ such that the following criterion is minimized

$$\sum_{c=1}^k \frac{\text{links}(\mathcal{V}_c, \mathcal{V} \setminus \mathcal{V}_c)}{\text{degree}(\mathcal{V}_c)} = \sum_{c=1}^k \frac{\sum_{\mathbf{x}_i \in \mathcal{V}_c, \mathbf{x}_j \notin \mathcal{V}_c} r_{ij}}{\sum_{\mathbf{x}_i \in \mathcal{V}_c, \mathbf{x}_j \in \mathcal{V}} r_{ij}} \quad (2.18)$$

Since $\sum_{\mathbf{x}_i \in \mathcal{V}_c, \mathbf{x}_j \notin \mathcal{V}_c} r_{ij} = \sum_{\mathbf{x}_i \in \mathcal{V}_c, \mathbf{x}_j \in \mathcal{V}} r_{ij} - \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{V}_c} r_{ij}$, the above minimization problem is equivalent to the maximization problem given as below

$$\max \sum_{c=1}^k \frac{\sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{V}_c} r_{ij}}{\sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{V}_c} r_{ij}} \quad (2.19)$$

Since each vertex only belongs to one partition, graph partitioning is hard clustering.

Although spectral clustering performs well in several tasks such as image segmentation and document clustering, the demand of eigendecomposition makes it infeasible for large datasets. In [71] it is shown that the objective function of weighted graph partitioning is equivalent to weighted kernel k-means, which indicates that graph partitioning can be solved without calculation of eigenvectors with weighted kernel k-means. To scale to larger graphs, a multi-level graph partitioning algorithm called METIS is further incorporated [71]. The METIS approach first introduced in [72] consists of three phases, where the graph is first coarsened by collapsing vertices and edges, and then the coarsened graph is partitioned, and finally the graph is uncoarsened in order to get a partitioning of the original graph.

2.4.6 Nonnegative Matrix Factorization Methods

In [73], Lee and Seung show the usefulness of non-negative matrix factorization (NMF) in pattern recognition, text mining and DNA gene expressions analysis. Later, Xu et al applied NMF for document categorization [74]. In NMF, the non-negative document-word matrix $\mathbf{X}$ is factorized into two low-dimensional non-negative matrices $\mathbf{U}$ and $\mathbf{V}$. The goal is to minimize the factorization error which is measured by some distortion criteria, such as the least squares error and the Kullback-leibler (KL) divergence. The least squares error based NMF problem is given as

$$\min_{\mathbf{X}, \mathbf{U}, \mathbf{V} \geq 0} \|\mathbf{X} - \mathbf{UV}^T\|_F \quad (2.20)$$

Where matrix $\mathbf{V}$ records the k base topics of the k document clusters and the other matrix $\mathbf{U}$ records the linear combination coefficients of the base vectors for each of the input documents. In other words, the factorization process is to represent each of the input documents as a linear combination of the k base vectors or topics. In the context of clustering, each column vector of $\mathbf{V}$ could be seen as a center vector of a cluster while $\mathbf{U}$ is the cluster assignment matrix.

Multiplicative update rules of $\mathbf{U}$ and $\mathbf{V}$ for the above minimization problem are derived as [75]:

$$u_{ic} = u_{ic} \frac{(\mathbf{XV})_{ic}}{(\mathbf{UV}^T\mathbf{V})_{ic}} \quad (2.21)$$

$$v_{jc} = v_{jc} \frac{(\mathbf{X}^T\mathbf{U})_{jc}}{(\mathbf{VU}^T\mathbf{U})_{ic}} \quad (2.22)$$

Different from other matrix factorization methods which require the k base vectors to be orthogonal to each other, such as SVD, the only type of constraints in NMF is the non-negativity. This property allows only additive liner operation and makes NMF a “parts-based” representation, which is more meaningful and easier to be interpreted. However, it is shown in later studies that the non-negative constraint itself does not necessary contributes to “parts-based” or localized representation. Additional regularization terms need to be incorporated to control the sparsity [76, 77].

Instead of two-matrices type factorization, some approaches factorize the original input matrix into three matrices [78, 79]. Other extensions of NMF have also been proposed by relaxing the non-negative constraint of the input matrix or adding other types of constraints. The semi-NMF proposed in [80] relaxes the negative constraint on the target matrix so it can handle data with negative values. The convex NMF poses additional restrictions on the basis vectors to make them lie within the space spanned by the input document vectors. Symmetric NMF [79] and kernel NMF [81] deal with a symmetric matrix, which may be a similarity matrix, or a kernel matrix. In a recent NMF extension [82], the authors proposed to consider geometrical structure of the data through incorporating graph regularization.

2.4.7 Co-clustering and Multi-type Relational Clustering

All the document clustering approaches that so far have been discussed only produce document clusters at the end of the clustering process. Now we review “multi-way clustering” approaches, which simultaneously cluster multiple types of objects or entities. We begin with co-clustering, or two-way clustering, which is applied to simultaneously cluster documents and words.

2.4.7.1 Co-clustering

The “bag of words” based vector space model of document representation always results in a high dimensional and sparse space, which is very difficult to find meaningful clusters. One possible way to handle high dimensional data is to transform the original space with feature selection or dimension reduction, such as PCA (principal component analysis) to a new space with a low dimensionality so that clusters may be easier to be found. Other than applying data transformation as an independent process before clustering, co-clustering approaches have been proposed to deal with high-dimensionality during clustering. Traditional document clustering approaches are “one-side” clustering only generates document clusters given the document-term co-occurrence matrix. Differently, co-clustering treats *document* and *term* as two object types, which are clustered simultaneously [6, 83–87]. In co-clustering, documents

are clustered based on the term clusters rather than individual terms as in traditional clustering, and the document clusters in turn are used to improve term clusters.

The co-clustering process usually involves dynamic dimension reduction by extracting or selecting a smaller number of features based on clusters of one type to represent and cluster objects of the other type. Take co-clustering of documents and terms as an example. In some approaches, the statistical information of each of the term clusters are calculated as new features to describe the documents, based on which documents are clustered; while in some other approaches, documents are represented by a subset of terms in the associated term cluster(s), based on which they are clustered. The clustering of terms based on document clusters is performed in similar ways. Compared with one-side clustering, co-clustering is shown to be more powerful for high dimensional data, such as text and microarray data. It has been shown in many studies that even the goal is only to obtain document clusters, clustering terms simultaneously with documents helps to improve the quality of the document clusters.

Like one-side clustering, different types of co-clustering approaches have been developed with different mathematic frameworks. A spectral co-clustering is studied in [6]. In this approach, the simultaneous clustering of documents and words is formulated as a bipartite graph partitioning problem, which can be solved with SVD decomposition of the document-word matrix. An information theoretic-based co-clustering proposed in [83] formulates the co-clustering problem as to find co-clusters to maximize the mutual information between the clustered random variables. This work is extended in [84] to a more general framework called Bregman co-clustering, where any Bregman divergence can be used as the distortion criterion between the original data matrix and the co-cluster reconstructed matrix. Other than these, several fuzzy co-clustering approaches have also been proposed such as Fuzzy co-clustering for documents and words (FCoDok) [85] and dual fuzzy-possibilistic co-clustering [58]. Approaches in [79, 87] are matrix factorization based co-clustering.

2.4.7.2 Multi-type Relational Clustering

We have discussed in Chapter 2.3 that a classic graph represents homogeneous relational data, and a bi-partite graph represents a bi-type relational data as illustrated

in Figure 2.5. Since co-clustering can be formulated as bi-partite graph partitioning, we may say that co-clustering is actually bi-type relational data clustering, which produces clusters of two object types simultaneously based on the relation between them.

In real-world data mining applications, the target data may contain relations involving more than two types. Multi-type relational clustering approaches have been proposed to simultaneously produce clusters of each type of objects based on the relations among them [29, 30, 88–92]. These approaches could be considered as generalized co-clustering or high-level co-clustering. Multi-type relational clustering approaches in [8, 30, 31, 89, 91] are formulated as different matrix factorization problems to simultaneously factorize a set of relation matrices into smaller matrices through minimizing the total factorization error. The multi-way clustering in [29] is an extension of the information theoretic-based co-clustering proposed in [83]. Approaches in [88] and [90] are generalized graph partitioning approaches from bi-partite graph partitioning to r -partite graph partitioning. [88] solves the problem with semi-definite programming, and [90] uses iterative spectral clustering. Among those approaches, some of them such as [8, 30, 89], handle relational data represented as a r -partite graph, where homogeneous links are not considered, and some others such as [88, 91, 92], are designed only for relational data with a star-structure. In another study [92], a clustering approach called NetClus is proposed for star-structured data. NetClus is a generative model-based approach, where in each iteration, a generative model for the objects of the central-type is established based on the current rankings of attribute type objects in each cluster, which are calculated with heuristically designed ranking rules. Recently, Long et al extended their previous work to handle general multi-type relational data [31]. This approach is also a non-negative matrix factorization based approach. We will review it with more details in Chapter 5.

In all these approaches, a matrix is used to represent a relation between two types of objects, and the multi-type relational data is represented by a set of relation matrices. Different representation of multi-type relational data called relation graph is adopted in [32]. In this approach, a relation is represented by a multi-modal tensor, which may involve multiple types of objects. With the definition of relation graph, the multi-way

clustering is formulated to capture the maximal “information” in the original relation graph, i.e., to accurately approximate the set of tensors corresponding to the various relations. This approach is also a generalization of the co-clustering approach in [84].

2.4.8 Semi-supervised Clustering

For these methods discussed above, despite the difference in concepts and mathematic models in formulating the clustering problem, they are all unsupervised, i.e., clustering is performed only based on unlabeled data. This fully data-driven characteristic differentiates document clustering from document classification, the supervised document categorization. Although being fully automatic makes clustering an efficient and widely used document analysis tool, the performance of unsupervised approaches is usually worse than that of the supervised ones. The fact that reducing of human interaction tradeoffs categorization accuracy inspires the invention of semi-supervised learning to find a balance between effectiveness and cost. Unlike traditional unsupervised clustering, semi-supervised clustering makes use of a small amount of supervision or external knowledge together with a large portion of unlabeled dataset to produce clusters of the dataset [93–97]. It has been shown in the literature that even little supervision could help to improve the clustering results significantly.

Two important factors need to be concerned in a semi-supervised clustering: the representation form of the supervision and the way the supervision is used for clustering. Up to now, supervision in semi-supervised clustering is typically in the form of labels of a small subsets of data, or constraints on a number of pairs of objects. The former directly specifies which class each labeled object belongs to, while the later indicates which two objects should or should not be grouped in the same cluster. Compared with the label information, which needs to be decided requiring the categories of the dataset to be well defined, pairwise constraints can be obtained without such a requirement, and thus the later possibly has a wider application domain. Once the representation of the supervision is decided, it may be used to train a distance metric [95], or incorporated into the objective function as a regularization term [94, 96].

The way of incorporating user specified pairwise constraints into the objective function is widely used to extend those unsupervised clustering into semi-supervised

ones, such as constrained fuzzy clustering [98, 99], constrained spectral clustering [97], and constrained matrix factorization [8]. These semi-supervised clustering approaches try to produce clusters not only to optimize the clustering criterion but also to make the final partitioning consistent with user provided knowledge as much as possible.

2.4.9 Discussion

Among all these clustering methods, hierarchical clustering generates a series of nested clusters and a partitioning is obtained by cutting the dendrogram from a certain level; while other types of methods only produce a “flat cutting” of the dataset and usually require the number of clusters k to be given as a parameter. How to estimate k is another related topic called cluster validity and is out of the focus of this thesis. Therefore, throughout this study, we assume that the number of clusters k is known. In most of these approaches, an objective function is defined to measure the quality of the partitioning, and the goal is to minimize or maximize the value of the objective function, e.g., to minimize the total within cluster distance from all objects to their closets clusters, minimize the square errors or KL divergence between the original data matrix and the approximated ones, or maximize the total within cluster associations. Due to the difficulty in solving the optimization problems usually with constraints directly, most of these approaches use iterative algorithm to alternately update one variable at once while keeping others constant. In many approaches, other than the cluster indicator, which directly reflects the output of clustering, other concepts or variables may also be defined. Many approaches explicitly define a way to represent, or model each of the clusters, based on which objects are assigned. For example, in center-based approach, a centroid or medoid is used to represent the cluster based on which the closeness of each object to a cluster is then measured; in model based approaches, each cluster is represented as a parameterized distribution; and in NMF, a cluster is captured by a basis vector. In these approaches, the iterative optimization process consists of alternating update of the representations of all the clusters and the cluster assignments of all the objects based on each other. Some other approaches, such as kernel k-means, spectral clustering are formulated by only involving cluster indicators.

Approaches including partitioning clustering k-means, fuzzy clustering fuzzy c-means, and NMF, are usually applied for document clustering based on the vector-based data. Topic modeling approaches such as Latent Dirichlet Allocation (LDA) assume that each of the documents is “a bag of words”. Some others including linkage-based hierarchical clustering, medoids-based partitioning approaches, graph-based approaches, and kernel-based clustering typically work with a document-document relation matrix. All of the co-clustering approaches take the document-term co-occurrence matrix as input but treat it as a bi-type heterogeneous relational data, and cluster documents and terms simultaneously. Multi-type relational clustering approaches are able to simultaneously cluster documents and multiple other related types of objects based on relations among them.

Other than those clustering methods in machine learning, artificial intelligent methods including Neural Networks approaches such as self-organizing map (SOM) [100, 101], and evolutionary algorithms such as genetic algorithms (GAs) [102, 103], particle swarm optimization (PSO) [104], etc., have also been applied for document clustering.

2.5 Why Treat Document Data as Multi-type Relational Data? Why Fuzzy Clustering?

In many data mining applications, the document dataset to be clustered is described by several types of data. For example, a majority of documents on the Web, which is the largest data source today, are linked Webpages. Compared to text document data, which is clustered based on the content of each document, the links among Web documents provide important information that could be used for document clustering. Some approaches derive new vector-representations or document similarities by combining different types of data, and then use an existing clustering approach to produce the clusters of Web documents [26–28].

Instead of incorporating multiple types of data into features or document similarities before clustering, some approaches directly make use of different types of data during clustering. Several recently proposed model-based approaches such as [66],

[16] and [17], [105] consider both content (words) and structure (links) data when establishing the topic models. In a recent work, a regularized k-means is proposed to combine both feature and spatial context for image pattern discovery [106]. Whereas those approaches consider both words and links, treating document data as multi-type relational data is more generic and flexible as more types of data may be considered. Other than that, similar to the merits of co-clustering, simultaneous clustering of documents and other related types of entities helps to improve the quality of document clusters, and moreover, provides more descriptive information to understand and interpret each of the document clusters produced.

However, most of the existing multi-type relational data clustering approaches are formulated without considering the document-document relation, which always provides very useful information for document clustering. Furthermore, many of them do not scale well to large datasets. It arises the need to develop a new approach for document clustering which have both effectiveness and scalability. Fuzzy clustering allows overlaps between clusters, which is a favorable property for capturing the common subtopics among different topics. Relationships between fuzzy clustering and other types of clustering including model-based clustering, soft competitive learning, have been discussed in [107]. Both as “soft” k-means, fuzzy c-means clustering is closely related to Gaussian Mixture Models (GMMs) when the mixture weights are equal. However, in probabilistic model-based clustering approaches, it is assumed that the data is generated from a mixture of certain probability distributions. In order for the probabilistic clustering to be practical, restrictive assumptions about the underlying distributions have to be made. Model-based clustering may be less effective when the assumed models are not suitable for the given dataset. Differently, instead of trying to learn any models, fuzzy clustering focuses on the partitioning solution of the task. Since no model parameters need to be estimated during clustering, fuzzy clustering usually has fewer variables to calculate during the clustering process and hence more computationally efficient. This allows fuzzy clustering to have a good trade-off between its performance and computational complexity. Moreover, the objective function based formulation and alternating optimization algorithm of fuzzy clustering of object data provides a sound foundation for proposing fuzzy multi-type relational data clustering.

2.6 Evaluation

Evaluation metrics or cluster validity measures are used to access the performance of a clustering algorithm and compare the results generated by different clustering approaches. In general, there are three types of evaluation metrics namely external metrics, internal metrics, and relative metrics [108]. External metrics are defined to measure the degree of agreement between the clusters generated by a clustering algorithm and pre-specified structure or reference partitions of the dataset. Internal metrics measure the quality of clusters produced by a clustering algorithm with the degree of inter-cluster separation and intra-cluster compactness. Since internal metrics are defined only based on the dataset itself, it is usually used for exploring the underlying structures and discovering the true number of clusters. Relative metrics are usually used to compare the clustering results of the same clustering algorithm but with different parameter settings. Extensive validation methods for different types of clustering algorithms are provided in [33]. Since the datasets used in our experiments are all with labels generated by human experts, which can be used as ground-truth categories or reference partitions, we use external metrics for evaluating the clustering results in our experiments. A clustering algorithm is said to be “good” if the clusters it produced show a high agreement with human generated ones, and it is said to be “bad” if the clusters it produced seldom agree with those human generated ones.

Now we review several most widely used external evaluation metrics of clustering. We use *cluster* to refer to the result created with clustering algorithms and *class* to refer to a ground truth category. n is the total number of objects; n_c is the number of objects in the c th *cluster*; n_j is the number of objects in the j th *class*; and n_c^j is the number of documents that in both class j and cluster c . k and q are the number of *clusters* and *classes*, respectively. The precision and recall of *class* j and *cluster* c denoted as P and R are two intuitive statistic metrics defined as below

$$\begin{aligned} P(j, c) &= n_c^j / n_c, \\ R(j, c) &= n_c^j / n_j. \end{aligned} \tag{2.23}$$

From (2.23), it can be known that there are $k \times q$ sets of precisions and recalls for each pair of *class* and *cluster*. Another similar metric *Purity* given as below [109] only

considers the largest precision for each cluster. More specifically, *Purity* of cluster c denoted as $P(S_c)$ is the largest precision of those related to cluster c . The total purity is calculated as the average of purities of all the clusters:

$$Purity = \sum_{c=1}^k \frac{n_c}{n} P(S_c) = \frac{\sum_{c=1}^k \operatorname{argmax}_j(n_c^j)}{n} \quad (2.24)$$

where

$$P(S_c) = \frac{1}{n_c} \operatorname{argmax}_j(n_c^j) = \operatorname{argmax}_j P(j, c) \quad (2.25)$$

is the purity of cluster c .

Although *Purity* is simple and compact, it may not be a reasonable metric when precision and recall are unbalanced, which means an algorithm with a high *Purity* due to high precisions might be associated with low recalls. Instead of just considering precision, the metric *F-measure* [110] reflects the overall quality of the resulting clusters with the a weighted combination of precision and recall as in Eq. (2.26). Typically, precision and recall are given equal weight with $\alpha = 1$. The total *F-measure* is calculated as the average of the largest F-measure of each cluster to all the classes.

$$F\text{-measure}(j, c)_\alpha = \frac{(1 + \alpha)P(j, c) \times R(j, c)}{\alpha P(j, c) + R(j, c)} \quad (2.26)$$

$$F\text{-measure} = \sum_{j=1}^q \frac{n_j}{n} \operatorname{argmax}_c \{F\text{-measure}(j, c)\} \quad (2.27)$$

For cases when $k = q$, we may use a combinatorial optimization algorithm, e.g., the Hungarian algorithm, to establish a one-to-one match between the k clusters and the k classes, so that the total number of common objects shared by a cluster and its matched class is the largest. Once this match is found, the *Accuracy* value is defined as

$$Accuracy = \frac{1}{n} \sum_{c=1}^k n_c^p \quad (2.28)$$

where n_c^p is the number of common objects in the c th cluster and its matched class p .

Other two metrics rooted from the information theory are *entropy* [111] and *normalized mutual information* (NMI) [112, 113] defined as

$$entropy = \sum_{c=1}^k \frac{n_c}{n} E(S_c) = -\frac{1}{n \log q} \sum_{c=1}^k \sum_{j=1}^q n_c^j \cdot \log \frac{n_c^j}{n_c} \quad (2.29)$$

where

$$E(S_c) = -\frac{1}{\log q} \sum_{j=1}^q \frac{n_c^j}{n_c} \log \frac{n_c^j}{n_c} \quad (2.30)$$

represent the entropy of cluster c , and

$$NMI = \frac{\sum_{c=1}^k \sum_{j=1}^q n_c^j \log(\frac{n \cdot n_c^j}{n_c \cdot n_j})}{\sqrt{(\sum_{c=1}^k n_c \log \frac{n_c}{n})(\sum_{j=1}^q n_j \log \frac{n_j}{n})}}. \quad (2.31)$$

Entropy evaluates the distribution of documents from each reference *class* within *clusters*. For ideal cases, the number of *classes* is the same as the number of *clusters* found by algorithms and a one to one correspondence is established between *classes* and *clusters*. However, in real applications, the number of *clusters* found by a clustering algorithm might be unequal to the number of *classes*. In such kind of cases, *NMI* is a superior measure than *Entropy*. 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, *Accuracy* and *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 is based, e.g. for information-theory based clustering, *Entropy* and *NMI* might be chosen, while for clustering adopts statistic theory, *Accuracy* 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. In our experimental study, we used *F-measure*, *Accuracy*, *Entropy* and *NMI* these four metrics. We found that the same conclusion always can be made with just *Accuracy* and *NMI*. Therefore, we only report the results in terms of these two metrics. Higher *Accuracy* and *NMI* means more agreement between algorithm-generated results and ground-truth categories, and thus better performance. Both *Accuracy* and *NMI* equal to 1 only when the clusters produced by an algorithm are identical to ground-truth categories.

2.7 Summary

In this Chapter, we provided some background knowledge on document clustering including document representation, document similarity, data formats, clustering ap-

proaches and evaluation metrics. Document representation and similarity have been generally regarded as two important components in document clustering; while data formats is discussed since we think the two basic data formats, vector-based data and proximity data may not be effective enough for document datasets described by several types of data to provide information from different angles. Such kind of document datasets may be treated as multi-type relational data, where each relation corresponds to one type of data. A literature review on different types of document clustering methods is provided. Other than classical unsupervised clustering, we also discussed recently developed semi-supervised clustering. After discussing the properties of existing clustering approaches, we found that fuzzy approach for multi-type relational data clustering is a good candidate for clustering of various document datasets. Finally, we reviewed several external evaluation metrics, which are used to measure how well the clustering result is compared to the reference one.

Chapter 3

k-Partitions Clustering with Weighted Medoids for Relational Data

3.1 Overview

Center-based partitioning clustering, including (fuzzy) k-means and (fuzzy) k-medoids, plays an important role in the research and practice of the clustering technology. Compared to k-means type approaches, where virtual objects called centroids are used as cluster centers, the way of selecting real objects from the given dataset as representatives of clusters in k-medoids type approaches is shown to be more robust. Moreover, clustering only with pairwise (dis)similarities of objects in the dataset allows medoids-based approaches to be able to make use of a wide range of advanced document similarity measures. However, in real applications, using only one object to capture or interpret each cluster may not be sufficient enough, which in turn could affect the accuracy of the clustering analysis. In this chapter, we present a new fuzzy k-partitions clustering approach called *Fk*-Parts, which works with a similarity matrix or a relation matrix. In *Fk*-Parts, each cluster is characterized by multiple weighted representative objects. The prototype weight of an object with respect to a cluster describes the degree of representativeness of this object in that cluster. It is introduced to enhance the ability of capturing the underlying clusters more precisely, and gain more detailed cluster-based information, which we believe is also very important and useful in data analysis. Experimental results on 20newsgroups data and CLUTO data show that,

significant improvements are made by Fk -Parts compared with existing medoid based approaches.

3.2 Existing Medoid-based Clustering Approaches

Before presenting the detailed formulation of our approach, we first review existing medoid-based hard and fuzzy clustering approaches.

3.2.1 PAM: A k -Medoids Approach

The k -medoids approach is similar to the k -means approach since both of them produce a flat partitioning of a data set with k non-overlapped clusters to minimize the total intra-cluster distance or dissimilarity. The main difference between k -medoids and k -means is the calculation of cluster centers, which are used to represent clusters. In k -medoids, each cluster is represented by a real object selected from the given dataset, while in k -means, the center of a cluster, called centroid, is computed as the mean of objects belonging to that cluster. This difference makes k -means approach always be applied to vector-based data, but k -medoids to relational data.

Given a dataset of n objects denoted as $\mathcal{X} = \{x_i\}_{i=1,2,\dots,n}$, the objective of k -medoids is to find k representative objects called medoids, such that the total dissimilarity of other objects to their closest medoids is minimized. We use δ_c to denote the medoid of cluster c , and $\delta_c \in \mathcal{X}$. The objective function of k -medoids can be written as:

$$J_{k\text{-medoids}} = \sum_{c=1}^k \sum_{x_i \in \mathcal{C}_c} d(x_i, \delta_c) \quad (3.1)$$

where $d(x_i, \delta_c)$ denotes any type of dissimilarity between object x_i and δ_c , and $\mathcal{C}_c$ denotes the set of objects that belong to cluster c . In k -medoids, x_i is only assigned to the cluster, the medoid of which is most similar to this object, i.e.

$$x_i \in \mathcal{C}_f, \text{ where } f = \arg \min_{c \in \{1,2,\dots,k\}} d(x_i, \delta_c) \quad (3.2)$$

Since solutions of medoids are located in the sample space, which is discrete, finding those medoids typically needs to be accomplished with heuristic search algorithms.

This is different from the case of k-means, in which the centroids of each of the clusters can be updated iteratively because centroids are in a continuous space. Due to the fact that exhaustive search of medoids is an NP hard problem, Kaufman *et al.* proposed one approximate search algorithm called Partitioning Around Medoids (PAM) [1]. The main task of PAM is to find k objects as medoids. Clusters are then formed by assigning each of the non-medoid objects to their nearest medoids. Two phases are used in PAM to first initialize and then modify the medoid set. In the initialization phase called BUILD, the first initial medoid is selected as the most central object of the dataset, or the object that has the smallest sum of dissimilarities to all other objects in the dataset, i.e., $\delta_1 = x_q$, with $q = \arg \min_i \sum_j d(x_i, x_j)$. Other $k - 1$ initial medoids are selected successively to decrease the objective function as much as possible. After the initial set of the k medoids are obtained, the second phase called SWAP is used to make possible improvement of the medoids so that the clusters are improved accordingly. During the SWAP phase, a new object x_j is selected as a representative object to replace one of the initial medoids δ_c if the swap between x_j and δ_c decreases the value of the objective function most. To check whether such an object exists, swaps between δ_c and all other non-medoid objects need to be considered. The process continues until no swap between the current medoids and other objects can decrease the value of the objective function. The PAM algorithm has a time complexity of $O(kn^2)$. Different scalable extensions of PAM, such as CLARA [1] and CLARANS [114] are proposed to deal with large datasets.

It is easy to see that PAM is a hard clustering approach since each object can only belong to one of the k clusters. Next, we review an existing fuzzy k-medoids approach which produces fuzzy clusters, i.e. each object may be assigned to several clusters with various degrees of belongings.

3.2.2 FCMdd: A Fuzzy Relative of k -Medoids

A k-medoids based fuzzy clustering called FCMdd has been proposed in [115] with the application to Web mining tasks. The objective function of FCMdd is given as:

$$J_{\text{FCMdd}} = \sum_{c=1}^k \sum_{i=1}^n u_{ci}^m d(x_i, \delta_c) \quad (3.3)$$

where u_{ci} is the fuzzy membership subject to Eq. (2.11), and $d(x_i, \delta_c)$ is defined in the same way as in Eq. (3.1). In fact, FCMdd is formulated in the same way as that of fuzzy c-means (FCM) in Eq. (2.12) except that δ_c in Eq. (3.3) is defined as the medoid (one of the real object in the sample space) not the centroid (a point in a continues space). Compared to k-medoids, where solutions of cluster assignments and medoids are both discrete, the solutions of cluster assignments in FCMdd now are continues variables. Given a current set of medoids, u_{ci} in FCMdd can be updated using membership update formulas of several existing methods, such as the one in Eq. (2.14).

However, like in hard k-medoids, the k medoids in FCMdd still need to be found with search algorithms due to the discrete nature of the solution. Unlike in PAM, where each medoid is selected as the object that minimizes the value of the objective function most, a simpler way is used in FCMdd for medoid selection. Given the current memberships, each medoid is decided using a simple heuristic algorithm as: object x_q is selected as the medoid of cluster c , i.e. $\delta_c = x_q$, if

$$q = \arg \min_{j \in \{1, 2, \dots, n\}} \sum_{i=1}^n u_{ci}^m d(x_i, x_j) \quad (3.4)$$

This way used in FCMdd to search for the medoids is simpler than that in PAM, hence makes FCMdd run faster in real time. However, the efficiency of a simplified heuristic search has the tradeoff in effectiveness. Our experimental study show that FCMdd is much easier to be trapped into local minimums than PAM.

3.2.3 Limitation of the One-Medoid-One-Cluster Strategy

In existing hard and fuzzy k-medoids clustering as reviewed, each cluster is represented by one medoid, which is the most central object in the cluster. However, the way of using one object to describe a cluster may not be sufficient enough in some cases.

Now we use a simple 2-D dataset as shown in Figure 3.1 to illustrate the limitation of the one-medoid for one-cluster formulation. The dataset contains two well separated clusters, of which the cluster on the left side referred as $\mathcal{C}_1$ consists of four objects, and the cluster on the right side referred as $\mathcal{C}_2$ consists of six objects. It can be seen that

Figure 3.1: A dataset with two clusters

in $\mathcal{C}_1$, it is insufficient to describe the cluster structure by choosing only one of the four objects, since no one of the four objects alone could be viewed as a more proper representative than the other three in that cluster. In $\mathcal{C}_2$, two objects out of the six are almost equal reasonable to be selected as the representative of the cluster. This means choosing one of the two, although may not distort the representation of this cluster much, may fail to discover the complete set of all the candidate representative objects. From this example, we see that in order to capture more detailed information of clusters, we may need more objects rather than one to be referred as medoids or representative objects of a cluster. Since the degree of representativeness varies from object to object, it is better to attach a certain weight to each object with respect to a cluster to show how much this object represents a cluster. Motivated by this idea, we now propose our new fuzzy clustering approach for similarity-based data analysis.

3.3 The Proposed Fk-Parts for Relational Data Clustering

3.3.1 Formulation

For a dataset of n objects denoted as $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$, partition the dataset into k fuzzy clusters based on the similarities between each pair of objects that may be formally expressed as below:

Definition 3.2 (Problem) *Given relation matrix $\mathbf{R}_{n \times n}$, where $r_{ij} \in \mathbf{R}$ is the similarity or association between x_i and x_j , and the number of clusters k , the current problem is to find membership matrix $\mathbf{U}_{n \times k}$, where $u_{ic} \in \mathbf{U}$ is the fuzzy membership of x_i with respect to cluster c so that the quality of the produced clusters in some sense is optimal.*

To present the new approach for solving the above problem, we first transform the objective function of dissimilarity-based k-medoids into a similarity-based one. We define

$$u_{ic} = \begin{cases} 1, & \text{if } x_i \in \mathcal{C}_c \\ 0, & \text{otherwise} \end{cases} \quad (3.5)$$

and

$$v_{jc} = \begin{cases} 1, & \text{if } x_j = \delta_c \\ 0, & \text{otherwise} \end{cases} \quad (3.6)$$

where $\mathbf{U}$ records whether an object belongs to a cluster, and $\mathbf{V}$ records which object is the medoid of a cluster. With the above definition, the objective function of k-medoids in Eq. (3.1) can be rewritten into a function of $\mathbf{U}$ and $\mathbf{V}$ as below

$$J_{k\text{-medoids}} = \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n u_{ic} v_{jc} d_{ij} \quad (3.7)$$

where $d_{ij} \in \mathbf{D}$ is the dissimilarity between x_i and x_j .

The above k-medoids formulation is to find u and v to minimize the total within cluster dissimilarity. Since in document clustering, the pairwise relational data is usually known as similarity $\mathbf{R}$ rather than dissimilarity $\mathbf{D}$, we give the similarity-based objective function of k-medoids as below:

$$J_{k\text{-medoids}'} = \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n u_{ic} v_{jc} r_{ij} \quad (3.8)$$

The equivalence between minimization of Eq. (3.7) and maximization of Eq. (3.8) is stated as theorem 3.1 followed by the proof.

Theorem 3.1 *With u and v defined in Eq. (3.5) and Eq. (3.6), maximization of Eq. (3.8) is equivalent to minimization of Eq. (3.7) when $r_{ij} = d_{\max} - d_{ij}$, where $d_{\max} = \max_{i,j} d_{ij}$ is the largest entry of $\mathbf{D}$.*

Proof: Substituting $r_{ij} = d_{\max} - d_{ij}$ into Eq. (3.8) gives

$$\begin{aligned} J_{k\text{-medoids}'} &= \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n u_{ic} v_{jc} (d_{\max} - d_{ij}) \\ &= - \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n u_{ic} v_{jc} d_{ij} - d_{\max} \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n u_{ic} v_{jc} \end{aligned} \quad (3.9)$$

according to Eq. (3.5) and Eq. (3.6), we have

$$\sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n u_{ic} v_{jc} = n \quad (3.10)$$

thus the second term of Eq. (3.9) is a constant that could be ignored, and this immediately ends the proof. In fact, any constant that larger than d_{max} also gives the same result. Theorem 3.1 can also be stated as maximizing the objective function of $J_{k-medoids'}$ in Eq. (3.8) is equivalent to minimizing Eq. (3.8) with the dissimilarity $d_{ij} = r_{max} - r_{ij}$, where r_{max} can be any constant that larger than $\max_{ij} r_{ij}$.

In Eq. (3.7) and Eq. (3.8), the values of u_{ic} and v_{jc} are binary, i.e., each object only belongs to one of the k clusters and each cluster only has one of the n objects to be selected as the medoid or representative object. As reviewed in Chapter 2.4.3, a typical way used to obtain fuzzy memberships is by introducing an exponent m of u_{ic} . By using this parameter, called fuzzifier, together with the membership constraints, the cluster assignment becomes fuzzy rather than crisp or hard. The fuzziness of memberships can also be generated by incorporating an additional term called regularization term or penalty term. Compared with the way of introducing an exponent m , using individual regularization terms to the objective function corresponds to a more clear physical meaning. Since the entropy regularization may cause overflow problem in computing for large scaled data, here as in [85] and [50], we use the quadratic term as the regularization in the objective function.

As illustrated above, the conventional way of selecting k representative objects with each to totally represent one cluster may be insufficient to fully characterize a cluster with a complex structure. This in turn indicates that multiple objects may be used in order to capture each cluster more accurately. Making it more general, each object in a cluster can be referred as the representative or medoid of that cluster to some extent. The prototype weight reflects the degree of representativeness of each object in a cluster is then defined and treated as a generalized form of the one-medoid representation as in Eq. (3.6). We apply the quadratic regulation on both u and v to get the fuzzy membership as well as the prototype weight, respectively. The proposed method Fk -Parts is now formulated as follows:

$$\begin{aligned}
 J_{\text{Fk-Parts}} = & \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n u_{ic} v_{jc} r_{ij} - \frac{\phi}{2} \sum_{c=1}^k \sum_{i=1}^n u_{ic}^2 \\
 & - \frac{\eta}{2} \sum_{c=1}^k \sum_{j=1}^n v_{jc}^2
 \end{aligned} \tag{3.11}$$

with U is subject to constraints the same as in Eq. (2.11) and V is subject to:

$$\forall c, \quad \sum_{j=1}^n v_{jc} = 1; \tag{3.12}$$

$$\forall c, j, \quad v_{jc} \geq 0. \tag{3.13}$$

where u_{ic} denotes the fuzzy membership of x_i in cluster c , and v_{jc} denotes the prototype weight of x_j in cluster c . The fuzzy membership allows each object to belong to several clusters with various degrees, and the representative weight or prototype weight allows each cluster to be represented by multiple objects, which are weighted based on their importance in a cluster. The larger u_{ic} is, the more possible that x_i belongs to cluster c . The larger v_{jc} is, the better that x_j represents cluster c . Matrices $\mathbf{U}_{n \times k}$ and $\mathbf{V}_{n \times k}$ record the memberships and prototype weights of all the n objects with respect to all the k clusters, respectively. From the definition, $\mathbf{U}$ corresponds to the fuzzy partitioning of the dataset and $\mathbf{V}$ is used to describe and represent each cluster. From $\mathbf{V}$, a more detailed description of the internal structure of each cluster can be found with cluster-based information, such as the ranks of objects in each cluster.

The objective function in Eq. (3.11) consists of three terms. The first term is the total within cluster dissimilarity that we want to minimize. The last two terms are the quadratic terms of u and v . Two parameters ϕ (> 0) and η (> 0) control the tradeoff between the first term and the two regularization terms. The larger values ϕ and η are set, the more the regularization terms contribute in the optimization process, and the “smoother” u and v are distributed. Here u is said to be “smoother” or fuzzier when the u values of objects in different clusters tend to be evenly assigned, and v becomes “smoother” when the v values of all objects in each cluster tend to be evenly distributed. In order to let $\mathbf{U}$ and $\mathbf{V}$ play different roles in clustering, i.e., one for assignment and the other for representation of the k clusters, we pose constraints on

$\mathbf{U}$ and $\mathbf{V}$ in two different ways as given in Eq. (2.11) and Eq. (3.12), respectively. The constraints on $\mathbf{U}$ is with respect to each object and on $\mathbf{V}$ is with respect to each cluster.

3.3.2 Solution

The clustering problem of finding optimal cluster assignments of objects and representatives of clusters is now formulated as a constrained optimization problem, i.e., to find the optimal values of u and v to maximize Eq. (3.11) subject to a set of constraints. Since U and V are continuous variables, we can use the method of Lagrange multipliers with the first order necessary condition to derive the solutions of them. The Lagrangian function is first constructed as

$$L_{\text{Fk-Parts}} = J_{\text{Fk-Parts}} + \sum_{i=1}^n \lambda_i \left(\sum_{c=1}^k u_{ic} - 1 \right) + \sum_{c=1}^k \beta_c \left(\sum_{j=1}^n v_{jc} - 1 \right) \quad (3.14)$$

where λ_i and β_c for all i and c are the Lagrange Multipliers. According to $\frac{\partial L_{\text{Fk-Parts}}}{\partial u_{ic}} = 0$, we have

$$\sum_{j=1}^n v_{jc} s_{ij} - \phi u_{ic} + \lambda_i = 0 \quad (3.15)$$

which gives

$$u_{ic} = \frac{1}{\phi} \left(\sum_{j=1}^n v_{jc} r_{ij} + \lambda_i \right) \quad (3.16)$$

Substituting the above equation into the constraints in Eq. (2.11), we have

$$\sum_{c=1}^k \frac{1}{\phi} \left(\sum_{j=1}^n v_{jc} r_{ij} + \lambda_i \right) = 1 \quad (3.17)$$

which requires

$$\lambda_i = \frac{1}{k} \sum_{c=1}^k \sum_{j=1}^n v_{jc} r_{ij} + \frac{\phi}{k} \quad (3.18)$$

substitute the above equation back into Eq. (3.16), we get the update equations of u_{ic} as below

$$u_{ic} = \frac{1}{k} + \frac{1}{\phi} \left[\sum_{j=1}^n v_{jc} r_{ij} - \frac{1}{k} \sum_{f=1}^k \sum_{j=1}^n v_{jf} r_{ij} \right], \quad (3.19)$$

Similarly, according to $\frac{\partial L_{\text{FK-Parts}}}{\partial v_{jc}} = 0$ and $\frac{\partial L_{\text{FK-Parts}}}{\partial \beta_c} = 0$ the update equation of v_{jc} is derived as below

$$v_{jc} = \frac{1}{n} + \frac{1}{\eta} \left[\sum_{i=1}^n u_{ic} r_{ij} - \frac{1}{n} \sum_{q=1}^n \sum_{i=1}^n u_{ic} r_{iq} \right]. \quad (3.20)$$

It is easy to see that when $\phi \rightarrow \infty$,

$$\begin{aligned} u_{ic} &= \frac{1}{k}, \\ v_{jc} &= \frac{1}{n} + \frac{1}{\eta * k} \left[\sum_{i=1}^n r_{ij} - \frac{1}{n} \sum_{q=1}^n \sum_{i=1}^n r_{iq} \right] \end{aligned} \quad (3.21)$$

and when $\eta \rightarrow \infty$

$$u_{ic} = \frac{1}{k}, \quad v_{jc} = \frac{1}{n}. \quad (3.22)$$

This indicates that when ϕ is set to be large enough, each object tends to have equal degrees of belonging to all the clusters, while all objects tend to be equally weighted as representatives in each cluster when η is set to be large enough. This is because a very large ϕ makes the second term $\frac{\phi}{2} \sum_{c=1}^k \sum_{i=1}^n u_{ic}^2$ dominate in the optimization. In this case, the optimal solution of $J_{\text{FK-Parts}}$ becomes the one that minimizes this term. Since for $i = \{1, 2, \dots, n\}$, the value of $\sum_{c=1}^k u_{ic}^2$ is minimized with all $u_{ic} = 1/k$, the second term is minimized with this solution. Similarly, when η is very large, the third term dominates the optimization, which leads the solution to minimize $J_{\text{FK-Parts}}$ is found by minimizing the third term. For $c = \{1, 2, \dots, n\}$, $\sum_{j=1}^n v_{jc}^2$ is minimized when all $v_{jc} = 1/n$, with which the third term is minimized. Obviously, these two extreme results are useless for data analysis. The quadratic terms are introduced to regularize $\mathbf{U}$ and $\mathbf{V}$ the first term or the original criterium that needs to be optimized from generating singular solutions. However, the regularization part can not be over emphasized to make it surpass the first term and dominate the optimization. In order to get information that closely captures the underlying data structure, ϕ and η should not be set too large.

From Eq. (3.19) and Eq. (3.20), it can be seen that negative values of u and v might occur during the successive optimization process. To ensure the non-negativity of u and v , we need to explicitly consider the non-negative constraints when constructing the Lagrangian:

$$L'_{\text{Fk-Parts}} = L_{\text{Fk-Parts}} + \sum_{c=1}^k \sum_{i=1}^n \psi_{ic} u_{ic} + \sum_{c=1}^k \sum_{j=1}^n \gamma_{jc} v_{jc} \quad (3.23)$$

where ψ_{ic} and γ_{jc} are Lagrange multipliers. For solving u_{ic} , the Karush-Kuhn-Tucker conditions to be satisfied are:

$$\frac{\partial L'_{\text{Fk-Parts}}}{\partial u_{ic}} = 0 \quad (3.24)$$

$$\psi_{ic} \geq 0 \quad (3.25)$$

$$\psi_{ic} u_{ic} = 0 \quad (3.26)$$

Now we give the detailed derivation of the update equations of U and V from the Lagrangian in Eq. (3.23) and Karush-Kuhn-Tucker conditions in Eqs. (3.24) to Eq. (3.26). Combining Eq. (3.23) and Eq. (3.24) gives

$$u_{ic} = \frac{1}{k} + \frac{1}{\phi} \left[(\alpha_{ic} + \psi_{ic}) - \frac{1}{k} \sum_{f=1}^k (\alpha_{if} + \psi_{if}) \right], \quad (3.27)$$

where

$$\alpha_{ic} = \sum_{j=1}^n v_{jc} r_{ij} \quad (3.28)$$

Now we consider the condition (3.25) from two cases, respectively.

(1) $\psi_{ic} = 0, \forall c \in \{1, 2, \dots, k\}$, so Eq. (3.27) becomes

$$u_{ic} = \frac{1}{k} + \frac{1}{\phi} \left[\alpha_{ic} - \frac{1}{k} \sum_{f=1}^k \alpha_{if} \right], \quad (3.29)$$

which is the same as Eq. (4.31). For each object, this set of solution is only valid if

$$\frac{1}{k} + \frac{1}{\phi} \left[\alpha_{ic} - \frac{1}{k} \sum_{f=1}^k \alpha_{if} \right] \geq 0 \quad \text{for } c = 1, 2, \dots, k \quad (3.30)$$

otherwise, we have to consider the second case.

(2) $\psi_{ic} > 0$ for at least one c . According to (3.26), when $\psi_{ic} > 0$, $u_{ic} = 0$, therefore

$$0 = \frac{1}{k} + \frac{1}{\phi} \left[(\alpha_{ic} + \psi_{ic}) - \frac{1}{k} \sum_{f=1}^k (\alpha_{if} + \psi_{if}) \right] \quad (3.31)$$

From the above equation, it can be obtained that

$$\alpha_{ic} + \psi_{ic} = \frac{1}{k} \left[\sum_{f=1}^k (\alpha_{if} + \psi_{if}) - \phi \right] \quad (3.32)$$

Constrained by Eq. (2.11), it is known that not all u_{ic} can be 0, which means for some c , $u_{ic} > 0$. So the set of clusters is split into two subsets denoted as k^- and k^+ , with

$$\begin{aligned} k^- &= \{c : u_{ic} = 0\} \\ k^+ &= \{c : u_{ic} > 0\} \neq \emptyset \end{aligned} \quad (3.33)$$

From the right hand side of Eq. (3.32), it shows that the value of $(\alpha_{ic} + \psi_{ic})$ is independent on c , which means $\forall c \in k^-$, these values are the same. Therefore (3.32) can further be written as

$$\begin{aligned} \alpha_{ic} + \psi_{ic} &= \frac{1}{k} \left[\sum_{f \in k^+} \alpha_{if} + \sum_{f \in k^-} (\alpha_{if} + \psi_{if}) - \phi \right] \\ &= \frac{1}{k} \left[\sum_{f \in k^+} \alpha_{if} + |k^-|(\alpha_{ic} + \psi_{ic}) - \phi \right] \\ \Rightarrow \alpha_{ic} + \psi_{ic} &= \frac{1}{|k^+|} \left(\sum_{f \in k^+} \alpha_{if} - \phi \right) \quad \text{for } c \in k^- \end{aligned} \quad (3.34)$$

With the above relation, for $c \in k^+$, Eq. (3.27) is rewritten as

$$\begin{aligned} u_{ic} &= \frac{1}{k} + \frac{1}{\phi} \alpha_{ic} - \frac{1}{\phi} \frac{1}{k} \left(\sum_{f \in k^+} \alpha_{if} + \sum_{f \in k^-} (\alpha_{if} + \psi_{if}) \right) \\ &= \frac{1}{k} + \frac{1}{\phi} \alpha_{ic} - \frac{1}{\phi} \frac{1}{k} \sum_{f \in k^+} \alpha_{if} - \frac{1}{\phi} \frac{1}{k} \frac{|k^-|}{|k^+|} \left(\sum_{f \in k^+} \alpha_{if} - \phi \right) \end{aligned} \quad (3.35)$$

After some transformation and with the relation $\frac{1}{k} \left(1 + \frac{|k^-|}{|k^+|}\right) = \frac{1}{|k^+|}$, we finally obtain the final form as below

$$u_{ic} = \begin{cases} 0, & \text{for } c \in k^- \\ \frac{1}{|k^+|} + \frac{1}{\phi} \left(a_{ic} - \frac{1}{|k^+|} \sum_{f \in k^+} a_{if} \right), & \text{for } c \in k^+ \end{cases} \quad (3.36)$$

where

$$a_{ic} = \sum_{j=1}^n v_{jc} r_{ij} \quad (3.37)$$

and

$$\begin{aligned} k^- &= \{c : u_{ic} = 0\}, \\ k^+ &= \{c : u_{ic} > 0\}. \end{aligned} \quad (3.38)$$

To a specific object x_i , k^+ represents a set of clusters where x_i belongs to or has positive values of memberships, and k^- represents a set of clusters this object does not belong to. We use $|k^+|$ and $|k^-|$ to denote the size of set k^+ and k^- , respectively. Based on the discussion in [50], we can decide k^+ and k^- for each object x_i with procedure-K given in the next section.

Follow the similar derivation, the update equation of v_{jc} for each cluster $c = \{1, 2, \dots, k\}$ can be obtained as

$$v_{jc} = \begin{cases} 0, & \text{for } j \in n^- \\ \frac{1}{|n^+|} + \frac{1}{\eta} \left(b_{jc} - \frac{1}{|n^+|} \sum_{q \in n^+} b_{qc} \right), & \text{for } j \in n^+ \end{cases} \quad (3.39)$$

where

$$b_{jc} = \sum_{i=1}^n u_{ic} r_{ij} \quad (3.40)$$

and

$$\begin{aligned} n^- &= \{j : v_{jc} = 0\} \\ n^+ &= \{j : v_{jc} > 0\} \end{aligned} \quad (3.41)$$

In a specific cluster c , the object set is split in to n^+ and n^- according to whether an object has a positive value of v , and $|n^+|$, $|n^-|$ are the size of n^+ and n^- , respectively.

For a cluster c , n^+ and n^- can be decided in a similar way as k^+ and k^- . Next, we give the details of the proposed algorithm.

3.3.3 Algorithm

The Fk-Parts algorithm as well as the procedure-K and procedure-N are now given in Algorithm 2 to Algorithm 4, respectively.

Algorithm 2: Fk-Parts

input : Similarity matrix $\mathbf{R}_{n \times n}$, the number of clusters k , parameters m and n or ϕ and η .
output: Fuzzy memberships $\mathbf{U}_{k \times n} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)$ and prototype weights $\mathbf{V}_{n \times k} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k)$.
Initialize membership matrix $\mathbf{U}^{(0)}$, and set iteration counter $l \leftarrow 0$;
repeat
 for each $c = \{1, 2, \dots, k\}$ **do**
 for each $j = \{1, 2, \dots, n\}$ **do**
 calculate b_{jc} with Eq. (3.40);
 decide n^+ and n^- using Procedure-N;
 update v_{jc} with Eq. (3.39).
 end
 end
 for each $i = \{1, 2, \dots, n\}$ **do**
 for each $c = \{1, 2, \dots, k\}$ **do**
 calculate a_{ic} with Eq. (3.37);
 decide k^+ and k^- using Procedure-K;
 update u_{ic} with Eq. (3.36).
 end
 end
 $l \leftarrow l + 1$;
until converge or reaching the maximum number of iterations;

As traditional center-based clustering, the iterative algorithm of Fk-Parts starts with a non-negative initialization of the membership matrix. Different initialization schemes may be used in Fk-Parts. A random initialization is a simple but usually not an effective one. During the the optimization process, U and V are updated alternately based the most latest value of the other one. The iteration continues until successive estimations of U is close enough or the maximum number of iterations is

Algorithm 3: Procedure-K

input : $\mathbf{u}$
output: k^+, k^-
 Initialize $k_0^+ = \emptyset, k_0^- = \{1, 2, \dots, n\}, h=0$;
while $u_{fi} > 0$ computed by Eq. (3.39), where $f = \arg \max_{c \in k_h^+} \{a_{ic}\}$. **do**
 | $h \leftarrow h + 1, k_h^+ = k_{h-1}^+ + \{p\}, k_h^- = k_{h-1}^- - \{p\}$; where $p = \arg \min_{j \in k_{h-1}^-} \{a_{ic}\}$;
end
 set $k^+ = k_{h-1}^+, k^- = k_{h-1}^-$;

Algorithm 4: Procedure-N

input : $\mathbf{v}$
output: n^+, n^-
 Initialize $n_0^+ = \emptyset, n_0^- = \{1, 2, \dots, n\}, h=0$;
while $v_{cg} > 0$ computed by Eq. (3.39), where $g = \arg \max_{j \in n_h^+} \{b_{jc}\}$. **do**
 | $h \leftarrow h + 1, n_h^+ = n_{h-1}^+ + \{q\}, n_h^- = n_{h-1}^- - \{q\}$; where $q = \arg \min_{j \in n_{h-1}^-} \{b_{jc}\}$;
end
 set $n^+ = n_{h-1}^+, n^- = n_{h-1}^-$;

reached. The alternating updating of U and V corresponds to successively improving the quality of clusters with reassignment of objects to clusters and re-selection of cluster medoids. At convergence, we obtain the fuzzy partitions with U as well as the soft weighted medoids for each generated cluster based on V . This cluster-based information would be very useful for users to gain more detailed descriptions on the internal structure of each individual cluster.

Next we show the algorithm of Fk -Parts will converge to a local maximum after finite iterations. We first prove that

Theorem 3.2 *The value of the objective function $J_{Fk-Parts}$ never decreases during the alternating updating of U and V , i.e. $J_{Fk-Parts}(U^{(l)}, V^{(l+1)}) \geq J_{Fk-Parts}(U^{(l)}, V^{(l)})$ and $J_{Fk-Parts}(U^{(l+1)}, V^{(l)}) \geq J_{Fk-Parts}(U^{(l)}, V^{(l)})$.*

Proof:

$$\begin{aligned}\nabla^2 L_{\text{Fk-Parts}}(U) &= \begin{bmatrix} \frac{\partial^2 L_{\text{Fk-Parts}}(U)}{\partial u_{11} \partial u_{11}} & \dots & \frac{\partial^2 L_{\text{Fk-Parts}}(U)}{\partial u_{11} \partial u_{kn}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 L_{\text{Fk-Parts}}(U)}{\partial u_{kn} \partial u_{11}} & \dots & \frac{\partial^2 L_{\text{Fk-Parts}}(U)}{\partial u_{kn} \partial u_{kn}} \end{bmatrix} \\ &= \begin{bmatrix} -\phi & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & -\phi \end{bmatrix}\end{aligned}\quad (3.42)$$

We always set $\phi, \eta > 0$, thus

$$\mathbf{x}^T \nabla^2 L_{\text{Fk-Parts}}(U) \mathbf{x} < 0, \quad \text{for any } \mathbf{x} \neq \mathbf{0} \quad (3.43)$$

which means $\nabla^2 L_{\text{Fk-Parts}}(U)$ is negative define. According to the second order conditions for constrained optimization, that for fixed V , the U computed by Eq.(3.36) is a local maximum of $J_{\text{Fk-Parts}}$ subject to constraints in Eq. (2.11). Therefore, $J_{\text{Fk-Parts}}(U^{(l)}, V^{(l+1)}) \geq J_{\text{Fk-Parts}}(U^{(l)}, V^{(l)})$. Similarly, it can be shown that for a fixed V , $J_{\text{Fk-Parts}}(U^{(l+1)}, V^{(l)}) \geq J_{\text{Fk-Parts}}(U^{(l)}, V^{(l)})$. It follows that the value of $J_{\text{Fk-Parts}}$ never decreases during the alternating updating of U and V .

According to **Theorem 3.2** and the fact that $J_{\text{Fk-Parts}} < \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n u_{ci} v_{jc} r_{ij} < k \sum_{i=1}^n \sum_{j=1}^n r_{ij}$, it yields that the algorithm of Fk-Parts will converge after a finite number of iterations.

For a dataset with n objects, the time complexity of Fk-Parts is $O(n^2)$ for one iteration, which is at the same level of complexity as that of kernal k-means, but lower than that of hierarchical clustering, which is typically $O(n^2 \log n)$, and eigendecomposition-based spectral clustering, which is $O(n^3)$. In vector space clustering, like (fuzzy) k-means, the distances between objects and cluster centers need to be recalculated in every iteration. This increases the computation when the dimension of the data grows. However, in relational data clustering, the relation matrix is always calculated in advance and it does not need to be updated during the optimization process. To speed up the computation as well as reduce the storage requirement, the relation matrix can be sparsified by setting small entries to zero. For very large relational data, sampling technique, such as the framework reported in [116] may be adopted and integrated into Fk-Parts to enhance its overall scalability.

As discussed early, regularization weights ϕ and η in Fk -Parts should be set properly to get reasonable results on different datasets. In practice, such kind of parameters always need to be tuned empirically due to the difficulty in developing generic parameter estimation approaches theoretically. Based on our experimental study, reasonable results are always obtained when ϕ and η are set to make ϕ/k and η/n close to each other or be in the same order. This is because the summation constraint on u is over k clusters while on v is over n objects, and the value of n is typically much larger than k . Therefore, the final clustering result is always more sensitive to ϕ than η when $n \gg k$. In such kind of cases, smaller steps should be used to tune ϕ than that of η .

3.4 Experimental Results

In this section, we provide experimental studies on the proposed Fk -Parts. We compare the performance of Fk -Parts with existing related approaches by several experiments. We first use an artificial 2-D data to show the superiority of weighted-medoids to single-medoid; then we study the difference roles u and v play in Fk -Parts with another 2-D dataset; after that, we compare the performance of Fk -Parts with other approaches in the task of grouping twelve countries into three categories based on their pairwise similarities directly given by human; finally, we test the performance of Fk -Parts for document categorization on eight real-world document datasets.

3.4.1 Weighted Medoids vs. Single Medoid

Before giving the experimental study of the proposed Fk -Parts on real-world datasets, we first test the performance of Fk -Parts on an artificial 2-D dataset X_{11} , which is shown in Figure 3.2. We want to compare the detailed clustering results of Fk -Parts with the existing fuzzy k-medoids approach (FCMdd) [115] to see whether the weighted medoids strategy helps Fk -Parts to achieve better clustering result than single-medoid based FCMdd.

The coordinates of X_{11} are included in Table 3.1. It can be seen that the X_{11} data is similar to the data shown in Figure 3.1 except including one more object x_{11} , which locates slightly biased to the cluster on the right hand side. The dissimilarity

Figure 3.2: Scatter plot of the X_{11} data. The red circle “o” represents the ideal center of a cluster.

Table 3.1: Results of FCMdd and Fk-Parts for the X_{11} dataset

ID	Coordinates		FCMdd ($m=1.9$)				Fk-Parts ($\phi=2.5, \eta=3.0$)				
	x	y	u_1	u_2	label	meoid	u_1	u_2	label	v_1	v_2
1	-1.20	1.00	0.8681	0.1319	1	-	0.8798	0.1202	1	0.1687	0.0000
2	-1.00	1.20	0.8974	0.1026	1	-	0.8484	0.1516	1	0.2471	0.0000
3	-1.00	0.80	0.8972	0.1028	1	-	0.8484	0.1516	1	0.2471	0.0000
4	-0.80	1.00	1.0000	0.0000	1	*	0.8134	0.1866	1	0.3370	0.0000
5	0.80	1.00	0.0816	0.9184	2	-	0.1838	0.8162	2	0.0000	0.1904
6	1.00	1.20	0.0880	0.9120	2	-	0.1478	0.8522	2	0.0000	0.1142
7	0.98	0.98	0.0000	1.0000	2	*	0.1309	0.8691	2	0.0000	0.2731
8	1.02	1.02	0.0207	0.9793	2	-	0.1242	0.8758	2	0.0000	0.2591
9	1.00	0.80	0.0719	0.9281	2	-	0.1477	0.8523	2	0.0000	0.1142
10	1.20	1.00	0.0796	0.9204	2	-	0.1154	0.8846	2	0.0000	0.0491
11	0.05	1.00	0.5250	0.4750	1	-	0.4801	0.5199	2	0.0000	0.0000

matrix is calculated with Euclidean distance, and is scaled by $d_{ij} = d_{ij} / \max_{i,j} \{d_{ij}\}$. We set $k = 2$. For FCMdd, $m = 2$ is used, and for Fk-Parts, $\phi = 1$, $\eta = 1$ are used. The results of FCMdd and Fk-Parts are shown in Table 3.1. The medoids found by FCMdd are denoted with “*”.

From Table 3.1, it can be seen that generally, the partitioning results obtained by two algorithms are similar, which indicates that x_1 to x_4 are in cluster 1 with object x_4 the most representative object, and object x_5 to x_{10} are in cluster 2 with x_7 the most representative object. However, when take a close look, it can be found that in FCMdd, the u values of medoids in the cluster they represent are always one, e.g. $u_{1,4} = u_{2,7} = 1$. This is not happen in Fk-Parts, since the objects with the largest v values are not necessary the same objects that have the largest u values, e.g. in cluster 1, x_4 has the largest v but its u value is the smallest among objects in that

cluster. The reason that in FCMdd, each medoid always has $u = 1$ in the cluster it represents is because the distance between a medoid to itself is 0! Therefore, according to Eq. (2.14), which is used in FCMdd for membership update, the fuzzy membership of the medoid in the cluster it belongs is always 1. In other words, in FCMdd, those representative objects are not allowed to belong to other clusters at all. In Fk -Parts, for each cluster, no object alone is served as the single medoid, but combines together with other objects with different weights to characterize the cluster. Therefore, each individual representative object still can belong to other clusters to a certain extent.

It is also observed the difference of two approaches in labeling x_{11} . Although x_{11} is closer to the center of cluster 2, in FCMdd, the crisp assignment of x_{11} is cluster 1. This is because the distance from x_{11} to the medoid of cluster 1, which is x_4 , is smaller than that to the medoid of cluster 2, which is x_7 , and this results in the u of x_{11} in cluster 1 is larger than that in the other cluster. This result illustrates the limitation of using a single object to characterize a cluster, especially in the cases where there is no object which is near the ideal center of the cluster that can be selected. Nevertheless, this situation has been handled to a large extent in Fk -Parts, which assigns a larger u to x_{11} in cluster 2. This demonstrates that using multiple objects with various weights is able to characterize the structure of a cluster more precisely than using only one object, and this in turn helps to produce partitions with better qualities.

3.4.2 Fuzzy Membership vs. Prototype Weight

In this experiment, the distributions of u and v are visualized through the contour plots to show the difference between u and v in describing data structures. The dataset used in this experiment is plot in Figure 3.3a. It is created by randomly generating two groups of 300 data points with Gaussian distributions of the same variance 0.1, but one is centered at (0, 1) while the other at (1.2, 0.7). The dissimilarity data is obtained with Euclidean distance. Parameters are set as $k = 2$, $m = 2$ for FCMdd, $\phi = 0.3$, $\eta = 200$ for Fk -Parts. Truncated clusters are obtained by assigning each of the objects to the cluster associated with the largest u values.

Figure 3.3b and Figure 3.3c respectively, show the contour plot of u in FCMdd and v in Fk -Parts of objects with respect to the cluster they belong to. It can be seen in

Figure 3.3b that with respect to the ideal center of a cluster, contours related to that cluster are asymmetric. This means that for objects in a cluster which are equally distant to the cluster center, the closer of an object to the other cluster, the smaller value of u is assigned to this object. This indicates that the u value of each object in a cluster is influenced by other clusters. Therefore, u may not be a proper or accurate measurement to discover the internal structure of individual cluster, although it may be plausible to obtain a partitioning of the data by labeling each of the objects based on u . From Figure 3.3c, it can be seen that all contours related to each cluster are concentric circles with the center located very near the ideal center of the cluster. This indicates that each object in a cluster is weighted only based on its closeness to the center of that cluster, i.e. the closer an object is to the ideal center, the larger value of v is assigned to this object. With respect to this, prototype weight is a good description of the structure within each cluster.

3.4.3 Country Categorization

Now, we test Fk -Parts on the Countries Data [1]. The task is to group twelve countries into clusters based on pairwise dissimilarities as given in Table 3.2. Different from dissimilarity data calculated from the object data with a distance metric, the Countries Data in Table 3.2 is provided subjectively by students in a political science class. Generally, these twelve countries belong to three categories: Western, Developing and Communist. Fk -Parts is compared with two k -medoids based approaches namely the hard clustering PAM and fuzzy clustering FCMdd. The number of clusters is set to $k = 3$, and other parameters are set as $m = 1.5$ for FCMdd, $\phi = 0.5$, $\eta = 3$ for Fk -Parts. We test the performances of the two fuzzy approaches with two different sets of initial representative countries which are

$$\begin{aligned}\Delta_1^0 &= \{USSR, Israel, India\}, \\ \Delta_2^0 &= \{France, Cuba, Belgium\}.\end{aligned}\tag{3.44}$$

The three countries in Δ_1^0 are well separated while in Δ_2^0 , *Belgium* is similar to *France*, which makes two initial medoids out of three are not well separated.

The results of FCMdd and Fk -Parts are given in Table 3.3 and Table 3.4, respectively. To make comparison, the results of PAM reported in [1] are also listed in Table

3.4. The final medoids found by FCMdd and PAM are labeled with “*”. The crisp labels of all the countries are also included in these tables by assigning each country to the cluster, where it has the largest membership.

From Table 3.3, it can be seen that FCMdd is very sensitive to initializations. When Δ_1^0 is used, two of the three final representative countries, are the same as the initial two representatives. In other words, only one of the medoids has been updated before reaching a local minimum. Things become worse when the initialization is changed to Δ_2^0 . Although the initial set of medoids are far from optimal, the optimization stopped before any improvement in the medoids. This cause the mixture up of Western and Developing these two categories. This example illustrates that FCMdd is rather easy to be stuck in a local minimum before approaching better solutions.

For Fk-Parts, the results with different initializations are the same. The labels of all the objects except *Egypt* in Fk-Parts are identical with those produced by PAM, which forms three reasonable categories of countries. In Fk-Parts, *Egypt* is labeled as a Developing country while in PAM as a Western country. However, if we take a close look at the u values of *Egypt* in Fk-Parts, it can be found that the u values of *Egypt* in these two clusters are close. This indicates that *Egypt* does not so well belong to Developing or Western alone, but belongs to both categories. This shows the advantage of fuzzy clustering, since compared with the crisp labels of objects, more detailed information can be derived from the fuzzy memberships produced.

Further more, based on the v values of Fk-Parts in Table 3.4, it can be seen that *USA*, *Cuba* and *Zaire* are respectively the most representative countries in their categories. This is consistent with the medoids found by PAM. Moreover, based on V , we obtain the ranks of countries in each category and more detailed information on the internal cluster structure, e.g. *Israel* is much less representative than other three countries in the category of Western, and in the Developing category, *Egypt* and *India* are much less representative than other two countries. This example shows that Fk-Parts is a new approach with high potential since it is able to perform well on relational data by producing both good clusters and representatives, meanwhile more detailed information on the data structure can be obtained at the end of the clustering process.

Table 3.2: Countries Data [1]: Dissimilarity Matrix ($d_{ij} \in D$)

Countries	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
C1: Belgium	0.00	5.58	7.00	7.08	4.83	2.17	6.42	3.42	2.50	6.08	5.25	4.75
C2: Brazil	5.58	0.00	6.50	7.00	5.08	5.75	5.00	5.50	4.92	6.67	6.83	3.00
C3: China	7.00	6.50	0.00	3.83	8.17	6.67	5.58	6.42	6.25	4.25	4.50	6.08
C4: Cuba	7.08	7.00	3.83	0.00	5.83	6.92	6.00	6.42	7.33	2.67	3.75	6.67
C5: Egypt	4.83	5.08	8.17	5.83	0.00	4.92	4.67	5.00	4.50	6.00	5.75	5.00
C6: France	2.17	5.75	6.67	6.92	4.92	0.00	6.42	3.92	2.25	6.17	5.42	5.58
C7: India	6.42	5.00	5.58	6.00	4.67	6.42	0.00	6.17	6.33	6.17	6.08	4.83
C8: Israel	3.42	5.50	6.42	6.42	5.00	3.92	6.17	0.00	2.75	6.92	5.83	6.17
C9: USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75	0.00	6.17	6.67	5.67
C10: USSR	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17	0.00	3.67	6.50
C11: Yugoslavia	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67	0.00	6.92
C12: Zaire	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92	0.00

Table 3.3: Results of FCMdd for the Countries Data

Country	FCMdd (m=1.5) with Δ_1^0					FCMdd (m=1.5) with Δ_2^0				
	u_1	u_2	u_3	label	medoid	u_1	u_2	u_3	label	medoid
Belgium	0.7572	0.1280	0.1148	1	-	0.0000	0.0000	1.0000	3	*
France	0.7963	0.1059	0.0978	1	-	1.0000	0.0000	0.0000	1	*
Israel	0.7371	0.1164	0.1464	1	-	0.3722	0.1388	0.4890	3	-
USA	1.0000	0.0000	0.0000	1	*	0.5251	0.0495	0.4254	1	-
China	0.2264	0.4896	0.2840	2	-	0.2024	0.6138	0.1838	2	-
Cuba	0.0997	0.7515	0.1488	2	-	0.0000	1.0000	0.0000	2	*
USSR	0.0000	1.0000	0.0000	2	*	0.1357	0.7246	0.1397	2	-
Yugoslavia	0.1816	0.5998	0.2186	2	-	0.2407	0.5028	0.2565	2	-
Brazil	0.3980	0.2166	0.3854	1	-	0.3654	0.2466	0.3880	3	-
Egypt	0.4014	0.2258	0.3727	1	-	0.3637	0.2590	0.3773	3	-
India	0.0000	0.0000	1.0000	3	*	0.3180	0.3640	0.3180	2	-
Zaire	0.3186	0.2424	0.4390	3	-	0.3247	0.2272	0.4481	3	-

Table 3.4: Results of Fk-Parts and PAM for the Countries Data

Country	Fk-Parts with Δ_1^0 and Δ_2^0							PAM	
	u_1	u_2	u_3	lable	v_1	v_1	v_3	label	medoid
Belgium	0.8018	0.0000	0.1982	1	0.2726	0.0000	0.0856	1	-
France	0.8343	0.0000	0.1657	1	0.2559	0.0000	0.0483	1	-
Israel	0.7707	0.0000	0.2293	1	0.1825	0.0000	0.0384	1	-
USA	0.8147	0.0000	0.1853	1	0.2891	0.0000	0.0923	1	*
China	0.0662	0.8623	0.0716	2	0.0000	0.1964	0.0000	2	-
Cuba	0.0000	1.0000	0.0000	2	0.0000	0.3040	0.0000	2	*
USSR	0.0247	0.9681	0.0072	2	0.0000	0.2865	0.0000	2	-
Yugoslavia	0.1530	0.8224	0.0246	2	0.0000	0.2131	0.0000	2	-
Brazil	0.2674	0.0000	0.7326	3	0.0000	0.0000	0.2261	3	-
Egypt	0.4056	0.0304	0.5640	3	0.0000	0.0000	0.1542	1	-
India	0.1646	0.2540	0.5815	3	0.0000	0.0000	0.1245	3	-
Zaire	0.2551	0.0000	0.7449	3	0.0000	0.0000	0.2306	3	*

3.4.4 Document Categorization

Now we simulate the task of document categorization to evaluate the effectiveness of the proposed clustering algorithm.

3.4.4.1 Data Sets From 20newsgroups and CLUTO

We use datasets from multiple benchmark document collections which are frequently used as sources for testing the performance of document clustering algorithms. A summary of eight datasets with various scales and topic balance is given in Table 3.5. A highlight of each is given below:

- **Multi5** and **Multi10** are two subsets extracted from **20Newsgroups** [117]. The total collection consists approximately 20,000 newsgroup articles collected from 20 different newsgroups. **Multi5** contains 100 documents from each of the following five topics: *comp.graphics*, *rec.motorcycles*, *rec.sports.baseball*, *sci.space*, and *talk.politics.mideast*. **Multi10** contains 50 documents from each of the following 10 topics: *alt.atheism*, *comp.sys.mac.hardware*, *misc.forsale*, *rec.autos*, *rec.sport.hockey*, *sci.crypt*, *sci.electronics*, *sci.med*, *sci.space*, and *talk.politics.guns*.
- **tr12/31** datasets are derived from the TREC collections (<http://trec.nist.gov>). These corpora have been considered the ideal test set for document clustering

Table 3.5: Summary of eight document datasets (n : # of documents, s : # of words, k : # of classes, $\bar{n}_c$: average number of documents per class, and *Balance*: the size ratio of the smallest class to the largest one)

Data	Source	n	s	k	$\bar{n}_c$	<i>Balance</i>
Multi5	20Newsgroups	494	1000	5	99	0.95
Multi10	20Newsgroups	497	1000	10	50	0.98
tr12	TREC	313	5804	8	39	0.0968
tr31	TREC	927	10128	7	132	0.0688
la1	LA Times (TREC)	3204	13195	6	534	0.290
la2	LA Times (TREC)	3075	12432	6	513	0.274
re0	Reuters-21578	1504	2886	13	115	0.0181
re1	Reuters-21578	1657	3758	25	66	0.0270

purpose. Other two large datasets **la1** and **la2** are derived from LA Times (TREC).

- **re0** and **re1** are derived from Reuters-21578 Text Categorization Test Collection Distribution 1.0.¹ The **Reuters**-21578 collection contains the documents appeared on the Reuters newswire in 1987. It is a standard text categorization benchmark that contains 135 categories in total.

The two 20Newsgroups datasets are processed with the Matrix Creation tool (MC) [118] by removing stop words and words with occurrence less than 0.05 and more than 0.95 of the number of documents. No other additional feature selection process or stemming was used. Other datasets are available from CLUTO² and have been already processed. We further remove words that occur in two or less than two documents.

Each document is represented as a vector with tf-idf weighting, and cosine measure defined in Eq. (2.2) is used to calculate the similarity of two documents $r_{cos}(ij)$. The corresponding cosine-based dissimilarity is computed as

$$d_{cos}(ij) = 1 - r_{cos}(ij) \quad (3.45)$$

To make comparison, we test the performance of the proposed approach *Fk*-Parts, together with several other representative (dis)similarity-based partitioning clustering

¹<http://www.daviddlewis.com/resources/testcollections>

²<http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download>

approaches, including KK-means (kernel k-means) [119], PAM (k-medoids approach) [1], and FCMdd (fuzzy k-medoids approach) [115]. We also compare Fk-Parts with Hierarchical clustering UPGMA (hierarchical agglomerative clustering with average group linkage) [36] and matrix factorization based approach SNMF (symmetric non-negative matrix factorization) [79]. For each dataset, the result of each approach except UPGMA is the mean value of 30 trials with random initializations. The results of PAM is obtained with the function *pam* in the cluster package of **R**. To make a direct comparison, the same set of initializations are used for PAM, KernalKmeans, FCMdd and Fk-Parts. For all datasets, the number of clusters k is set to be the real number of classes and we set $m = 1.01$ for FCMdd and $\phi = 0.0001, \eta = 1$ for Fk-Parts for all the datasets.

To illustrate the significance of improvement made by the proposed approach compared with each of the existing ones, the *two-sample t-test* is used to determine if there are significant differences between the mean values of each of the measures (Accuracy and NMI) obtained by two approaches. The *null hypothesis* H_0 is “no significance of differences”. The t score is given by

$$t = \frac{\bar{q}_1 - \bar{q}_2}{\sqrt{\frac{sd_1^2}{h_1} + \frac{sd_2^2}{h_2}}} \quad (3.46)$$

where for $i = 1, 2$, $\bar{q}_i$ and sd_i are the mean and standard deviation of sample i , respectively, h_i is the size of sample i . Within 95% confidence interval, the critical value of t , $t_{critical}$ is 2.002 at degree of freedom $df = 58$ ($df = h_1 + h_2 - 2$, and in our experiments $h_1 = h_2 = 30$). Thus, if the calculated $t < t_{critical}$, the *null hypothesis* H_0 is *Accepted*, otherwise it is *Rejected*, which means there is a significance of differences between two results. We use “ $\gg$ ” to represent that the result on the left side is significantly better than the one on the right side, and use “ $>$ ” to denote that the result on the left side is better than the one on the right side, but the improvement is not significant.

3.4.4.2 Results and Discussions

The *Accuracy* and *NMI* scores of the six approaches on each of the datasets are given in Table 3.6. We first compare the four partitioning clustering approaches, namely

Figure 3.4: Clustering results of four (dis)similarity-based partitioning approaches for document datasets in Table 3.5.

PAM, FCMdd, KK-means and *Fk*-Parts. To make it easier for comparison, we plotted the results of these four approaches in Figure 3.4.

It is shown from Table 3.6 and Figure 3.4 that *Fk*-Parts overall performs the best among the four partitioning approaches. Compared to traditional medoid-based approaches PAM and FCMdd, significant improvements are achieved by *Fk*-Parts on all the eight datasets with respect to both *Accuracy* and *NMI*. This confirms that the way used in *Fk*-Parts to characterize each cluster is more effective than the one used in the other two. Compared with other cases, the improvements in terms of *NMI* on *re0* and *re1* are relatively less significant. For the two single-medoid-based approaches PAM and FCMdd, it is observed that both the *Accuracy* and *NMI* scores of FCMdd are lower than those PAM. This indicates that the improvements in efficiency of medoid search in FCMdd has the tradeoff in its effectiveness for clustering. We have observed in our experiments that the FCMdd algorithm frequently stops at premature

solutions due to the simple search procedure of medoids used in FCMdd. This makes FCMdd only produce clusters that have not been improved much from the initial state. Other than medoid-based approach, the *Fk*-Parts approach is also shown to be substantially better than the kernel-based partitioning approach KKmeans on almost all the datasets. The *NMI* values of *Fk*-Parts and KK-means are comparable on *re0*, while the *Accuracy* value of *Fk*-Parts is till much better than that of KK-means on this dataset.

We now compare *Fk*-Parts with hierarchial clustering UPGMA and matrix factorization based approach SNMF. Table 3.7 shows the results of significance test of the *Accuracy* and *NMI* values on each dataset produced by these three approaches. It is seen that *Fk*-Parts achieves significant improvements compared with both of other two approaches on five out of the eight datasets. Compared to SNMF, the proposed *Fk*-Parts achieves better *Accuracy* and *NMI* values on all the datasets except *re0*, where SNMF has a slightly better *NMI* value. Although the results of UPGMA on *tr31* and *re1* are the best of the three, UPGMA performs worse than the other two on most of the left datasets, especially on *la1* and *la2*. All these results show that the proposed partitioning approach *Fk*-Parts is a more favorable choice for document clustering, since it is more effective than existing partitioning clustering approaches and matrix factorization based approach, and with a smaller complexity, the performance of *Fk*-Parts is more consistent and stable on various datasets than the hierarchical clustering approach UPGMA.

3.5 Summary

We have proposed a new partitioning fuzzy clustering approach called *Fk*-Parts, which finds not only fuzzy clusters but also weighted medoids or representative objects for each cluster. We believe that in real-world similarity-based data analysis, information on both cluster labeling and internal structure of each of the produced clusters is important. One distinctive characteristic of the proposed method is that we apply the quadratic regularization technique to obtain both the fuzzy memberships and prototype weights of objects in each cluster for relational data analysis. Compared

Table 3.6: Performance comparison of the six relational clustering approaches for the eight datasets given in Table 3.5

Accuracy						
	PAM	FCMdd	KKmeans	SNMF	UPGMA	Fk-Parts
Multi5	77.94 $\pm$ 0.00	56.01 $\pm$ 8.64	74.69 $\pm$ 12.09	80.35 $\pm$ 11.75	88.87 $\pm$ 0.00	89.76 $\pm$ 9.05
Multi10	51.51 $\pm$ 0.00	33.53 $\pm$ 2.90	44.44 $\pm$ 4.12	67.38 $\pm$ 5.42	58.95 $\pm$ 0.00	79.57 $\pm$ 3.64
tr12	49.20 $\pm$ 0.00	32.79 $\pm$ 5.18	52.68 $\pm$ 6.82	60.32 $\pm$ 4.48	49.52 $\pm$ 0.00	68.00 $\pm$ 1.73
tr31	53.40 $\pm$ 0.00	50.33 $\pm$ 8.70	52.39 $\pm$ 6.62	59.16 $\pm$ 6.16	71.95 $\pm$ 0.00	60.77 $\pm$ 3.11
la1	57.18 $\pm$ 0.00	39.64 $\pm$ 3.82	50.80 $\pm$ 6.43	60.51 $\pm$ 6.52	29.40 $\pm$ 0.00	63.80 $\pm$ 3.67
la2	47.90 $\pm$ 0.00	31.51 $\pm$ 4.99	50.99 $\pm$ 4.04	61.45 $\pm$ 5.97	29.50 $\pm$ 0.00	64.79 $\pm$ 2.56
re0	33.38 $\pm$ 0.00	33.61 $\pm$ 2.82	36.55 $\pm$ 3.34	41.20 $\pm$ 3.69	43.05 $\pm$ 0.00	43.15 $\pm$ 1.56
re1	39.83 $\pm$ 0.00	35.61 $\pm$ 3.56	37.92 $\pm$ 2.57	38.53 $\pm$ 2.22	53.17 $\pm$ 0.00	47.44 $\pm$ 1.31

NMI						
	PAM	FCMdd	KKmeans	SNMF	UPGMA	Fk-Parts
Multi5	51.13 $\pm$ 0.00	30.93 $\pm$ 8.33	63.22 $\pm$ 10.01	75.20 $\pm$ 5.25	70.98 $\pm$ 0.00	79.79 $\pm$ 5.32
Multi10	39.52 $\pm$ 0.00	22.03 $\pm$ 3.01	34.68 $\pm$ 4.17	59.33 $\pm$ 3.59	52.33 $\pm$ 0.00	68.18 $\pm$ 2.11
tr12	46.43 $\pm$ 0.00	15.21 $\pm$ 7.55	48.93 $\pm$ 7.02	59.58 $\pm$ 3.34	47.44 $\pm$ 0.00	66.78 $\pm$ 1.09
tr31	39.40 $\pm$ 0.00	30.37 $\pm$ 6.83	49.98 $\pm$ 4.89	49.52 $\pm$ 5.24	58.97 $\pm$ 0.00	54.05 $\pm$ 5.04
la1	32.65 $\pm$ 0.00	16.34 $\pm$ 4.36	39.48 $\pm$ 5.05	46.55 $\pm$ 3.56	1.48 $\pm$ 0.00	51.18 $\pm$ 2.19
la2	27.28 $\pm$ 0.00	5.98 $\pm$ 6.53	40.59 $\pm$ 4.47	46.18 $\pm$ 3.34	1.71 $\pm$ 0.00	51.82 $\pm$ 2.02
re0	34.24 $\pm$ 0.00	25.14 $\pm$ 2.53	38.74 $\pm$ 3.23	37.88 $\pm$ 2.22	18.94 $\pm$ 0.00	37.76 $\pm$ 0.93
re1	49.36 $\pm$ 0.00	34.61 $\pm$ 2.85	50.68 $\pm$ 1.87	52.56 $\pm$ 1.28	54.54 $\pm$ 0.00	52.25 $\pm$ 0.79

Table 3.7: Significance test of improvement of Fk-Parts compared with SNMF and UPGMA

	Accuracy	NMI
Multi5	$Fk\text{-Parts} > UPGMA \gg SNMF$	$Fk\text{-Parts} \gg UPGMA \gg SNMF$
Multi10	$Fk\text{-Parts} \gg SNMF \gg UPGMA$	$Fk\text{-Parts} \gg SNMF \gg UPGMA$
tr12	$Fk\text{-Parts} \gg SNMF \gg UPGMA$	$Fk\text{-Parts} \gg SNMF \gg UPGMA$
tr31	$UPGMA \gg Fk\text{-Parts} > SNMF$	$UPGMA \gg Fk\text{-Parts} \gg SNMF$
la1	$Fk\text{-Parts} \gg SNMF \gg UPGMA$	$Fk\text{-Parts} \gg SNMF \gg UPGMA$
la2	$Fk\text{-Parts} \gg SNMF \gg UPGMA$	$Fk\text{-Parts} \gg SNMF \gg UPGMA$
re0	$Fk\text{-Parts} > UPGMA \gg SNMF$	$SNMF > Fk\text{-Parts} \gg UPGMA$
re1	$UPGMA \gg Fk\text{-Parts} \gg SNMF$	$UPGMA \gg SNMF > Fk\text{-Parts}$

“ $\gg$ ” means $t > t_{critical}$, and an improvement is significant;
 “ $>$ ” means $0 < t \leq t_{critical}$, and an improvement is not significant.

with existing (dis)similarity-based fuzzy clustering, where no medoid is defined or each cluster is represented by one object, the Fk -Parts approach has several merits. Rather than using heuristic search methods, the solution of medoids can be obtained in terms of the prototype weights, which are found in a continuous space with the formula derived from the objective function. Moreover, using prototype weights enables Fk -Parts to capture the internal cluster structure more precisely and completely, and hence to improve the quality of the final clusters. Our experimental results on benchmark document datasets show that Fk -Parts can be a very promising tool for relational data analysis. Although, the pairwise similarity relation of documents used in our experiments is the cosine similarity based on the “bag of words” representation of documents, it could be obtained with any other text similarity measures, or derived from other types of data, such as links among documents.

Chapter 4

Relation Integrated Fuzzy c-Means

4.1 Overview

Most of the existing fuzzy clustering approaches cluster documents in a dataset based on either the vector representation or their pairwise relationships. In this chapter, we develop a new fuzzy clustering approach called LinkFCM, which makes use of information from both vector-based data and relational data to form clusters. The objective function of LinkFCM consists of two terms as two different criteria to measure the quality of clusters. The first term is the sum of intra-cluster distances, which is the same as the criterion used in fuzzy c-means type approaches. The other term is the sum of intra-cluster connectivity defined in terms of pairwise relationships among objects in the same cluster. Minimizing the objective function of LinkFCM tends to minimize the total intra-cluster distance and at the same time maximize the total intra-cluster connectivity. The LinkFCM approach is specially useful for those document datasets, in which other than the vector-represented data, relational information of documents is also available. The relational information may be directly obtained or derived with some measures from the dataset, or provided by the user as a kind of domain knowledge. We will show the connection between LinkFCM and constrained fuzzy c-means when the relational data is in the form of user specified pairwise constraints. Experimental studies are conducted to show the clustering performance of LinkFCM on both simulated and real-world datasets. The results of our experiments show that LinkFCM achieves significant improvement in the effectiveness of clustering compared with existing fuzzy approaches.

4.2 Existing Fuzzy Clustering Approaches

In Chapter 2, we have reviewed two versions of fuzzy c-means referred as FCMm and FCMq, in which the fuzzifier m and quadratic regularization are used, respectively, to generate the fuzziness in memberships. Before presenting the proposed approach, we discuss other three related fuzzy approaches namely a cosine distance based fuzzy c-means, fuzzy clustering for relational data, and fuzzy clustering with user specified constraints.

4.2.1 Hyperspherical fuzzy c-means (HFCM)

Since Euclidean norm is not effective for distance measure of high dimensional data, cosine distance is used instead to obtain a modified fuzzy c-means for document clustering called HFCM (Hyperspherical fuzzy c-means) [51]. Given a document set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, where each $\mathbf{x}_i \in \mathbb{R}^s$ is an s dimensional vector. Assume that before clustering, each document is normalized to have a unit length, i.e., $\forall i, \|\mathbf{x}_i\| = 1$. The objective function of HFCM is given as

$$J_{HFCM} = \sum_{c=1}^k \sum_{i=1}^n u_{ic}^m d_{ic}^2(cos) \quad (4.1)$$

which is in the same form as the original fuzzy c-means as in Eq. (2.12) but $d_{ic}^2(cos)$ now is a cosine-based distance calculated as below:

$$d_{ic}^2(cos) = 1 - \langle \mathbf{x}_i, \delta_c^{HFCM} \rangle = 1 - \sum_{p=1}^s x_{pi} \delta_{pc}^{HFCM} \quad (4.2)$$

where δ_c^{HFCM} is the centroid of cluster c , and $\|\delta_c^{HFCM}\| = 1$. Since both $\mathbf{x}_i$ and δ_c^{HFCM} have a unit length, the cosine similarity between them equals to the inner product between them, and thus Eq. (4.2) defines a cosine-based distance.

The update equation of δ_c^{HFCM} for HFCM is accordingly derived as [51]:

$$\delta_c^{HFCM} = \frac{\sum_{i=1}^n u_{ic}^m \mathbf{x}_i}{\sqrt{\sum_{p=1}^s (\sum_{i=1}^n u_{ic}^m x_{pi})^2}} \quad (4.3)$$

It is easy to check that with the above formula, $\|\delta_c^{HFCM}\| = 1$ is always guaranteed. Since u_{ic} is only dependent on $d_{ic}^2(cos)$, the update formula of u_{ic} is in the same form as Eq. (2.14).

4.2.2 Fuzzy Relational Clustering

Fuzzy relational clustering produces fuzzy clusters based on the relation matrix which records dissimilarities of each pair of objects. In FCMm as shown in Eq. (2.14), the membership is updated based on d_{ic}^2 , which is the Euclidean distance between $\mathbf{x}_i$ and δ_c . By substituting δ_c with its update equation Eq. (2.13) into Eq. (2.14), the following equation is derived [120]:

$$\begin{aligned}
 d_{ic}^2 &= \|\mathbf{x}_i - \delta_c\|^2 \\
 &= (\mathbf{x}_i - \delta_c)^T (\mathbf{x}_i - \delta_c) \\
 &= \left(\mathbf{x}_i - \frac{\sum_{j=1}^n u_{jc}^m \mathbf{x}_j}{\sum_{j=1}^n u_{jc}^m} \right)^T \left(\mathbf{x}_i - \frac{\sum_{l=1}^n u_{lc}^m \mathbf{x}_l}{\sum_{l=1}^n u_{lc}^m} \right) \\
 &= \frac{\sum_{j=1}^n u_{jc}^m d'_{ji}}{\sum_{j=1}^n u_{jc}^m} - \frac{\sum_{j=1}^n u_{jc}^m \sum_{l=1}^n u_{lc}^m d'_{jl}}{2(\sum_{j=1}^n u_{jc}^m)^2}
 \end{aligned} \tag{4.4}$$

where $d'_{ij} \in \mathbf{D} = d_{ij}^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2$. let

$$\delta'_c(\mathbf{u}_c) = \frac{\mathbf{u}'_c}{\sum_{j=1}^n u_{jc}^m} \tag{4.5}$$

where $\mathbf{u}' = [u_{1c}^m, u_{2c}^m, \dots, u_{nc}^m]^T$, then

$$d_{ic}^2 = (\mathbf{D}\delta'_c)_i - \frac{1}{2}\delta_c'^T \mathbf{D}\delta'_c \tag{4.6}$$

It can be observed that d_{ic}^2 in the above equation is calculated only based on memberships in $\mathbf{U}$ and dissimilarities in $\mathbf{D}$. This means no vector-based data is required. This approach is called Relational dual of fuzzy c-means (RFCM) [120]. The alternating updating of d_{ic}^2 in Eq. (4.6) and u_{ic} in Eq. (2.14) locally minimizes the following objective function

$$J_{RFCM} = \sum_{c=1}^k \frac{\sum_{i=1}^n \sum_{j=1}^n u_{ic}^m u_{jc}^m d'_{ij}}{2 \sum_{i=1}^n u_{ic}^m} \tag{4.7}$$

Since in RFCM, $\mathbf{D}$ is required to be “Euclidean type” distance, i.e., each entry of $\mathbf{D}$ corresponds to the Euclidean distance of two objects in some p dimensional space, an extended version of RFCM called NERFCM (Non-Euclidean RFCM) is further

proposed to handle non-Euclidean dissimilarity data in [121]. Another closely related study in [122] derives the update formula of memberships directly to minimize the objective function in Eq. (4.7). Other than handling one dissimilarity matrix, the work in [123] extends several existing fuzzy relational approaches to be able to consider multiple dissimilarity matrices. All these approaches assume that $\mathbf{D}$ is a pairwise dissimilarity relation between objects in the dataset, and the following conditions are satisfied:

$$\forall i, j, \quad d_{ij} \geq 0, d_{ii} = 0, d_{ij} = d_{ji} \quad (4.8)$$

4.2.3 Constrained Fuzzy Clustering

In constrained clustering, pairwise constraints are given as a type of supervision to guide the clustering process. Two types of constraints are typically used in the literature, namely must-link and cannot-link. If a pair of objects is in the must-link set $\mathcal{M}$, then these two objects should be grouped in the same cluster. If a pair of objects is in the cannot-link set $\mathcal{C}$, then these two objects should not be clustered in the same cluster or in other words, they should be placed in two different clusters. In [124], a term defined as below is used to measure the violation of constraints in terms of fuzzy memberships:

$$T_v(\mathbf{U}, \mathcal{M}, \mathcal{C}) = \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}} \sum_{c=1}^k \sum_{f \neq c} u_{ic} u_{jf} + \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}} \sum_{c=1}^k u_{ic} u_{jc} \quad (4.9)$$

Adding this term into the objective function of the competitive agglomeration forms the Pairwise Constrained Competitive Agglomeration (PCCA) given as [124]:

$$J_{PCCA} = \sum_{c=1}^k \sum_{i=1}^n u_{ic}^2 d_{ic}^2 - \alpha \sum_{c=1}^k \left(\sum_{i=1}^n u_{ic} \right)^2 + \beta T_v(\mathbf{U}, \mathcal{M}, \mathcal{C}) \quad (4.10)$$

where the first two terms in Eq. (4.10) form the objective function of the unsupervised competitive agglomeration approach. α and β are two parameters. Minimizing J_{PCCA} requires to minimize T_v , and thus the clustering results tend to be consistent with user given constraints as much as possible. The same term T_v has also been incorporated

into a fuzzy relational clustering approach to form a semi-supervised fuzzy relational clustering in [99].

The update formula of memberships for minimizing J_{PCCA} in Eq. (4.10) is derived as [124]:

$$u_{ic} = \frac{\frac{1}{d_{ic}^2}}{\sum_{f=1}^k \frac{1}{d_{if}^2}} + \frac{\beta(\bar{Q} - Q_c)}{2d_{ic}^2} + \frac{\alpha(N_c - \bar{N})}{d_{ic}^2} \quad (4.11)$$

where

$$Q_c = \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}} \sum_{f=1, f \neq c}^k u_{jf} + \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}} u_{jc} \quad (4.12)$$

$$\bar{Q} = \frac{\sum_{f=1}^k \frac{Q_f}{d_{if}^2}}{\sum_{f=1}^k \frac{1}{d_{if}^2}} \quad (4.13)$$

$$N_c = \sum_{i=1}^n u_{ic}, \quad \bar{N} = \frac{\sum_{f=1}^k \frac{N_f}{d_{if}^2}}{\sum_{f=1}^k \frac{1}{d_{if}^2}} \quad (4.14)$$

A more complete and extended study of this approach is given in [98], where the selection of effective constraints and parameter settings are discussed.

4.3 Motivation

In many real-world applications, given a set of objects, we may have both the vector representation of the objects and pairwise relationships among these objects. For a synthetic document dataset illustrated in Figure 4.1, together with the document-word co-occurrence matrix, where each document is described by a set of words, we may also extract a matrix to record the relational information among these documents, e.g., we may define that two documents are related if one of them is cited by the other, or if the two documents are written by the same author. Since existing fuzzy c-means type clustering approaches only deal with vector-based data, while fuzzy relational clustering approaches only handle relational data, the content information and link information in the forms of these two types of data formats are unable to be considered at the same time in those existing fuzzy approaches without any data transformation

Figure 4.1: Illustration of a document dataset for which two matrices are extracted: the paper-word co-occurrence matrix and the paper-paper relation matrix. The relation between two papers may be defined in various ways to describe specific relationships between two papers, e.g., two papers are said to be related if one of them is cited by the other, or the paper-paper relation defines whether two papers are written by the same author.

process. A new fuzzy clustering approach which makes use of both vector-based data and relational data is thus desirable in order to make use of both in the clustering process.

4.4 Proposed LinkFCM

Now we formally present the details of our new approach LinkFCM, including the formulation, algorithm and parameter setting.

4.4.1 Formulation

Given a dataset with n objects $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, where each object is an s dimensional vector, i.e., $\mathbf{x}_i \in \mathbb{R}^s$, we use $\mathbf{X}_{s \times n}$ to denote the object-feature matrix or data matrix, and $\mathbf{W}_{n \times n}$ to denote the relation matrix. Each $x_{pi} \in \mathbf{X}$ represents the weight of the p th feature in the i th object, and each $w_{ij} \in \mathbf{W}$ represents the relationship or association between the i th object and the j th object. The larger the w_{ij} , the more

strongly related or more similar of the two objects. We assume $\forall i, w_{ii} = 0$. In existing fuzzy c-means type approaches as reviewed early, the intra-cluster distance based on the vector-based data is used to measure the quality of clusters. In order to make use of the information represented as relational data, we define another criterion for measuring the quality of clusters in terms of pairwise relationships as below:

$$T_R^c(c, \mathbf{W}, \mathbf{U}) = \sum_{i=1}^n \sum_{j=1}^n w_{ij} u_{ic} u_{jc} \quad (4.15)$$

$$T_R(\mathbf{W}, \mathbf{U}) = \sum_{c=1}^k T_R^c = \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n w_{ij} u_{ic} u_{jc} \quad (4.16)$$

If the partitions are hard, i.e., $\exists f, u_{jf} = 1$, and $\forall c \neq f, u_{jc} = 0$, then T_R^c is the total pairwise relationships between objects that assigned to cluster c , which measures the degree of compactness of a cluster. The meaning of T_R^c still can be explained in the same way when the partitions are soft. Now we give the objective function of the proposed LinkFCM in the form as

$$J_{LinkFCM} = T_X(\mathbf{X}, \mathbf{U}, \Delta) - \beta T_R(\mathbf{W}, \mathbf{U}) \quad (4.17)$$

where $T_X(\mathbf{X}, \mathbf{U}, \Delta)$ represents the sum of intra-cluster distance as which is used as the objective function in fuzzy c-means type approaches, and $T_R(\mathbf{W}, \mathbf{U})$ is the term defined in Eq. (4.16). Minimizing $J_{LinkFCM}$ requires to minimize T_X and maximize T_D , so the information in vector-based data and relational data are both considered in the clustering process. Parameter β balances the contributions between the two terms.

Here, we consider two instances of $T_X(\mathbf{X}, \mathbf{U}, \Delta)$. One is based on fuzzifier m and the other is based on quadratic regularization:

$$T_X^m(\mathbf{X}, \mathbf{U}, \Delta) = \sum_{c=1}^k \sum_{i=1}^n u_{ic}^2 d_{ic}^2(\mathbf{x}_i, \delta_c) \quad (4.18)$$

$$T_X^q(\mathbf{X}, \mathbf{U}, \Delta, \eta) = \sum_{c=1}^k \sum_{i=1}^n u_{ic} d_{ic}^2(\mathbf{x}_i, \delta_c) u_{ic} u_{jc} + \frac{\eta}{2} \sum_{c=1}^k \sum_{i=1}^n u_{ic}^2 \quad (4.19)$$

Accordingly, we have two variants of $J_{LinkFCM}$ called LinkFCMm and LinkFCMq with their objective functions given as

$$J_{LinkFCMm} = \sum_{c=1}^k \sum_{i=1}^n u_{ic}^2 d_{ic}^2 - \beta \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n w_{ij} u_{ic} u_{jc} \quad (4.20)$$

$$J_{LinkFCMq} = \sum_{c=1}^k \sum_{i=1}^n u_{ic} d_{ic}^2 - \frac{\beta}{2} \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n w_{ij} u_{ic} u_{jc} + \frac{\eta}{2} \sum_{c=1}^k \sum_{i=1}^n u_{ic}^2 \quad (4.21)$$

where the membership u_{ic} is subject to the same conditions as in Eq. (2.11). In LinkFCMm, we use a fuzzifier $m = 2$, while in LinkFCMq, we use the quadratic regularization. Fixing fuzzifier $m = 2$ in a fuzzy approach is a typical choice in the literature, e.g., [124], [125] and [126]. This setting simplifies the derivation of the update equation of memberships, and those fuzzy approaches usually work well with the fuzzifier being set to 2. However, in some cases, we do need to adjust the value of the fuzzifier in order to obtain reasonable results, e.g., when the memberships produced with $m = 2$ are too fuzzy to be useful, or more specifically, the memberships of objects in all the clusters are very similar. In such situations, we may need to decrease the value of the fuzzifier in order to reduce the fuzziness in the memberships. The parameter η in LinkFCMq plays the similar role as the fuzzifier. So in LinkFCMq, we can tune this parameter to adjust the amount of fuzziness of memberships if necessary.

4.4.2 Solution

Like many other fuzzy approaches, it is difficult to solve the problem formulated above in a closed form. We need to use an iterative algorithm to find the local solutions. We now use the method of Lagrange Multiplier to derive the update equations of u_{ic} for LinkFCMm and LinkFCMq, respectively. For LinkFCMm, the Lagrangian is constructed as

$$L_{LinkFCMm} = J_{LinkFCMm} + \sum_{i=1}^n \lambda_i \left(\sum_{c=1}^k u_{ic} - 1 \right) \quad (4.22)$$

where $\{\lambda_i\}_{i=1}^n$ are the Lagrange multipliers. According to the first order necessary condition

$$\frac{\partial L_{LinkFCMm}}{\partial u_{ic}} = 2u_{ic}d_{ic}^2 - 2\beta \sum_{j=1}^n w_{ij}u_{jc} + \lambda_i = 0 \quad (4.23)$$

which gives

$$u_{ic} = \frac{\beta \sum_{j=1}^n w_{ij} u_{jc} - \frac{\lambda_i}{2}}{d_{ic}^2} \quad (4.24)$$

To satisfy the summation constraints in Eq. (2.11), λ_i should follow

$$\frac{\lambda_i}{2} = \frac{\sum_{f=1}^k \frac{\beta \sum_{j=1}^n w_{ij} u_{jf}}{d_{if}^2} - 1}{\sum_{f=1}^k \frac{1}{d_{if}^2}} \quad (4.25)$$

We can write u_{ic} into two parts as

$$u_{ic} = u_{ic}^{FCMm} + \beta u_{ic}^{Re1} \quad (4.26)$$

with

$$u_{ic}^{FCMm} = \frac{\frac{1}{d_{ic}^2}}{\sum_{f=1}^k \frac{1}{d_{if}^2}} \quad (4.27)$$

$$u_{ic}^{Re1} = \frac{1}{d_{ic}^2} \left(\sum_{j=1}^n w_{ij} u_{jc} - \frac{\sum_{f=1}^k \frac{\sum_{j=1}^n w_{ij} u_{jc}}{d_{if}^2}}{\sum_{f=1}^k \frac{1}{d_{if}^2}} \right) \quad (4.28)$$

As shown in Eq. (4.27), u_{ic}^{FCMm} is the membership update formula of FCMm with $m = 2$ in Eq. (2.14). u_{ic}^{FCMm} is dependent on d_{ic}^2 , which is the distance between $\mathbf{x}_i$ to the centroid of cluster c . The smaller this distance is, the larger u_{ic}^{FCMm} the $\mathbf{x}_i$ is assigned with respect to that cluster. u_{ic}^{Re1} in Eq. (4.28) considers both the object-to-centroid distance d_{ic}^2 and the total association between the object and all other objects in that cluster $\sum_{j=1}^n w_{ij} u_{jc}$. A large u_{ic}^{Re1} is assigned to the object with respect to cluster c , if this object has a strong overall association with many other objects in that cluster and is close to the centroid of that cluster.

The update equation of u_{ic} for LinkFCMq can be derived in the similar way. First, we form the Lagrangian with Lagrange multipliers $\{\lambda_i\}_{i=1}^n$ as

$$L_{LinkFCMq} = J_{LinkFCMq} + \sum_{i=1}^n \lambda_i \left(\sum_{c=1}^k u_{ic} - 1 \right) \quad (4.29)$$

according to

$$\frac{\partial L_{LinkFCMq}}{\partial u_{ic}} = d_{ic}^2 - \beta \sum_{j=1}^n w_{ij} u_{jc} + \eta u_{ic} + \lambda_i = 0 \quad (4.30)$$

we obtain

$$u_{ic} = \frac{1}{\eta} \left(\beta \sum_{j=1}^n w_{ij} u_{jc} - d_{ic}^2 \right) - \frac{1}{\eta} \lambda_i \quad (4.31)$$

In order to satisfy the summation constraints in Eq. (2.11), λ_i needs to follow

$$\lambda_i = \frac{1}{k} \sum_{f=1}^k \left(\beta \sum_{j=1}^n w_{ij} u_{jf} - d_{if}^2 \right) - \frac{\eta}{k} \quad (4.32)$$

Similar as in LinkFCMm, here u_{ic} can also be written as the combination of two terms as below:

$$u_{ic} = \frac{1}{k} + \frac{1}{\eta} (u_{ic}^{FCMq} + \beta u_{ic}^{Re2}) \quad (4.33)$$

with

$$u_{ic}^{FCMq} = \frac{1}{k} \sum_{f=1}^k d_{if}^2 - d_{ic}^2 \quad (4.34)$$

$$u_{ic}^{Re2} = \sum_{j=1}^n w_{ij} u_{jc} - \frac{1}{k} \sum_{f=1}^k \sum_{j=1}^n w_{ij} u_{jf} \quad (4.35)$$

Here, u_{ic}^{FCMq} is the membership of quadratic regularized fuzzy c-means, which is inversely proportional to the object-to-centroid distance like u_{ic}^{FCMm} . It is observed that u_{ic}^{Re1} in Eq. (4.28) is a function of both object-to-centroid distance i.e., d_{ic}^2 , and the total association of the object to all other objects in that cluster, i.e., $\sum_{j=1}^n w_{ij} u_{jc}$, while u_{ic}^{Re2} in Eq. (4.35) is only decided with the later factor.

Given the fuzzy memberships, the updating of cluster centroid is dependent on the choice of d_{ic}^2 , which could be Euclidean distance as in Eq. (2.8), or cosine distance as in Eq. (4.2), or even other types of distances. In LinkFCMm, if d_{ic}^2 is defined as Euclidean distance, the updating of the cluster centroid follows Eq. (2.13) with $m = 2$. If d_{ic}^2 is in cosine distance, the formula in Eq. (4.3) with $m = 2$ is used to update the cluster centroid. In LinkFCMq, if d_{ic}^2 is defined as Euclidean distance, the updating of the cluster centroid follows Eq. (2.16), and if d_{ic}^2 is the cosine distance, δ_c is updated with the following formula:

$$\delta_c = \frac{\sum_{i=1}^n u_{ic} \mathbf{x}_i}{\sqrt{\sum_{p=1}^s (\sum_{i=1}^n u_{ic} x_{pi})^2}} \quad (4.36)$$

Like the fuzzy c-means algorithm, the final solutions of $\mathbf{U}$ and Δ in LinkFCM are obtained with an iterative algorithm, during which the memberships and cluster centroids are updated alternately based on each other. Since the update formulas of u_{ic} currently derived in Eq. (4.26) for LinkFCMm and in Eq. (4.33) for LinkFCMq do not guarantee the non-negativity, negative values of u_{ic} may appear during the optimization process. To derive solutions that guarantee the fuzzy memberships to be non-negative in each iteration, we need to use the Karush-Kuhn-Tucker (KKT) conditions. Since the KKT conditions would yield more complicated solutions, here we use the clipping strategy: after the memberships being updated at each iteration, we set those negative values to 0 and perform the re-normalization so that the summation constraint in Eq. (2.11) is still satisfied. This simple clipping has been employed in several fuzzy clustering approaches such as [85] and [125], as it is a convenient choice in such kind of situations. Although it works well in practice, convergence is not guaranteed.

For a dataset of n objects and s features, the time complexity of LinkFCM is $O(lknt)$, where l is the number of iterations, k is the number of clusters, and $t = \max\{s, n\}$.

4.4.3 Choice of β

According to Eq. (4.26) and Eq. (4.33), parameter β balances the contributions of two terms, respectively, i.e. u_{ic}^{FCMm} and u_{ic}^{Re1} in LinkFCMm and u_{ic}^{FCMq} and u_{ic}^{Re2} in LinkFCMq. To allow both terms to make balanced impact on the final memberships, β is usually set to make the values of two terms of the same magnitude. Here let $u_{ic}^{FCM} = \{u_{ic}^{FCMm}, u_{ic}^{FCMq}\}$ and $u_{ic}^{Re} = \{u_{ic}^{Re1}, u_{ic}^{Re2}\}$. Then we define

$$\beta_0 = \frac{\sum_{i=1}^n \sum_{c=1}^k |u_{ic}^{FCM}| / nnz(\mathbf{U}^{FCM})}{\sum_{i=1}^n \sum_{c=1}^k |u_{ic}^{Re}| / nnz(\mathbf{U}^{Re})} \quad (4.37)$$

as a reference value of β , where $|\cdot|$ is to get the absolute value, and $nnz(\cdot)$ is to calculate the number of non-zero entries in a matrix. β is proportional to β_0 as

$$\beta = \alpha \beta_0 \quad (4.38)$$

where α is specified by the user. The value of β is estimated at each iteration after u_{ic}^{FCM} and u_{ic}^{Re} are updated. The detailed description of the complete LinkFCMm algorithm is given in Algorithm 5. LinkFCMq follows the similar procedure as LinkFCMm.

Algorithm 5: LinkFCMm

input : Data matrix $\mathbf{X}_{s \times n}$, relation matrix $\mathbf{W}_{n \times n}$, a distance measure, number of clusters k , parameter α .
output: Fuzzy memberships $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k]$ and cluster centroids $\Delta = [\delta_1, \delta_2, \dots, \delta_k]$.
 Initialize membership matrix $\mathbf{U}^{(0)}$;
repeat
 for each $c = \{1, 2, \dots, k\}$ **do**
 update δ_c by Eq. (2.13) with $m = 2$ if *Euclidean* distance, or update δ_c by Eq. (4.3) with $m = 2$ if *cosine* distance;
 end
 for each $i = \{1, 2, \dots, n\}$ **do**
 for each $c = \{1, 2, \dots, k\}$ **do**
 calculate d_{ic}^2 by Eq. (2.8) if *Euclidean* distance or Eq. (4.2) if *cosine* distance;
 end
 for each $c = \{1, 2, \dots, k\}$ **do**
 calculate u_{ic}^{FCMm} and u_{ic}^{Re1} with Eq. (4.27) and Eq. (4.28);
 estimate β with Eq. (4.38);
 update u_{ic} by Eq. (4.26);
 end
 find $\mathcal{K}^- = \{f : u_{if} < 0\}$ and $\mathcal{K}^+ = \{f : u_{if} > 0\}$;
 if $\mathcal{K}^- \neq \emptyset$ **then**
 $\forall c \in \mathcal{K}^-$, set $u_{ic} = 0$;
 for each $c \in \mathcal{K}^+$ **do**
 $u_{ic} = u_{ic} / \sum_{f \in \mathcal{K}^+} u_{if}$;
 end
 end
 end
until converge or reaching the maximum iterations;

4.5 Constrained LinkFCM

We have just presented the formulation and the algorithm of a new approach LinkFCM, which searches for a fuzzy partitioning of the dataset with small intra-cluster

distances based on the vector-based data $\mathbf{X}$, and at the same time with large intra-cluster associations based on the relational data $\mathbf{W}$. Now we discuss a special case where instead of data-derived relation, the pairwise relationships in $\mathbf{W}$ now correspond to some constraints provided by the user as supervision in the clustering process. More specifically, assume that user specified must-link set $\mathcal{M}$ and cannot-link set $\mathcal{C}$ are available, where each pair of objects $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$ should be grouped together in the same cluster and each pair of objects $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$ should be placed in different groups. The relation matrix $w_{ij} \in \mathbf{W}$ is then constructed as below:

$$w_{ij} = \begin{cases} \theta_{ij}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \\ -\tilde{\theta}_{ij}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \\ 0, & \text{otherwise} \end{cases} \quad (4.39)$$

where θ_{ij} and $\tilde{\theta}_{ij}$ are penalties for violating the constraints in $\mathcal{M}$ and $\mathcal{C}$, respectively. Let

$$V = -T_R = -\sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n w_{ij} u_{ic} u_{jc}, \quad (4.40)$$

and thus maximizing T_R is equivalent to minimizing V . As being reviewed early, T_v defined in Eq. (4.9) measures the violation of constraints which is to be minimized in PCCA. We now show the equivalence between the above term V in Eq. (4.40) and T_v in PCCA through Theorem 1 given as follows.

Theorem 1 *Minimizing V in Eq. (4.40) is equivalent to minimizing the violation term in Eq. (4.9) when $\theta_{ij} = \tilde{\theta}_{ij} = 1$, i.e.,*

$$-a \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n w_{ij} u_{ic} u_{jc} + b = \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}} \sum_{c=1}^k \sum_{f \neq c} u_{ic} u_{jf} + \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}} \sum_{c=1}^k u_{ic} u_{jc} \quad (4.41)$$

with

$$w_{ij} = \begin{cases} 1, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \\ -1, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \\ 0, & \text{otherwise} \end{cases} \quad (4.42)$$

where a and b are two constants.

Proof: By substituting Eq. (4.42) into the left-hand side of Eq. (4.41), we have

$$-a \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n w_{ij} u_{ic} u_{jc} + b = 2a(I_1 + I_2) + b \quad (4.43)$$

with

$$I_1 = - \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}} \sum_{c=1}^k u_{ic} u_{jc} \quad (4.44)$$

$$I_2 = \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}} \sum_{c=1}^k u_{ic} u_{jc}. \quad (4.45)$$

Set $a = \frac{1}{2}$. I_2 is the same as the second term in the right-hand side of Eq. (4.41). So we only need to show

$$I_1 + b = \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}} \sum_{c=1}^k \sum_{f \neq c}^k u_{ic} u_{jf}$$

Since

$$\begin{aligned} I_1 &= - \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}} \sum_{c=1}^k u_{ic} u_{jc} \\ &= \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}} \left(\sum_{c=1}^k \sum_{f \neq c}^k u_{ic} u_{jf} - \sum_{c=1}^k \sum_{f=1}^k u_{ic} u_{jf} \right) \\ &= \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}} \left(\sum_{c=1}^k \sum_{f \neq c}^k u_{ic} u_{jf} - 1 \right) \end{aligned} \quad (4.46)$$

Let $b = \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}} 1$, we get

$$I_1 = \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}} \sum_{c=1}^k \sum_{f \neq c}^k u_{ic} u_{jf} - b \quad (4.47)$$

Combining all, now we complete the proof of Theorem 1. This equivalence shows that the proposed LinkFCM becomes a constrained fuzzy c-means when $\mathbf{W}$ is defined by Eq. (4.39). The constrained fuzzy c-means with $m = 2$ has been discussed in [127]. To our best knowledge, the constrained fuzzy c-means with quadratic regularization has not been discussed in any other place yet.

For a more general case where both data-derived association $\mathbf{W}_d$ and user specified constraints $\mathbf{W}_c$ are available, we may have the following objective function for the constrained LinkFCM (CLinkFCM):

$$J_{CLinkFCM} = T_X(\mathbf{U}, \mathbf{\Delta}, \mathbf{X}) - \beta_d T_D(\mathbf{W}_d, \mathbf{U}) - \beta_c T_C(\mathbf{W}_c, \mathbf{U}) \quad (4.48)$$

Accordingly we have

$$J_{CLinkFCMm} = \sum_{c=1}^k \sum_{i=1}^n u_{ic}^2 d_{ic}^2 - \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n (\beta_d w_{ij}^d + \beta_c w_{ij}^c) u_{ic} u_{jc} \quad (4.49)$$

$$\begin{aligned} J_{CLinkFCMq} &= \sum_{c=1}^k \sum_{i=1}^n u_{ic} d_{ic}^2 - \frac{1}{2} \sum_{c=1}^k \sum_{i=1}^n \sum_{j=1}^n (\beta_d w_{ij}^d + \beta_c w_{ij}^c) u_{ic} u_{jc} \\ &\quad + \frac{\eta}{2} \sum_{c=1}^k \sum_{i=1}^n u_{ic}^2 \end{aligned} \quad (4.50)$$

It is observed that if convexly combine the two relation matrices, i.e., $\mathbf{W} = \beta_d \mathbf{W}_d + \beta_c \mathbf{W}_c$, then CLinkFCM is equivalent to applying LinkFCM on the combined relation matrix.

4.6 Experiments

In this section, we conduct experimental studies of the proposed approach LinkFCM. To make it easier to discuss the clustering results thorough visualization, an simulated 2-D dataset is first tested. After that, we applied LinkFCM for real-world document categorization. For each dataset, other than the vector representation of the objects, we use different ways to obtain the relation matrix, and apply LinkFCM with both vector-based data and relational data to see whether better clustering results can be achieved than existing fuzzy approaches.

4.6.1 Artificial Data

We first show the clustering results of LinkFCMm on a simulated 2-D data plotted in Figure 4.2a. This dataset contains 175 objects that form two non-linearly separable

Figure 4.2: A 2-D dataset with two semi-circle shaped clusters. (a) Distribution of the dataset; (b) The clustering results with FCMm.

semi-circle shaped clusters. The clustering result produced by FCMm (fuzzy c-means with $m=2$) is shown in Figure 4.2b. Since clusters are assumed to have spherical shapes in Euclidean-based fuzzy c-means, it is observed from Figure 4.2b that 11 objects in the upper semi-circle are mis-grouped into the lower semi-circle; 12 objects in the lower semi-circle are mis-labeled as the upper one. Now we show the clustering result that LinkFCMm with Euclidean distance can achieve on this dataset with the help of additional information from the relation matrix.

4.6.1.1 Data-derived Relation: $\mathbf{W} = \mathbf{W}_d$

We first apply LinkFCMm with data-derived relation matrix, which defines the neighbors of each object. Specifically, each entry $w_{ij} \in \mathbf{W}$ indicates an association between an object and one of its 3-nearest neighbors, i.e., $w_{ij} = 1$ if $x_j \in \mathcal{N}_3(\mathbf{x}_i)$, where $\mathcal{N}_3(\mathbf{x}_i)$ is the set of 3-nearest neighbors of $\mathbf{x}_i$, otherwise $w_{ij} = 0$. Any distance measures may be used to select the neighbors of each object in different senses. Here we use the Euclidean norm to measure the distance between two objects. The links between each object and their 3-nearest neighbors is plotted in Figure 4.3a. Since the relational data is derived still based on the Euclidean distance, it contains a large amount of overlapped information as the vector-based data. Therefore, the results of LinkFCMm are expected to be similar as those of FCMm. Nevertheless, LinkFCMm still achieves

Figure 4.3: Clustering result of LinkFCMm using data-based relational information for the 2-D dataset. (a) Links between each object and their 3-nearest neighbors; (b) clustering result of LinkFCMm.

a better clustering result than FCMm as shown in Figure 4.3b. It can be observed that all objects in the upper semi-circle have been correctly labeled, although 9 objects in the lower semi-circle are still mis-labeled as the upper one.

4.6.1.2 User-provided Relation: $\mathbf{W} = \mathbf{W}_c$

Now we apply LinkFCMm with user specified pairwise constraints as relational data $\mathbf{W}$. Since we know the ground truth labels of all objects in the dataset, these pairwise constraints are generated in the following way: we first randomly select a pair of object, and then put them into the must-link set $\mathcal{M}$ if these two objects have the same labels, otherwise this pair is put into the cannot-link set $\mathcal{C}$. This process continues until a desired number of total pairwise constraints have been generated. For a dataset with n objects, the number of all possible pairs is $0.5 \times (n^2 - n)$. Here, $n = 175$, so the total number of pairs is 15225. We denote

$$y = |\mathcal{M}| + |\mathcal{C}|$$

as the number of pairwise constraints. Once $\mathcal{M}$ and $\mathcal{C}$ are formed, $\mathbf{W}$ is constructed as in Eq. (4.39) with $\theta_{ij} = \tilde{\theta}_{ij} = 1$. The clustering results with the number of pairwise constraints $y = 20, 80$ and 200 are shown in Figure 4.4.

From this figure, it is seen that the number of mis-labeled objects decreases when more constraints are used. With $y = 80$, only 6 objects are grouped into the wrong clusters. All the objects are correctly labeled when y is increased to 200, which accounts for 1.3% of the total number of pairs. Since the vector-based data used in different trials remains the same, the accuracy of LinkFCMm is dependent on the effectiveness of information obtained from the relation matrix. Comparing the links in Figure 4.4a, 4.4c, and 4.4e with those in Figure 4.3a, it can be observed that links in the former contain both must-link and cannot-link information while the later only contain must-link information. Moreover, the must-links in Figure 4.3a are between those geometrically close objects, which repeat the information contained in the vector-based data to a large extent, while in Figure 4.4a, 4.4c, and 4.4e, those must-links may exist between objects which are far away from each other in terms of Euclidean distance. On the other hand, the cannot-links in Figure 4.4a, 4.4c, and 4.4e may exist between objects which are relatively close in terms of Euclidean distance. This indicates that the information in the constraint-based relation matrix effectively compensates or enriches the information obtained from the vector-based data. Thus, LinkFCMm is able to produce the correct clustering result even for a dataset with non-spherical shaped clusters as long as enough constraints are provided.

4.6.2 Cora Data

Now we carry out experimental study on the real-world benchmark data Cora [128]. This data collection records the extracted information of computer science research papers published in journals and conferences in several related areas. Five sets of papers in topics of *Data Structures* (DS), *Hardware Architecture* (HA), *Machine Learning* (ML), *Operation System* (OS) and *Program Language* (PL) are used in our experiment. A summary of these five datasets is given in Table 4.1, and an original sample record is shown in Figure 4.5. We use the processed data which has been used in [129]. Each entry of the data matrix $x_{pi} \in \mathbf{X}$ is the number of occurrences of the p th word in the abstract of the i th paper. Each $\mathbf{x}_i$ is normalized to have unit length. The relation matrix of each dataset are obtained in two different ways as described below.

Figure 4.4: Clustering results of LinkFCMm with user-provided relation for the 2-D dataset. (a), (c) and (e) plots the pairwise constraints with $\gamma = \{20, 80, 200\}$, respectively, where black-solid lines indicate must-links and red-dash lines indicate cannot-links; (b), (d) and (f) show the corresponding clustering results of LinkFCMm.

<p>URL: http://pertserver.cs.uiuc.edu/papers/HaLi94a.ps</p> <p>Title: Validating Timing Constraints in Multiprocessor and Distributed Real-Time Systems</p> <p>Author: Rhan Ha and Jane W. S. Liu</p> <p>....</p>
<p>Abstract: In multiprocessor and distributed real-time systems, scheduling jobs dynamically on processors is likely to achieve better performance. However, analytical and efficient validation methods to determine whether all the timing constraints are met do not exist for systems using modern dynamic scheduling strategies, and exhaustive simulation and testing are unreliable and expensive. This paper describes several worst-case bounds and efficient algorithms for validating systems in which jobs have arbitrary timing constraints and variable execution times and are scheduled on processors dynamically in a priority-driven manner. ...</p>
<p>Reference: [1] <author> J. A. Stankovic, K. Ramamritham, and S. Cheng. </author> <title> Evaluation of a flexible task scheduling algorithm for distributed hard real-time systems. </title> <journal> IEEE Transactions on Computers, </journal> <volume> 34(12) </volume> <pages> 1130-1143, </pages> <month> December </month> <year> 1985. </year></p> <p>Reference: [2] <author> K.G. Shin and Y.C. Chang. </author> <title> Load sharing in distributed real-time systems with state-change broadcasts. </title> <journal> IEEE Transactions on Software Engineering, </journal> <volume> 38(8) </volume> <pages> 1124-1142, </pages> <month> August </month> <year> 1989. </year></p> <p>...</p> <p>References-found: 13</p>

(a)

Figure 4.5: A sample record of extracted information of a paper in the Cora data. Each record usually consists of three parts: the header part, which includes URL, Title, Author(s) and some other related information like Address, Affiliation and Email; the content of abstract; a list of formatted references.

4.6.2.1 Data-derived Relation: $\mathbf{W} = \mathbf{W}_d$

We first collect the relational data based on the citations among papers. To be more specifically, the value of $w_{ij}^d \in \mathbf{W}_d$ is 1 if any of the two papers is in the reference list of the other, and is 2 if two papers are mutually cited. Two papers $\mathbf{x}_i$ and $\mathbf{x}_j$ are said to be linked if $w_{ij} > 0$. As shown in Table 4.1, the relation matrix is quite sparse since each paper only cites a small number of other papers.

We compare LinkFCMm and LinkFCMq with the following three fuzzy approaches:

Table 4.1: Summary of Cora Datasets. n : # of papers, s : # of words, k : # of subfields, $\bar{l}$: average number of citations per paper.

	n	s	k	$\bar{l}$
Data Structure (DS)	751	6234	9	3
Hardware and Architecture (HA)	400	3989	7	4
Machine Learning (ML)	1617	8329	7	5
Operation System (OS)	1246	6737	4	7
Programming Language (PL)	1575	7949	9	6

- HFCM [51]: fuzzy c-means with cosine distance;
- NERFCM [121]: relational dual of fuzzy c-means for dissimilarity data;
- Fk -Parts: the weighted-medoids based fuzzy clustering proposed in Chapter 3 which works on a similarity matrix.

For HFCM, NERFCM and Fk -Parts, we get two sets of results based on different input matrices. Specifically, for the HFCM approach which works with vector-represented data, HFCM (word) refers to the result produced by HFCM based on the original vector-based data $\mathbf{X}$, and HFCM (word+link) refers to the result produced by HFCM based on $\mathbf{X}'$, the concatenated matrix of $\mathbf{X}$ and $\mathbf{W}$, i.e. $\mathbf{X}' = \begin{bmatrix} \mathbf{X} \\ \mathbf{W} \end{bmatrix}$; For relational clustering approaches NERFCM and Fk -Parts, NERFCM (link)/ Fk -Parts(link) refers to the results of NERFCM/ Fk -Parts based on the citation relation $\mathbf{W}$, and NERFCM (word+link)/ Fk -Parts (word+link) refers to the result of NERFCM/ Fk -Parts with the combined relation matrix $\mathbf{W}' = \mathbf{W} + \mathbf{S}_{cos}$, where $\mathbf{S}_{cos}$ is the pairwise similarity matrix calculated with the cosine similarity. Therefore, HFCM (word+link), NERFCM (word+link) and Fk -Parts (word+link) are the results produced with each of the approaches based on a transformed vector-based data or relational data. We randomly generate 30 sets of soft partitions as initializations, and run each approach with this group of initializations. The following parameter setting of each approach is used in our experiment: $m = 1.1$ in HFCM (word), $m = 1.02$ in HFCM (word+link), $m = 1.01$ in NERFCM (word) and NERFCM (word+link), $T_u = 0.01$, $T_v = 1$ in Fk -Parts (word) and Fk -Parts (word+link), $\alpha = 1$ in LinkFCMm and LinkFCMq, and $\eta = 0.1$ in LinkFCMq.

The clustering results are evaluated by *Accuracy* and *Normalized Mutual Information (NMI)* as defined in Chapter 2.6. Table 4.2 shows the means (%) and standard deviations (%) of *Accuracy* and *NMI* values over 30 trials. From this table, it is shown that in a majority of cases, LinkFCMm and LinkFCMq perform much better than the other three approaches. Comparing two variants of LinkFCM, the quadratic-regularization based LinkFCMq overall performs better than LinkFCMm. It is shown that LinkFCMq achieves the best results on three out of five data sets, and produces comparable good results as LinkFCMm on PL. It is also observed that in most of the cases, HFCM and *Fk*-Parts produce better results with the combination data than only based on vector-based data for HFCM or only based on relational data for *Fk*-Parts. Unlike HFCM and *Fk*-Parts, the results of NERFCM on the combined data degrade to a very low level. We found that on the combined data, even the fuzzifier is tuned to a very small value, the output membership of NERFCM is still over fuzzy. Further data scaling and normalization may be required for NERFCM to produce reasonable results on the combined data. Compared to HFCM and NERFCM, the performance of our previously proposed approach *Fk*-Parts is shown to be much better in most of the cases.

Table 4.3 shows the *t-test* results of the four best algorithms LinkFCMq, LinkFCMm, HFCM (word+link) and *Fk*-Parts (word+link), where the last two are denoted as HFCM and *Fk*-Parts for simplicity. When the t-value is larger than $t_{critical} = 2.002$, we use “ $\gg$ ” to show that the result on the left is significantly better than the one on the right, and if a t-value is positive but smaller than $t_{critical}$, we use “ $>$ ” to show that the improvement of the left compared to the right is not significant. From this table, it is shown that the improvements made by LinkFCMq with respect to HFCM are significant in all the cases in terms of both *Accuracy* and *NMI*. LinkFCMq is significantly better than *Fk*-Parts in all the cases except on ML, where *Fk*-Parts is the best. LinkFCMm also achieves significant improvements compared with HFCM for most of the cases. The performance of LinkFCMm is also much better than *Fk*-Parts on DS and PL, but comparable or worse on the others.

These results indicate that the formulation of the proposed LinkFCM enables the content information and the citation information to be used more effectively than

Table 4.2: Performance comparison of fuzzy approaches using Cora data

Accuracy					
	DS	HA	ML	OS	PL
HFCM (word)	31.37 $\pm$ 0.68	37.40 $\pm$ 2.71	28.15 $\pm$ 2.39	57.43 $\pm$ 2.05	27.37 $\pm$ 0.56
HFCM (link + word)	36.64 $\pm$ 3.15	37.69 $\pm$ 3.64	43.34 $\pm$ 5.29	45.59 $\pm$ 3.46	31.89 $\pm$ 2.35
NERFCM (link)	28.86 $\pm$ 0.68	38.10 $\pm$ 2.71	38.82 $\pm$ 2.39	56.82 $\pm$ 2.05	26.74 $\pm$ 0.56
NERFCM (link + word)	26.09 $\pm$ 0.02	34.65 $\pm$ 1.97	24.80 $\pm$ 1.11	57.22 $\pm$ 0.00	26.73 $\pm$ 0.00
Fk-Parts (link)	30.22 $\pm$ 2.10	38.29 $\pm$ 2.60	46.54 $\pm$ 3.27	50.54 $\pm$ 3.86	32.46 $\pm$ 2.40
Fk-Parts (link + word)	40.85 $\pm$ 1.75	40.13 $\pm$ 1.99	68.75 $\pm$ 2.22	58.94 $\pm$ 0.37	35.72 $\pm$ 0.69
LinkFCMm	42.48 $\pm$ 2.70	42.39 $\pm$ 6.05	48.01 $\pm$ 6.73	59.58 $\pm$ 3.05	41.01 $\pm$ 3.64
LinkFCMq	42.72 $\pm$ 3.70	47.92 $\pm$ 3.62	58.03 $\pm$ 6.42	64.75 $\pm$ 5.43	40.53 $\pm$ 3.03

NMI					
	DS	HA	ML	OS	PL
HFCM (word)	9.83 $\pm$ 0.52	18.73 $\pm$ 1.09	11.35 $\pm$ 1.61	17.31 $\pm$ 1.15	1.67 $\pm$ 0.16
HFCM (link+word)	25.53 $\pm$ 2.16	32.90 $\pm$ 3.57	28.89 $\pm$ 3.42	25.74 $\pm$ 2.53	23.58 $\pm$ 1.79
NERFCM (link)	13.05 $\pm$ 3.83	28.52 $\pm$ 3.65	21.71 $\pm$ 3.89	24.78 $\pm$ 0.00	13.00 $\pm$ 3.97
NERFCM (link+word)	4.69 $\pm$ 0.04	27.24 $\pm$ 1.67	5.42 $\pm$ 1.43	16.12 $\pm$ 0.00	2.23 $\pm$ 0.00
Fk-Parts (link)	21.21 $\pm$ 1.99	30.54 $\pm$ 3.35	30.06 $\pm$ 3.11	24.33 $\pm$ 3.73	23.85 $\pm$ 2.00
Fk-Parts (link+word)	31.60 $\pm$ 0.93	32.76 $\pm$ 2.11	49.44 $\pm$ 1.57	30.30 $\pm$ 0.32	22.86 $\pm$ 0.19
LinkFCMm	34.22 $\pm$ 1.97	30.27 $\pm$ 3.47	38.58 $\pm$ 5.25	26.41 $\pm$ 2.46	30.15 $\pm$ 2.46
LinkFCMq	36.57 $\pm$ 2.20	38.21 $\pm$ 2.90	45.85 $\pm$ 3.98	30.69 $\pm$ 4.40	28.5 $\pm$ 2.58

vector-represented based approach or similarity-based approach with pre-combined data. It is also observed that for document datasets, quadratic-regularization is a better choice than the fuzzifier to generate fuzziness in memberships.

4.6.2.2 User-provided Relation

Now we test the performance of LinkFCM on Cora data with user-provided relation. First, we show the results of LinkFCMm and LinkFCMq on the OS dataset with the relation matrix $\mathbf{W} = \mathbf{W}_c$ records constraints obtained in the same way as stated in Section 4.6.1.2. We select different numbers of pairwise constraints y , and for each y , run LinkFCMm and LinkFCMq 30 times with random samplings of constraints. The total possible pairs of objects for 1246 objects in OS is 775635. The largest number of pairwise constraints used in our experiment is 7000, which accounts for 1.0% of the total. We set relative small values for α when only y is small, and increase it when y is increasing. This is because with an increasing number of pairwise constraints, the information provided by the relational data becomes more and more reliable. Specifically, the following α values are used in LinkFCMm and LinkFCMq, respectively:

Table 4.3: Significance test of improvement of LinkFCM compared to HFCM and Fk-parts. The results of HFCM and Fk-Parts are based on word+link

Accuracy	
DS	LinkFCMq > LinkFCMm >> Fk-Parts > HFCM
HA	LinkFCMq >> LinkFCMm > Fk-Parts >> HFCM
ML	Fk-Parts >> LinkFCMq >> LinkFCMm >> HFCM
OS	LinkFCMq >> LinkFCMm > Fk-Parts >> HFCM
PL	LinkFCMm > LinkFCMq >> Fk-Parts >> HFCM
NMI	
DS	LinkFCMq >> LinkFCMm >> Fk-Parts >> HFCM
HA	LinkFCMq >> HFCM > Fk-Parts >> LinkFCMm
ML	Fk-Parts >> LinkFCMq >> LinkFCMm >> HFCM
OS	LinkFCMq > Fk-Parts >> LinkFCMm > HFCM
PL	LinkFCMm >> LinkFCMq >> HFCM >> Fk-Parts

y	1000	2000	3000	4000	5000	6000	7000
LinkFCMm	0.1	0.1	3	3	3	5	5
LinkFCMq	0.1	3	5	5	5	7	7

Figure 4.6 shows the errorbar plot of the *Accuracy* values with respect to y . From this figure, it is shown that as expected, the *Accuracy* values of both LinkFCMm and LinkFCMq increase when y is increasing. The most significant improvement is observed when y increases from 1000 to 3000. For $y = [2000, 4000]$, LinkFCMq performs slightly better than LinkFCMm. With $y = 5000$, LinkFCMm and LinkFCMq already achieve an *Accuracy* of 96%. Comparing the results of this experiment with those obtained in the previous one in Table 4.2, it is observed that the performances of LinkFCMm and LinkFCMq with $y = 2000$ are comparative to their results obtained using the citation information. There are 4686 pairwise links in the citation-based relation matrix. This implies that compared with citation information, constraints generated according to the ground truth classes are more accurate and informative. However, in real applications, it is expensive or difficult to obtain such kind of domain knowledge. Therefore, compared to the semi-supervised clustering, the unsupervised clustering with data-derived relation is more practical.

Table 4.4: Comparison of *Accuracy* of semi-supervised learning for Cora data

Approaches	DS	HA	ML	OS	PL
Lap-Reg (content)	23.03 $\pm$ 1.94	24.09 $\pm$ 2.37	23.78 $\pm$ 2.76	43.82 $\pm$ 1.08	24.14 $\pm$ 5.27
Fk-Parts (content)	37.40 $\pm$ 1.28	58.90 $\pm$ 5.33	50.56 $\pm$ 1.84	54.87 $\pm$ 0.83	37.97 $\pm$ 1.17
LinkFCMq (content)	38.43 $\pm$ 2.49	70.65 $\pm$ 4.78	41.57 $\pm$ 3.50	72.42 $\pm$ 4.44	28.39 $\pm$ 1.68
Lap-Reg (content+link)	32.52 $\pm$ 1.85	52.39 $\pm$ 2.21	40.73 $\pm$ 1.09	51.42 $\pm$ 0.89	36.16 $\pm$ 1.36
Fk-Parts (content+link)	43.18 $\pm$ 1.43	58.83 $\pm$ 6.53	62.33 $\pm$ 1.65	63.30 $\pm$ 0.84	40.24 $\pm$ 0.61
MVLL-Reg (content+link)	45.00 $\pm$ 2.13	58.27 $\pm$ 2.40	54.26 $\pm$ 1.44	61.32 $\pm$ 2.73	45.84 $\pm$ 1.57
LinkFCMq (content+link)	68.39 $\pm$ 5.09	84.39 $\pm$ 3.61	77.33 $\pm$ 2.71	89.45 $\pm$ 1.54	59.94 $\pm$ 2.40

Figure 4.6: Clustering accuracy of LinkFCMm and LinkFCMq with different numbers of pairwise constraints.

Finally, we compare the results of LinkFCMq with three other approaches listed as below:

- Lap-Reg [129]: Local learning with Laplacian regularization.
- MVLL-Reg [129]: Local learning with Multi-view regularization.
- Fk-Parts: the weighted-medoids based fuzzy clustering proposed in Chapter 3 which works on a similarity matrix.

The Lap-Reg and MVLL-Reg approaches are semi-supervised local learning with Laplacian regularization matrix and Multi-View regularization matrix, respectively. Let $\mathbf{X}$ denotes the data matrix recoding the document-word co-occurrences and $\mathbf{W}_d$ is the link-based relation matrix recording the citation among documents. The results of Lap-Reg and MVLL-Reg shown here are quoted from [129], which are obtained by training with 30% labeled data. Two groups of results namely Lap-Reg (content) and Lap-Reg (content+link) are reported for Lap-Reg, where the former refers to the results generated by Lap-Reg based on $\mathbf{X}$, i.e., the content or words contained in each

document, and the later based on both $\mathbf{X}$ and $\mathbf{W}_d$, i.e., the combined content and citation information. In MVLL-Reg, the two types of data namely content and link, are treated as two views of the dataset, and a regularization matrix derived with these two views are used in MVLL-Reg. To compare with them, we also produce two groups of results for *Fk*-Parts and LinkFCMq with one only based on content information and the other based on both content and link information. Instead of using labeled data as supervision, 2000 pairwise constraints recorded with relation matrix $\mathbf{W}_c$, are used as supervision in *Fk*-Parts and LinkFCMq. Specifically, *Fk*-Parts (content) and *Fk*-Parts (content+link) denote the results of *Fk*-Parts with the $\mathbf{W}' = \mathbf{W}_c + \mathbf{X}^T \mathbf{X}$ and $\mathbf{W}' = \mathbf{W}_c + \mathbf{W}_d + \mathbf{X}^T \mathbf{X}$, respectively; LinkFCMq (content) and LinkFCMq (content+link) denote the results of LinkFCMq with $\mathbf{W} = \mathbf{W}_c$ and $\mathbf{W} = \mathbf{W}_c + \mathbf{W}_d$, respectively.

Table 4.4 and the figure below show the mean and standard deviation of the classification accuracy of Lap-Reg and MVLL-Reg, and the clustering accuracy of *Fk*-Parts and LinkFCMq. From this table, we have the following observations: LinkFCMq gives the best results with content+link relation on all the five datasets. Even only with content, the constrained LinkFCMq gives the second best results on HA and OS, which are better than other approaches based on both content and link. *Fk*-Parts (content+link) is comparable with MVLL-Reg (content+link) on DA, HA and OS, and better on PL. Lap-Reg performs worse than others. Compared with those content-abased results, those content+link based results of the same algorithm are usually better, i.e., Lap-Reg (content+link), *Fk*-Parts (content+link), and LinkFCMq (content+link) are better than Lap-Reg (content), *Fk*-Parts (content), and LinkFCMq (content), respectively, except that *Fk*-Parts (content+link) and *Fk*-Parts (content) are comparable on HA. These observations show the benefit of using both content and link information in document clustering. Moreover, compared to other approaches, the formulation of the proposed LinkFCM approach provides it a more effective way to make use these two types of information.

4.7 Summary

In this Chapter, we studied the clustering problem of document datasets, where other than the vector representation of the content of each document, document-document

relation is also available. To handle this problem, we consider to incorporate the relational information into a fuzzy c-mean type clustering so that both vector-based data and relational data are considered in the clustering process. The main idea is that we define a new term to measure the total intra-cluster connectivity in terms of pairwise relationships, and then combine it with another term which measures the total intra-cluster distance with vector-based data. Two versions of LinkFCM referred as LinkFCM_m and LinkFCM_q are proposed, where the fuzziness of memberships is produced with fuzzifier m and quadratic regularization, respectively. We also showed the connection between LinkFCM and constrained fuzzy c-means when the relation matrix is constructed with pairwise constraints. Experimental studies on simulated 2-D data and a set of five real-world document datasets have been conducted. For each dataset, we tested LinkFCM with relational data obtained in different ways. Our experimental results show that compared with other approaches, the formulation of LinkFCM enables it to make use of the content information and link information more effectively, and hence to produce clusters with better qualities.

Chapter 5

Fuzzy Approach for Multi-type Relational Data Clustering

5.1 Overview

In the previous two chapters, we have proposed two approaches *Fk*-Parts and LinkFCM for document clustering. *Fk*-Parts works with a document-document relation matrix, and LinkFCM is formulated to make use of the vector representation of documents together with document-document relation for document clustering. In some applications, the document dataset to be clustered may be described by several types of data from different angles. Other than document, multiple other types of objects may be involved. We have discussed in Chapter 2.5 that such kind of document data is more naturally to be described as multi-type relational data, which is represented by a set of relation matrices. For example, for a dataset of published research papers, other than knowing a set of words each paper contains, we may also know the authors of each paper, the names of conference or journal where those papers are published, and the references of each paper. This dataset may be referred as a four-type relational data, which consists of *paper*, *term*, *author*, *venue* (e.g., conference, journal) these four types of objects or entities, together with four relations: *paper-term* relation, *paper-author* relation, *paper-venue* relation and *paper-paper* relation. The first three relations characterize each individual paper with respect to different aspects, namely the content, the person(s) who wrote the paper, and the place where the paper is published, respectively; while the last relation records the citation information

of one paper to another paper. Multi-type relational data may form different structures depending on which relations are available. A star-structure is a special case where relations only exist between the central type, i.e., document, and several attribute types. For example, the research paper data forms a four-type star-structured relational data with the *paper-term*, *paper-author*, and *paper-venue* relations, where *paper* is the central type, and *term*, *author*, and *venue* are three attribute types. It is not a star-structure any more when the *paper-paper* relation is considered.

In this chapter, we propose a fuzzy clustering approach of multi-type relational data referred as FC-MR for document categorization and analysis. This approach simultaneously clusters documents and each of other involved types of objects. In FC-MR, each object is associated with a membership and a ranking with respect to each of the clusters. The membership is the soft assignment or soft cluster label, while the ranking is the weight of representativeness. An object is assigned a large membership with respect to a cluster if its related objects have high rankings in that cluster. In each cluster, an object tends to have a high ranking if its related objects have large memberships in that cluster. The FC-MR approach is formulated to deal with multi-type relational data with various structures so that both document-document relation and relations between document and other types of objects are considered for document clustering. The objective function of FC-MR is locally optimized by an efficient iterative algorithm which updates the fuzzy membership matrix and the ranking matrix of one type of objects at once, while keeping those of the other types of objects constant. We also discuss the simplified FC-MR for multi-type relational data with two special structures namely star-structure and extended star-structure, and establish connections between FC-MR and two other fuzzy approaches to show that the fuzzy relational clustering approach *Fk*-Parts proposed in Chapter 3 and the fuzzy co-clustering approach FCoDok [85] are two special cases of FC-MR. Experimental studies are conducted on benchmark document datasets to illustrate how the proposed approach can be applied flexibly under different scenarios in real-world applications. The experimental results demonstrate the feasibility and effectiveness of the new approach compared with existing ones.

5.2 Motivation

In Chapter 3, we have proposed a weighted-medoid based clustering approach Fk -Parts for document clustering based on the document-document relation. Since in some cases, both vector representation of the content of each document and document-document relation are available for a document dataset, we proposed LinkFCM in Chapter 4 to consider relational data together with vector-based data for document clustering. In our experimental studies, LinkFCM is shown to perform much better in document clustering than other existing approaches. Nevertheless, there is still room in improvement in document clustering. First, LinkFCM still relies on the vector-space model to encoding the content of each document, the difficulty of clustering in a sparse and high-dimensional space remains. Second, other than the content and link information, there may be more other types of data that can be used for the document clustering purpose. Third, both Fk -Parts and LinkFCM only produce document clusters, but no document ranking information or clusters of other related types, e.g., clusters of words, are produced at the end of clustering. Therefore, a new approach which is able to simultaneously take care of the above three concerns would be a more favorable choice for document clustering.

In the literature, co-clustering has been proposed for handling high-dimensional data. Given the document-word co-occurrence matrix, a co-clustering approach treats *document* and *word* as two object types, and produce document clusters and word clusters simultaneously. The documents are clustered based on the word clusters and the words are clustered based on the document clusters. Compared to conventional one-side clustering, co-clustering of documents and words provides a dynamic feature selection process, which leads to mutual improvement of document clusters and word clusters. Other than that, at the end of clustering, the word clusters can be used as descriptive information for explanation and analysis of their associated document clusters. Therefore, co-clustering provides possible solutions to the first and third concerns we discussed above, i.e., the high-dimensional problem and the descriptive information for interpretation of document clusters produced. However, co-clustering itself is unable to make use of multiple types of data as it only operate on the co-occurrence matrix between documents and one other types of objects, which

is typically the words. It is possible to use some data transformation process to combine multiple types of data into one data matrix, so that existing approaches can be applied. However, information which may be helpful for the clustering purpose is lost during the transformation process. Moreover, clustering documents with combined data is unable to capture the interrelated patterns among different types of objects which may be interested in some applications.

Instead of working on data transformation techniques, we are going to present a fuzzy clustering of multi-type relational data for document clustering. This approach, called FC-MR, could be regarded as a generalized fuzzy co-clustering. Since FC-MR is formulated to handle multi-type relational data of various structures, it is able to consider relations between document and other types of objects as well as pairwise document relation when applied for document clustering. Under the framework of fuzzy clustering, the proposed approach maintains computational simplicity.

5.3 Fuzzy Clustering for Multi-type relational Data

Now we present the details of the proposed approach for clustering relational data involving several types of objects.

5.3.1 Problem Formulation

We denote a dataset with m object types as $\mathcal{X} = \{\mathcal{X}_\mu\}_{\mu=1}^m$, where $\mathcal{X}_\mu = \{x_i^\mu\}_{i=1}^{n_\mu}$ is a set of objects of the μ th type and $n_\mu = |\mathcal{X}_\mu|$. The relation between $\mathcal{X}_\mu$ and $\mathcal{X}_\nu$ denoted as $\mathcal{X}_\mu \sim \mathcal{X}_\nu$ is stored in matrix $\mathbf{R}_{\mu\nu}$. The relation is between two different types if $\mu \neq \nu$ or is between objects of the same type if $\mu = \nu$. We assume $\mathbf{R}_{\mu\mu} = \mathbf{R}_{\mu\mu}^T$. Each entry $r_{ij}^{\mu\nu} \in \mathbf{R}_{\mu\nu}$ records the value of relationship between object x_i^μ and x_j^ν . Each relation matrix $\mathbf{R}_{\mu\nu}$ is associated with a weight $\beta_{\mu\nu}$. The clustering problem now is to partition the dataset $\mathcal{X}$ into k clusters, i.e., $\mathcal{X} = \{\bigcup \mathcal{C}_f\}_{f=1}^k$, by making use of all relations effectively and collaboratively. Each cluster $\mathcal{C}_f$ is a union of subsets of objects of each of the types, i.e., $\mathcal{C}_f = \{\bigcup \mathcal{C}_f^\mu\}_{\mu=1}^m$, where $\mathcal{C}_f^\mu$ represents the subset of objects of the μ th type in the f th cluster. This clustering problem is equivalent to be described as to simultaneously produce a partitioning of each type of objects

$\mathcal{X}_\mu = \{\bigcup_{f=1}^k \mathcal{C}_f^\mu\}$, and at the same time to establish one-to-one associations among the clusters of different types.

To formulate this problem, for each type $\mu = \{1, 2, \dots, m\}$, we define a cluster membership u_{if}^μ and a ranking v_{if}^μ for each object x_i^μ with respect to each cluster f . The membership is a soft or fuzzy cluster indicator that measures how possible an object is labeled as that cluster, and the ranking measures how representative or typical x_i^μ is compared with other objects in $\mathcal{X}_\mu$ in cluster f . Assuming the number of clusters to be produced is k , then each type $\mathcal{X}_\mu$ is associated with two $n_\mu \times k$ matrices $\mathbf{U}_\mu$ and $\mathbf{V}_\mu$. To be more clear, we write out the membership matrix and its transpose in terms of their column vectors as

$$\mathbf{U}_\mu = (\mathbf{u}_1^\mu, \mathbf{u}_2^\mu, \dots, \mathbf{u}_k^\mu), \quad \mathbf{U}_\mu^T = (\mathbf{u}'_1^\mu, \mathbf{u}'_2^\mu, \dots, \mathbf{u}'_{n_\mu}^\mu). \quad (5.1)$$

Each column of $\mathbf{U}_\mu$, i.e. for $f = 1, 2, \dots, k$, $(\mathbf{u}_f^\mu)_{n_\mu \times 1}$ records the memberships of n_μ objects in $\mathcal{X}_\mu$ in the f th cluster, and each row of $\mathbf{U}_\mu$, i.e. for $i = 1, 2, \dots, n_\mu$, $(\mathbf{u}'_i^\mu)_{1 \times k}^T$ records the memberships of object x_i^μ in all the k clusters, where $\mathbf{u}'_i^\mu$ is the column vector of $\mathbf{U}_\mu^T$. Denotations of $\mathbf{V}_\mu$ are defined similarly.

It is intuitive that an object is more likely to be assigned into a cluster where many of its related objects are ranked high in that cluster. Within a cluster, an object is more likely to be a representative object if it is related to many of the objects assigned in that cluster. For an object x_i^μ , the following term

$$R_{if}^\mu = \sum_{\nu=1}^m \sum_{j=1}^{n_\nu} r_{ji}^{\nu\mu} v_{jf}^\nu \quad (5.2)$$

gives the sum of rankings of each of its related objects of all the m types in cluster f , weighted by the strength of relationships, and

$$M_{if}^\mu = \sum_{\nu=1}^m \sum_{j=1}^{n_\nu} r_{ji}^{\nu\mu} u_{jf}^\nu \quad (5.3)$$

calculates the sum of memberships of each of its related objects in cluster f weighted by the strength of relationships. We may expect that the membership of x_i^μ is larger in cluster f than in cluster c if $R_{if}^\mu > R_{ic}^\mu$, and within a cluster f , the ranking of x_i^μ

is larger than that of x_j^μ if $M_{if}^\mu > M_{jf}^\mu$. Based on this idea, we now formulate the proposed FC-MR as below:

$$\max J = Tr\left(\sum_{\mu=1}^m \sum_{\nu=1}^m \beta_{\mu\nu} \mathbf{U}_\mu^T \mathbf{R}_{\mu\nu} \mathbf{V}_\nu\right) - \sum_{\mu=1}^m \frac{\phi_\mu}{2} \|\mathbf{U}_\mu\|_F^2 - \sum_{\nu=1}^m \frac{\theta_\nu}{2} \|\mathbf{V}_\nu\|_F^2 \quad (5.4)$$

subject to

$$\mathbf{U}_\mu \mathbf{1} = \mathbf{1}, \mathbf{V}_\nu^T \mathbf{1} = \mathbf{1}, \mathbf{U}_\mu, \mathbf{V}_\nu \geq 0 \quad \text{for } \mu, \nu = 1, 2, \dots, m. \quad (5.5)$$

where $\|\mathbf{A}\|_F^2 = Tr(\mathbf{A}^T \mathbf{A})$ denotes the squared Frobenius norm of matrix $\mathbf{A}$.

The formulation of the clustering problem into the above maximization problem consists of three components: quality measure of a partitioning, constraints and regularization. Now we give our analytical study to see how this formulation matches well with our intuitive idea. Without loss of generality, $\forall \mu, \nu$, we assume $\beta_{\mu\nu} = 1$. We use $\mathbf{r}_i^{\nu\mu} = [r_{1i}^{\nu\mu}, r_{2i}^{\nu\mu}, \dots, r_{n_\nu i}^{\nu\mu}]^T$ to denote the i th column of $\mathbf{R}_{\nu\mu}$. Now we rewrite the first term of (5.4) as below by writing out the trace expression:

$$Tr\left(\sum_{\mu=1}^m \sum_{\nu=1}^m \mathbf{U}_\mu^T \mathbf{R}_{\mu\nu} \mathbf{V}_\nu\right) = \sum_{\mu=1}^m \sum_{i=1}^{n_\mu} \sum_{\nu=1}^m \mathbf{u}_i'^{\mu T} \mathbf{V}_\nu^T \mathbf{r}_i^{\nu\mu} \quad (5.6)$$

where $\mathbf{u}_i'^\mu = [u_{i1}^\mu, u_{i2}^\mu, \dots, u_{ik}^\mu]^T$. In order to maximize the value of the above equation, for each μ, i , the following value needs to be maximized:

$$\sum_{\nu=1}^m \mathbf{u}_i'^{\mu T} \mathbf{V}_\nu^T \mathbf{r}_i^{\nu\mu} = \sum_{c=1}^k \sum_{\nu=1}^m u_{ic}^\mu \mathbf{V}_c^{\nu T} \mathbf{r}_i^{\nu\mu} = \sum_{c=1}^k u_{ic}^\mu R_{ic}^\mu \quad (5.7)$$

where R_{ic}^μ is defined in (5.2). With constraints that $\forall i, \mu, \mathbf{1}^T \mathbf{u}_i'^\mu = 1$, maximizing (5.7) tends to assign x_i^μ a large membership in cluster f if R_{if}^μ is large. This rule is consistent with our idea discussed early about how the memberships of different clusters should be assigned to an object. To discuss how the rankings are distributed in a cluster among objects of each type when maximizing the objective function (5.4), we change (5.6) into another form as below:

$$Tr\left(\sum_{\mu=1}^m \sum_{\nu=1}^m \mathbf{U}_\mu^T \mathbf{R}_{\mu\nu} \mathbf{V}_\nu\right) = \sum_{c=1}^k \sum_{\mu=1}^m \sum_{\nu=1}^m \mathbf{u}_c^{\mu T} \mathbf{R}_{\mu\nu} \mathbf{v}_c^\nu \quad (5.8)$$

We have

$$\sum_{\mu=1}^m \sum_{\nu=1}^m \mathbf{u}_c^{\mu T} \mathbf{R}_{\mu\nu} \mathbf{v}_c^\nu = \sum_{\nu=1}^m \sum_{j=1}^{n_\nu} v_{jc}^\nu \sum_{\mu=1}^m \mathbf{u}_c^{\mu T} \mathbf{r}_j^{\mu\nu} = \sum_{\nu=1}^m \sum_{j=1}^{n_\nu} v_{jc}^\nu M_{jc}^\nu \quad (5.9)$$

where M_{jc}^ν is defined in (5.3). Maximizing (5.8) is to maximize the value of (5.9) for each $c = \{1, 2, \dots, k\}$ under the constraints that $\forall \nu, c, \mathbf{1}^T \mathbf{v}_c^\nu = 1$. This requires the values of $\mathbf{v}_c^\nu = (v_{1c}^\nu, v_{2c}^\nu, \dots, v_{n_\nu c}^\nu)^T$ to be distributed in the way that object x_p^ν has a large ranking value v_{pc}^ν if M_{pc}^ν is large. This possible solution follows our general idea on how objects of the same type should be ranked in a cluster.

We just discussed that the first term of the objective function in (5.4) can be used to measure the quality of clusters, and maximizing this term requires memberships and rankings to be distributed in the way that matches well with our basic idea. The other terms in (5.4) calculate the squared Frobenius norm of each of the membership matrices and ranking matrices. Those terms are used as regularization for preventing each object from being only assigned to a single cluster f , with $f = \arg \max_c R_{ic}^\mu$, or only one object x_p^μ of each type having non-zero ranking value in a cluster c , where $p = \arg \max_j M_{jc}^\mu$. Parameters ϕ_μ and θ_ν tradeoff the contribution between the first term and the regularization terms. We can write the Frobenius norm of a membership matrix and a ranking matrix as the sum of l_2 norm of rows and columns, respectively, i.e., $\|\mathbf{U}_\mu\|_F = \sum_{i=1}^{n_\mu} \|\mathbf{u}_i^\mu\|_2$, and $\|\mathbf{V}_\mu\|_F = \sum_{c=1}^k \|\mathbf{v}_c^\mu\|_2$. Since each $\|\mathbf{u}_i^\mu\|_2$ and $\|\mathbf{v}_c^\mu\|_2$ are minimized when all elements are equal, i.e., $\forall c, u_{ic}^\mu = \frac{1}{k}$ and $\forall j, v_{jc}^\mu = \frac{1}{n_\mu}$, the larger ϕ_μ is, generally the smoother the memberships of each object of type μ distributed over the k clusters; and the larger θ_μ is, the smoother the rankings are distributed over all objects of type μ in each cluster. In fuzzy clustering, the smoothness of membership distribution over all the clusters is called fuzziness. Such kind of regularization referred as quadratic regularization has been used in our approaches Fk-Parts and LinkFCM proposed in previous two chapters, as well as several fuzzy approaches, such as [50, 85].

5.3.2 Solution

We use the method of Lagrange multiplier to derive the local solutions for the constrained optimization problem formulated above. With vectors $\vec{\gamma}_\mu$, $\vec{\lambda}_\nu$, and matrices

$\alpha_{i,c}^\mu \in \mathfrak{Q}_\mu$, $\beta_{i,c}^\nu \in \mathfrak{B}_\nu$ being Lagrange multipliers, the Lagrangian is formed as

$$\begin{aligned} L = & J + \sum_{\mu=1}^m \tilde{\gamma}_\mu^T (\mathbf{U}_\mu \mathbf{1} - \mathbf{1}) + \sum_{\nu=1}^m \tilde{\lambda}_\nu^T (\mathbf{V}_\nu^T \mathbf{1} - \mathbf{1}) \\ & + \sum_{\mu=1}^m \mathbf{1}^T (\mathfrak{Q}_\mu \odot \mathbf{U}_\mu) \mathbf{1} + \sum_{\nu=1}^m \mathbf{1}^T (\mathfrak{B}_\nu \odot \mathbf{V}_\nu) \mathbf{1} \end{aligned} \quad (5.10)$$

According to the KKT (Karush-Kuhn-Tucker) conditions:

$$\frac{\partial L}{\partial u_{ic}^\mu} = 0 \quad (5.11)$$

$$\alpha_{i,c}^\mu u_{ic}^\mu = 0 \quad (5.12)$$

$$\alpha_{i,c}^\mu \geq 0 \quad (5.13)$$

after some algebraic manipulations, the memberships of object x_i^μ in the k clusters can be derived as

$$\mathbf{u}'_i^\mu = \frac{1}{\phi_\mu} \mathbf{g}_i^\mu + \frac{1}{|\mathcal{K}_i^{\mu+}|} \left(1 - \frac{1}{\phi_\mu} \mathbf{e}_i^T \mathbf{g}_i^\mu\right) \mathbf{1} \quad (5.14)$$

with

$$\mathbf{g}_i^\mu = \sum_{\nu=1}^m \beta_{\mu\nu} \mathbf{V}_\nu^T \mathbf{r}_i^{\nu\mu} = [g_{i1}^\mu, g_{i2}^\mu, \dots, g_{ik}^\mu]^T \quad (5.15)$$

$$\mathbf{e}_i^T = [e_{i1}, e_{i2}, \dots, e_{ik}], \quad \text{and} \quad (5.16)$$

$$e_{if} = \begin{cases} 1 & \text{for } f \in \mathcal{K}_i^{\mu+} \\ 0 & \text{others} \end{cases} \quad (5.17)$$

where

$$\mathcal{K}_i^{\mu+} = \{f : u_{if}^\mu > 0\}.$$

Similarly, the rankings of all objects of the μ th type in cluster c is

$$\mathbf{v}_c^\mu = \frac{1}{\theta_\mu} \mathbf{h}_c^\mu + \frac{1}{|\mathcal{N}_c^{\mu+}|} \left(1 - \frac{1}{\theta_\mu} \mathbf{e}_c^T \mathbf{h}_c^\mu\right) \mathbf{1} \quad (5.18)$$

with

$$\mathbf{h}_c^\mu = \sum_{\nu=1}^m \beta_{\mu\nu} \mathbf{R}_{\mu\nu} \mathbf{u}_c^\nu = [h_{1c}^\mu, h_{1c}^\mu, \dots, h_{n_\mu c}^\mu]^T \quad (5.19)$$

$$\mathbf{e}_c^T = [e'_{1c}, e'_{2c}, \dots, e'_{n_\mu c}], \quad \text{and} \quad (5.20)$$

$$e'_{jc} = \begin{cases} 1 & \text{for } j \in \mathcal{N}_c^{\mu+} \\ 0 & \text{others} \end{cases} \quad (5.21)$$

where

$$\mathcal{N}_c^{\mu+} = \{l : v_{lc}^\mu > 0\}$$

With $\mathbf{U}_\mu = (\mathbf{u}_1^\mu, \mathbf{u}_2^\mu, \dots, \mathbf{u}_{n_\mu}^\mu)^T$ and $\mathbf{V}_\mu = (\mathbf{v}_1^\mu, \mathbf{v}_2^\mu, \dots, \mathbf{v}_k^\mu)$, we get the membership matrix and ranking matrix of objects in $\mathcal{X}_\mu$.

The first term of (5.14) decides the membership distribution of each object x_i^μ in the k clusters while the second term is a normalization term to ensure the summation constraint to be satisfied. Similarly, the first term in (5.18) decides the distribution of ranking values among objects in $\mathcal{X}_\mu$ in each cluster c , which is normalized by the second term so that the sum of ranking values of objects of the same type in each cluster is 1.

The last problem left is to decide $\mathcal{K}_i^{\mu+}$ and $\mathcal{N}_c^{\mu+}$. According to the discussions in [50] and Chapter 3, it can be proved that if $c \in \mathcal{K}_i^{\mu+}$, then $\{\forall f \in \mathcal{K}_i^{\mu+} | f : g_{if}^\mu > g_{ic}^\mu\}$ and if $j \in \mathcal{N}_c^{\mu+}$, then $\{\forall p \in \mathcal{N}_c^{\mu+} | p : h_{pc}^\mu > h_{jc}^\mu\}$. Based on this, $\mathcal{K}_i^{\mu+}$ and $\mathcal{N}_c^{\mu+}$ can be calculated in an incremental way which is similar to Procedure-K and Procedure-N given in Chapter 3.3.3.

5.3.3 Algorithm

Given relation matrices among m types, and the membership matrices at iteration l for each of the types, i.e., $\{\mathbf{U}_\mu^{(l)}\}_{\mu=1}^m$, ranking values for each type $\{\mathbf{V}_\mu^{(l)}\}_{\mu=1}^m$ are updated with the current memberships. Based on $\{\mathbf{V}_\mu^{(l)}\}_{\mu=1}^m$, the memberships are re-estimated to get a set of updated $\{\mathbf{U}_\mu^{(l+1)}\}_{\mu=1}^m$. This alternating iteration continues until convergence or reaching the maximum number of iterations specified by the user. For a more efficient implementation of the algorithm, we use the following simplified procedure to update $\mathbf{U}_\mu$ and $\mathbf{V}_\mu$ in a matrix form, which works well in practice.

- Updating $\mathbf{U}_\mu$: calculate

$$\mathbf{U}_\mu = \frac{1}{\phi_\mu} \mathbf{G}_\mu + \frac{1}{k} \left(\mathbb{1} - \frac{1}{\phi_\mu} \mathbb{1} \mathbf{G}_\mu \right) \quad (5.22)$$

with

$$\mathbf{G}_\mu = \sum_{\nu=1}^m \beta_{\mu\nu} \mathbf{R}_{\mu\nu} \mathbf{V}_\nu \quad (5.23)$$

For each row i of $\mathbf{U}_\mu$, find

$$\mathcal{K}_i^{\mu+} = \{f : u_{if}^\mu > 0\} \quad (5.24)$$

modify

$$u_{ic}^\mu = \begin{cases} 0, & \text{for } c \notin \mathcal{K}_i^{\mu+} \\ \frac{u_{ic}^\mu}{\sum_{f \in \mathcal{K}_i^{\mu+}} u_{if}^\mu}, & \text{for } f \in \mathcal{K}_i^{\mu+} \end{cases} \quad (5.25)$$

- Updating $\mathbf{V}_\mu$: calculate

$$\mathbf{V}_\mu = \frac{1}{\theta_\mu} \mathbf{H}_\mu + \frac{1}{n_\mu} (\mathbb{1} - \frac{1}{\theta_\mu} \mathbb{1} \mathbf{H}_\mu) \quad (5.26)$$

with

$$\mathbf{H}_\mu = \sum_{\nu=1}^m \beta_{\mu\nu} \mathbf{R}_{\mu\nu} \mathbf{U}_\nu \quad (5.27)$$

For each column c of $\mathbf{V}_\mu$, find

$$\mathcal{N}_c^{\mu+} = \{p : v_{pc}^\mu > 0\} \quad (5.28)$$

modify

$$v_{jc}^\mu = \begin{cases} 0, & \text{for } j \notin \mathcal{N}_c^{\mu+} \\ \frac{v_{jc}^\mu}{\sum_{p \in \mathcal{N}_c^{\mu+}} v_{pc}^\mu}, & \text{for } j \in \mathcal{N}_c^{\mu+} \end{cases} \quad (5.29)$$

The complete procedure of the FC-MR algorithm is given in Algorithm 6. The most costly step of updating $\mathbf{U}_\mu$ (or $\mathbf{V}_\mu$) is the calculation of $\mathbf{G}_\mu$ in (5.23) (or $\mathbf{H}_\mu$ in (5.27)). If for each μ , relation matrix $\mathbf{R}_{\mu\nu}$ exists for all ν , both of these two steps have a time complexity of $O(n_\mu n^{\max} k)$, where k is the number of clusters, and $n^{\max} = \max_\nu n_\nu$ is the largest scale of m types of object sets. When those relation matrices are sparse, this complexity reduces to $O(e_\mu^{\max} k)$, where $e_\mu^{\max} = \max_\nu e_{\mu\nu}$ and $e_{\mu\nu}$ is the number of nonzero entries of $\mathbf{R}_{\mu\nu}$. Assume that the algorithm converges after l iterations, the total time complexity of the algorithm is $O(l m n^{\max} n^{\max} k)$ for dense relation matrices or $O(l m e^{\max} k)$ for sparse case with $e^{\max} = \max_{\mu,\nu} e_{\mu\nu}$ and m is the number of object types.

Algorithm 6: FC-MR

input : $n_\mu \times n_\nu$ Relation matrix $\{\mathbf{R}_{\mu\nu}\}_{\mu,\nu=\{1,2,\dots,m\}}$, the number of clusters k ,
 parameter $\{\phi_\mu\}_{\mu=\{1,2,\dots,m\}}$ and $\{\theta_\mu\}_{\mu=\{1,2,\dots,m\}}$, positive weights
 $\{\beta_{\mu\nu}\}_{\mu,\nu=\{1,2,\dots,m\}}$.
output: Fuzzy membership $\mathbf{U}_\mu$ and ranking value $\mathbf{V}_\mu$ for each
 $\mu = \{1, 2, \dots, m\}$.
 Initialize $\{\mathbf{U}_\mu^{(0)}\}_\mu = \{1, 2, \dots, m\}$ with nonnegative values;
repeat
 for each $\mu = \{1, 2, \dots, m\}$ **do**
 Update $\mathbf{V}_\mu$ from (5.26) with $\mathbf{H}_\mu$ calculated as (5.27);
 Modify $\mathbf{V}_\mu$ according to (5.29);
 end
 for each $\mu = \{1, 2, \dots, m\}$ **do**
 Update $\mathbf{U}_\mu$ from (5.22) with $\mathbf{G}_\mu$ calculated as (5.23);
 Modify $\mathbf{U}_\mu$ according to (5.25);
 end
until convergence;

5.3.4 Parameter Setting

In FC-MR, there are sets of parameters $\{\phi_\mu\}_{\mu=\{1,2,\dots,m\}}$ and $\{\theta_\mu\}_{\mu=\{1,2,\dots,m\}}$ which control how smooth the distribution of memberships over k clusters and the distribution of rankings over n_μ objects, respectively. In practice, such kind of parameters always need to be tuned empirically due to the difficulty in developing generic parameter estimation approaches theoretically. Based on our experimental study, reasonable results can be obtained when ϕ_μ and θ_μ are set to make ϕ_μ/k and θ_μ/n_μ around the same order of magnitude. Since typically $k \ll n_\mu$, compared to θ_μ , the same degree of change in ϕ_μ always creates a larger impact on the final clustering results. This indicates that the value of ϕ_μ should be smaller than θ_μ , and ϕ_μ usually needs to be tuned with a smaller step than that of θ_μ when $k \ll n_\mu$. According to this observation, we usually begin with an initial setting as: $\theta_i = 1$ with $i = \arg \max_\mu n_\mu$, and other parameters follow $\theta_{\mu \neq i} = n_\mu/n_i$, $\phi_\mu = k/n_i$. Based on this initial setting, further adjustment may be made to obtain a proper result, e.g., increase (decrease) the value of ϕ_μ if the memberships of the μ th type are too crisp (fuzzy). We also noted that the proper values of ϕ_μ and θ_μ are usually smaller for highly overlapped datasets than for well separated ones.

The weights of each of the relation matrices $\beta_{\mu\nu}$ also make impact on the final clustering results as they decide how much of each relation contributes during the clustering process. Generally, a relation which provides complete and reliable information should be assigned a large weight while a relation which plays a less significant role or provides supplementary information should be assigned a small weight. Further study on these weights may be needed but is out of the scope of this paper.

5.4 Star-Structure and Extended Star-Structure

We have presented the FC-MR algorithm for relational data with various structures. Now, we discuss simplified FC-MR for relational data with two special structures namely star-structure and extended star-structure. Connections between FC-MR and other two fuzzy approaches will be also shown.

5.4.1 FC-MR for Star-Structured Relational Data

Star-structured relational data consists of one central type and several attribute types and only central-attribute relations are considered. For a m -type star-structured relational data, we assume $\mu = 1$ is the central type and $\mu = \{2, 3, \dots, m\}$ are $m - 1$ attribute types. The relation between the central type and the $(\mu - 1)$ th attribute type is recorded by a matrix $\mathbf{R}_{1\mu}$. The objective function of FC-MR for the star-structured relational data is reduced to

$$J_{star} = J_{star1}(\mathbf{U}_1, \{\mathbf{V}_\mu\}_{\mu=2}^m) + J_{star2}(\mathbf{V}_1, \{\mathbf{U}_\mu\}_{\mu=2}^m) \quad (5.30)$$

where

$$J_{star1} = Tr \left(\mathbf{U}_1^T \sum_{\mu=2}^m \beta_{1\mu} \mathbf{R}_{1\mu} \mathbf{V}_\mu - \frac{\phi_1}{2} \mathbf{U}_1^T \mathbf{U}_1 - \sum_{\mu=2}^m \frac{\theta_\mu}{2} \mathbf{V}_\mu^T \mathbf{V}_\mu \right) \quad (5.31)$$

$$J_{star2} = Tr \left(\sum_{\mu=2}^m \beta_{\mu 1} \mathbf{U}_\mu^T \mathbf{R}_{\mu 1} \mathbf{V}_1 - \frac{\theta_1}{2} \mathbf{V}_1^T \mathbf{V}_1 - \sum_{\mu=2}^m \frac{\phi_\mu}{2} \mathbf{U}_\mu^T \mathbf{U}_\mu \right) \quad (5.32)$$

It can be seen that J_{star1} and J_{star2} are decoupled since $\mathbf{U}_1$ and $\{\mathbf{V}_\mu\}_{\mu=2}^m$ are mutually dependent and $\mathbf{V}_1$ and $\{\mathbf{U}_\mu\}_{\mu=2}^m$ are dependent on each other. Therefore, we

can choose to only maximize J_{star1} to get $\mathbf{U}_1$ and $\{\mathbf{V}_\mu\}_{\mu=2}^m$ as in Eq. (5.22) and Eq. (5.26) with the reduced $\mathbf{G}_1$ and $\{\mathbf{H}_\mu\}_{\mu=2}^m$ given below:

$$\mathbf{G}_1 = \sum_{\mu=2}^m \beta_{1\mu} \mathbf{R}_{1\mu} \mathbf{V}_\mu \quad (5.33)$$

and

$$\mathbf{H}_\mu = \beta_{\mu 1} \mathbf{R}_{\mu 1} \mathbf{U}_1 \quad \text{for } \mu = \{2, 3, \dots, m\}, \quad (5.34)$$

or to maximize J_{star2} to obtain $\mathbf{V}_1$ and $\{\mathbf{U}_\mu\}_{\mu=2}^m$ as in Eq. (5.26) and Eq. (5.22) with $\mathbf{H}_1$ and $\{\mathbf{G}_\mu\}_{\mu=2}^m$ reduced to

$$\mathbf{H}_1 = \sum_{\mu=2}^m \beta_{1\mu} \mathbf{R}_{1\mu} \mathbf{U}_\mu \quad (5.35)$$

and

$$\mathbf{G}_\mu = \beta_{\mu 1} \mathbf{R}_{\mu 1}^T \mathbf{V}_1 \quad \text{for } \mu = \{2, 3, \dots, m\}. \quad (5.36)$$

When $m = 2$, i.e., a bi-type star-structure with one central type and one attribute type,

$$J_{star1} = Tr(\beta_{12} \mathbf{U}_1^T \mathbf{R}_{12} \mathbf{V}_2 - \frac{\phi_1}{2} \|\mathbf{U}_1\|_F^2 - \frac{\theta_2}{2} \|\mathbf{V}_2\|_F^2) \quad (5.37)$$

$$J_{star2} = Tr(\beta_{21} \mathbf{U}_2^T \mathbf{R}_{21} \mathbf{V}_1 - \frac{\phi_2}{2} \|\mathbf{U}_2\|_F^2 - \frac{\theta_1}{2} \|\mathbf{V}_1\|_F^2) \quad (5.38)$$

It could be observed that J_{star1} in Eq. (5.37) is identical to the following function if we set $\beta_{12} = 1$, $\phi_1 = 2T_u$ and $\theta_2 = 2T_v$,

$$J'_{star1} = \sum_{c=1}^k \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} u_{ci} r_{ij} v_{cj} - T_u \sum_{c=1}^k \sum_{i=1}^{n_1} u_{ci}^2 - T_v \sum_{c=1}^k \sum_{j=1}^{n_2} v_{cj}^2 \quad (5.39)$$

which is the objective function of fuzzy co-clustering approach called FCoDok [85]. This means, the FCoDok approach is a special case of the proposed approach which handles bi-type star-structured relational data and produces the fuzzy partition of the central-type objects and the rankings of attribute-type objects. The other group of information on the partition of the attribute type and rankings of central type objects can be obtain by solving J_{star2} in Eq. (5.38).

5.4.2 FC-MR for Extended Star-Structured Relational Data

The standard star-structure only consists of heterogeneous relations between the central type and attribute types. However, in some situations, the homogeneous relation of the central type is also available. We refer this structure as an extended star-structure – extends the star-structure by further considering the pairwise relation of objects of the central type. For example, in Webpage categorization, a Webpage contains both content information, i.e., the words on the page, and linkage information, i.e., the outgoing and ingoing links to other Webpages. Thus the Webpage dataset has both Webpage-word heterogeneous relation and Webpage-Webpage homogeneous relation. Another example is the technical paper data. Together with the words occurred in a paper, it also contains citations or references of other papers. We believe that both the content information of each individual document and the linkage relation among documents are helpful for clustering these documents. In fact, several recent studies on topic modeling began to incorporate the network structure when establish topic models, such as [66], [16] and [17]. The objective function of FC-MR for the extended star-structured relational data can be written as

$$J_{estar} = J_{star} + Tr(\beta_{11} \mathbf{U}_1^T \mathbf{R}_{11} \mathbf{V}_1) \quad (5.40)$$

with J_{star} given in Eq. (5.30) and $\mathbf{R}_{11}$ is the homogeneous relation matrix of the central type. Compared with the star-structure case, the objective function in Eq. (5.40) has one more additional term, which makes $\mathbf{U}_1$ been dependent on $\mathbf{V}_1$. Given initial $\mathbf{U}_1^{(0)}$ and $\mathbf{V}_1^{(0)}$, we update each $\mathbf{V}_\mu$, $\mathbf{U}_\mu$ for $\mu = \{2, 3, \dots, m\}$, and then re-estimate $\mathbf{V}_1$ and $\mathbf{U}_1$. The updating of any $\mathbf{U}_\mu$ and $\mathbf{V}_\mu$ still follow Eq. (5.22) and Eq. (5.26), respectively, where $\{\mathbf{G}_\mu\}_{\mu=2}^m$ used for updating attribute type $\{\mathbf{U}_\mu\}_{\mu=2}^m$ is the same as Eq. (5.36) and $\{\mathbf{H}_\mu\}_{\mu=2}^m$ for updating $\{\mathbf{V}_\mu\}_{\mu=2}^m$ is the same as Eq. (5.34); while for central type $\mathbf{U}_1$ and $\mathbf{V}_1$, Eq. (5.41) and Eq. (5.42) given below are used instead of Eq. (5.33) and Eq. (5.35), respectively.

$$\mathbf{G}_1 = \sum_{\mu=1}^m \beta_{1\mu} \mathbf{R}_{1\mu} \mathbf{V}_\mu \quad (5.41)$$

$$\mathbf{H}_1 = \sum_{\mu=1}^m \beta_{1\mu} \mathbf{R}_{1\mu} \mathbf{U}_\mu \quad (5.42)$$

5.4.3 FC-MR for Homogeneous Relational Data

In the case when only the matrix $\mathbf{R}_{11}$ records the relationships between each pair of objects of the central type is known, FC-MR reduces to a homogeneous relational data clustering. The objective function becomes

$$J_{hom} = Tr(\mathbf{U}_1^T \mathbf{R}_{11} \mathbf{V}_1) - \frac{\phi_1}{2} \|\mathbf{U}_1\|_F^2 - \frac{\theta_1}{2} \|\mathbf{V}_1\|_F^2 \quad (5.43)$$

which is identical to the objective function of Fk -Parts as in Eq. (3.11). This means the weighted medoids based fuzzy clustering approach Fk -Parts can be regarded as a special case of FC-MR for homogeneous relational data clustering.

5.5 Related Work

Several multi-type relational clustering approaches have been proposed. We have provided a literature review of these approaches in Chapter 2.4.7.2. Here we discuss the Matrix factorization based multi-type relational data clustering in details. Several existing approaches including [8, 30, 31, 89, 91] belong to this framework. We begin with the one for star-structured relational data, and then give a general form of update equations for non-negative matrix factorization based multi-type relational data clustering.

In [91], a non-negative matrix factorization based approach here referred as NMF-Star is proposed for star-structured multi-type relational data. Assume $\mathbf{R}_p$ denotes the relation between the central type and the p th attribute type, the approach in [91] aims to minimize the following objective:

$$\sum_p \|\mathbf{R}_p - \mathbf{U}_c \mathbf{B}_p \mathbf{U}_p^T\|^2 \quad (5.44)$$

where $\mathbf{U}_c$ is the cluster indicator matrix $n_c \times k_c$ of the central type, $\mathbf{U}_p$ is the cluster indicator matrix of the p th attribute type $n_p \times k_p$, and $\mathbf{B}_p$ records the association between the k_c clusters of central type and the k_p clusters of the p th attribute type. With $\mathbf{B}_p$, each type of objects need not to be grouped into the same number of clusters.

The iterative algorithm of NMF-Star alternately updates the cluster indicator matrix of each type with the following multiplicative formulas:

$$(\mathbf{U}_c)_{ic} \leftarrow (\mathbf{U}_c)_{ic} \frac{\sum_p (\mathbf{R}_p \mathbf{U}_p \mathbf{B}_p^T)_{ic}}{\sum_p (\mathbf{U}_c \mathbf{B}_p \mathbf{U}_p^T \mathbf{U}_p \mathbf{B}_p^T)_{ic}} \quad (5.45)$$

$$(\mathbf{U}_p)_{ic} \leftarrow (\mathbf{U}_p)_{ic} \frac{(\mathbf{R}_p^T \mathbf{U}_c \mathbf{B}_p)_{ic}}{(\mathbf{U}_p \mathbf{B}_p^T \mathbf{U}_c^T \mathbf{U}_c \mathbf{B}_p)_{ic}} \quad (5.46)$$

$$(\mathbf{B}_p)_{cf} \leftarrow (\mathbf{B}_p)_{cf} \frac{(\mathbf{U}_c^T \mathbf{R}_p \mathbf{U}_p)_{cf}}{(\mathbf{U}_c^T \mathbf{U}_c \mathbf{B}_p \mathbf{U}_p^T \mathbf{U}_p)_{cf}} \quad (5.47)$$

In another study [31], a more general objective function is proposed as below for relational data of various structures:

$$\sum_{\mu} w_{\mu} \|\mathbf{S}_{\mu} - \mathbf{U}_{\mu} \mathbf{D}_{\mu} \mathbf{U}_{\mu}^T\|^2 + \sum_{\mu < \nu} w_{\mu\nu} \|\mathbf{A}_{\mu\nu} - \mathbf{U}_{\mu} \mathbf{B}_{\mu\nu} \mathbf{U}_{\nu}^T\|^2 \quad (5.48)$$

where $\mathbf{S}_{\mu}$ denotes the pairwise relation between objects of the μ th type, and $\mathbf{A}_{\mu\nu}$ denotes the relation between two different types of objects. The update equations given in [31] is for a special case where only $\mathbf{S}_1$, $\mathbf{A}_{12}$ is available or nonzero, and $w_1 = w_{12} = 1$, i.e.,

$$\|\mathbf{S}_1 - \mathbf{U}_1 \mathbf{D}_1 \mathbf{U}_1^T\|^2 + \|\mathbf{A}_{12} - \mathbf{U}_1 \mathbf{B}_{12} \mathbf{U}_2^T\|^2 \quad (5.49)$$

For the above objective function, the update equations derived in [31] is in the form as

$$(\mathbf{U}_1)_{ic} \leftarrow (\mathbf{U}_1)_{ic} \frac{(\mathbf{S}_1 \mathbf{U}_1 \mathbf{D} + 0.5 \mathbf{A}_{12} \mathbf{U}_2 \mathbf{B}_{12}^T)_{ic}^{\frac{1}{4}}}{(\mathbf{U}_1 \mathbf{D}_1 \mathbf{U}_1^T \mathbf{U}_1 \mathbf{D}_1 + 0.5 \mathbf{U}_1 \mathbf{B}_{12} \mathbf{U}_2^T \mathbf{U}_2 \mathbf{B}_{12}^T)_{ic}^{\frac{1}{4}}} \quad (5.50)$$

$$(\mathbf{U}_2)_{ic} \leftarrow (\mathbf{U}_2)_{ic} \frac{(\mathbf{A}_{12}^T \mathbf{U}_1 \mathbf{B}_{12})_{ic}}{(\mathbf{U}_2 \mathbf{B}_{12}^T \mathbf{U}_1^T \mathbf{U}_1 \mathbf{B}_{12})_{ic}} \quad (5.51)$$

$$(\mathbf{D}_1)_{ic} \leftarrow (\mathbf{D}_1)_{ic} \frac{(\mathbf{U}_1^T \mathbf{S}_1 \mathbf{U}_1)_{ic}}{(\mathbf{U}_1^T \mathbf{U}_1 \mathbf{D}_1 \mathbf{U}_1^T \mathbf{U}_1)_{ic}} \quad (5.52)$$

$$(\mathbf{B}_{12})_{ic} \leftarrow (\mathbf{B}_{12})_{ic} \frac{(\mathbf{U}_1^T \mathbf{A}_{12} \mathbf{U}_2)_{ic}}{(\mathbf{U}_1^T \mathbf{U}_1 \mathbf{B}_{12} \mathbf{U}_2^T \mathbf{U}_2)_{ic}} \quad (5.53)$$

With the same denotation as used in our fuzzy approach FC-MR, we denote $\mathbf{R}_{\mu\mu} = \mathbf{S}_{\mu}$, $\mathbf{R}_{\mu\nu} = \mathbf{A}_{\mu\nu}$, $\mathbf{B}_{\mu\mu} = \mathbf{D}_{\mu}$ and let $\beta_{\mu\mu} = w_{\mu}$, $\beta_{\mu\nu} = w_{\mu\nu}$. Now we can write Eq. (5.48)

into a more concise form as below:

$$\sum_{\mu \leq \nu} \beta_{\mu\nu} \|\mathbf{R}_{\mu\nu} - \mathbf{U}_\mu \mathbf{B}_{\mu\nu} \mathbf{U}_\nu^T\|^2 \quad (5.54)$$

Following the derivation in [91], the update equation of $\mathbf{U}$ and $\mathbf{B}$ can be derived as:

$$(\mathbf{U}_\mu)_{ic} \leftarrow (\mathbf{U}_\mu)_{ic} \frac{(\sum_\nu \beta_{\mu\nu} \mathbf{R}_{\mu\nu} \mathbf{U}_\nu \mathbf{B}_{\mu\nu}^T)_{ic}}{(\sum_\nu \mathbf{U}_\mu \mathbf{B}_{\mu\nu} \mathbf{U}_\nu^T \mathbf{B}_{\mu\nu}^T)_{ic}} \quad (5.55)$$

$$(\mathbf{B}_{\mu\nu})_{ic} \leftarrow (\mathbf{B}_{\mu\nu})_{ic} \frac{(\beta_{\mu\nu} \mathbf{U}_\mu^T \mathbf{R}_{\mu\nu} \mathbf{U}_\nu)_{ic}}{(\mathbf{U}_\mu^T \mathbf{U}_\mu \mathbf{B}_{\mu\nu} \mathbf{U}_\nu^T \mathbf{U}_\nu)_{ic}} \quad (5.56)$$

We refer the algorithm with the above update formulas to minimize Eq. (5.54) as NMF-MR (Nonnegative matrix factorization for Multi-type relational clustering).

For the special case where only $\mathbf{R}_{11}$ and $\mathbf{R}_{12}$ exists, and $\beta_{11} = \beta_{12} = 1$, Eq. (5.54) reduces to (5.49) with $\mathbf{R}_{11} = \mathbf{S}$, $\mathbf{R}_{12} = \mathbf{A}$, $\mathbf{B}_{11} = \mathbf{D}$. In this case, the update equations of $\mathbf{U}_2$, $\mathbf{B}_{11}$ and $\mathbf{B}_{12}$ according to (5.55) and (5.56) are identical to Eqs. (5.51) to (5.53), respectively. But according to (5.55), the formula for updating $\mathbf{U}_{11}$ as shown below is slightly different from Eq. (5.50):

$$(\mathbf{U}_1)_{ic} \leftarrow (\mathbf{U}_1)_{ic} \frac{(\mathbf{R}_{11} \mathbf{U}_1 \mathbf{B}_{11}^T + \mathbf{R}_{12} \mathbf{U}_2 \mathbf{B}_{12}^T)_{ic}}{(\mathbf{U}_1 \mathbf{B}_{11} \mathbf{U}_1^T \mathbf{B}_{11}^T + \mathbf{U}_1 \mathbf{B}_{12} \mathbf{U}_2^T \mathbf{B}_{12}^T)_{ic}} \quad (5.57)$$

Although these two formulas of $\mathbf{U}_1$ result in comparable performance in both effective and efficiency, we prefer Eq. (5.57) for updating of $\mathbf{U}_1$ concerning theoretical completeness. To see a further simplified case, where only $\mathbf{R}_{12}$ exists, or $\mathbf{R}_{11} = \mathbf{S} = 0$ in Eq. (5.49). Due to the symmetry of the objective function with respect to $\mathbf{U}_1$ and $\mathbf{U}_2$, it is more reasonable to expect that the solutions of $\mathbf{U}_1$ and $\mathbf{U}_2$ are in the same form. Given $\mathbf{U}_2$ as in (5.51), the formula of $\mathbf{U}_1$ in Eq. (5.57) satisfies this property while Eq. (5.50) does not, because there is an additional exponent $1/4$.

5.6 Experimental Results

This section presents experimental studies of the proposed approach for clustering of different document datasets which are represented as multi-type relational data with different structures.

Figure 5.1: A toy problem of tri-type relational data clustering. The data consists of 30 objects, where $|\mathcal{X}_1| = 10$, $|\mathcal{X}_2| = 12$, and $|\mathcal{X}_3| = 8$ and two relations: $\mathcal{X}_1 \sim \mathcal{X}_2$ and $\mathcal{X}_1 \sim \mathcal{X}_3$. The corresponding relation matrices $\mathbf{R}_{12}$, $\mathbf{R}_{13}$ are plotted. Each entry in a relation matrix is plotted as a dot, and a larger dot indicates a stronger relationship.

5.6.1 A Running Example

First, we give a running example of FC-MR on a toy problem of a tri-type star-structured relational data. The simulated data contains 30 objects of three types, i.e., $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \mathcal{X}_3$, where $|\mathcal{X}_1| = 10$, $|\mathcal{X}_2| = 12$, and $|\mathcal{X}_3| = 8$. Two matrices $\mathbf{R}_{12}$ and $\mathbf{R}_{13}$ record the $\mathcal{X}_1 \sim \mathcal{X}_2$ relation and $\mathcal{X}_1 \sim \mathcal{X}_3$ relation, respectively. For this dataset, $\mathcal{X}_1$ is the central type while $\mathcal{X}_2$ and $\mathcal{X}_3$ are two attribute types. Fig. 5.1 shows the value distribution in two matrices, where each entry is plotted as a dot, and the size of the dot is proportional to the value of the relationship, i.e., a large dot indicates a large relationship and a small value of relationship turns out to be a small dot. According to the $\mathcal{X}_1 \sim \mathcal{X}_2$ relation shown in Figure 5.1a, it can be observed that the objects in $\mathcal{X}_1$ and $\mathcal{X}_2$ form two clusters, respectively, i.e., $\mathcal{X}_1 = \mathcal{C}_1^1 \cup \mathcal{C}_2^1$, with $\mathcal{C}_1^1 = \{x_1^1, x_2^1, x_3^1, x_4^1, x_5^1\}$, $\mathcal{C}_2^1 = \{x_6^1, x_7^1, x_8^1, x_9^1, x_{10}^1\}$, and $\mathcal{X}_2 = \mathcal{C}_1^2 \cup \mathcal{C}_2^2$, with $\mathcal{C}_1^2 = \{x_1^2, x_2^2, x_3^2, x_4^2, x_5^2, x_6^2, x_7^2\}$, $\mathcal{C}_2^2 = \{x_8^2, x_9^2, x_{10}^2, x_{11}^2, x_{12}^2\}$. From the other relation $\mathcal{X}_1 \sim \mathcal{X}_3$ in Figure 5.1b, it is seen that the objects in $\mathcal{X}_1$ form the same clusters as $\mathcal{C}_1^1$, $\mathcal{C}_2^1$, and $\mathcal{X}_3 = \mathcal{C}_1^3 \cup \mathcal{C}_2^3$, with $\mathcal{C}_1^3 = \{x_1^3, x_2^3, x_3^3, x_4^3\}$, $\mathcal{C}_2^3 = \{x_5^3, x_6^3, x_7^3, x_8^3\}$. It is also shown that $\mathcal{C}_1^1$ is associated to both $\mathcal{C}_1^2$ and $\mathcal{C}_1^3$, and $\mathcal{C}_2^1$ is associated to both $\mathcal{C}_2^2$ and $\mathcal{C}_2^3$.

Starting with a random partitioning of the central type, the updating values of memberships of the central type $\mathbf{U}_1$, and the rankings of two attribute types $\mathbf{V}_2$ and $\mathbf{V}_3$ are given in Table 5.1, where equal weights of two relation matrices are used. From

this table, it is seen that the algorithm quickly converges after 5 iterations. After two iterations, $\mathbf{V}_2$, $\mathbf{V}_3$ and $\mathbf{U}_1$ are already close to the final results. The cluster structures of the three types indicated by the converged values of $\mathbf{V}_2$, $\mathbf{V}_3$ and $\mathbf{U}_1$ are consistent with those expected ones.

5.6.2 Experiments on Star-Structured Relational data

In this experiment, we test the performance of FC-MR in clustering of tri-type star-structured relational data.

5.6.2.1 Data

We use datasets extracted from the 20newsgroups-18828. The original data of 20newsgroups¹ contains 18828 non-duplicated documents, which are categorized into 20 topics. As in [88] and [90], we generate three subsets of documents consisting of different subtopics as listed in Table 5.2. For example, dataset TM1 consists of clusters C_1 and C_2 , where C_1 is the *rec.sport* cluster which contains two subtopics or categories on *baseball* and *hockey*, and C_2 is the *talk.politics* cluster that has three subtopics on *guns*, *mid-east* and *misc*. Each of the three data sets is formed by randomly selecting 100 documents from each of the chosen subtopics.

5.6.2.2 Relations Derived

- *Document-Word* Relation: For each dataset, we use the *rainbow* toolkit [130] to get the *document-word* co-occurrence relation. Stop-words have been removed and 2000 words with the largest information gain are kept. Non-text documents are skipped by *rainbow*. After preprocessing, documents consisting of less than two words are removed. Finally we have 497, 598 and 794 documents for TM1, TM2 and TM3, respectively. The *tf-idf* weighting [20] is used for weighting the words in each document.
- *Document-Category* Relation: Other than the *document-word* relation which captures the document content by statistical information in terms of word frequency, we also generate another relation matrix *document-category* to indicate

¹<http://people.csail.mit.edu/jrennie/20Newsgroups/>

Table 5.1: Updating of rankings and memberships in FC-MR in iteration

Iteration 0: initialization					
Id			$\mathbf{U}_1$		
1	0.6311	0.7463			
2	0.0899	0.0103			
3	0.0809	0.0484			
4	0.7772	0.6679			
5	0.9051	0.6035			
6	0.5338	0.5261			
7	0.1092	0.7297			
8	0.8258	0.7073			
9	0.3381	0.7814			
10	0.2940	0.2880			

Iteration 1								
Id		$\mathbf{V}_2$	Id		$\mathbf{V}_3$	Id		$\mathbf{U}_1$
1	0.1066	0.0160	1	0	0	1	0.7866	0.2134
2	0.1010	0	2	0.1608	0	2	0.7368	0.2632
3	0.0929	0	3	0.3013	0.1480	3	0.7702	0.2298
4	0	0	4	0.1545	0	4	0.8055	0.1945
5	0.2368	0.0481	5	0	0	5	0.7670	0.2330
6	0	0	6	0.1203	0.3397	6	0.1910	0.8090
7	0.1173	0	7	0.2632	0.5123	7	0.2434	0.7566
8	0	0	8	0	0	8	0.1688	0.8312
9	0	0.0448				9	0.1177	0.8823
10	0.1197	0.4275				10	0.2327	0.7673
11	0.0975	0.1823						
12	0.1282	0.2813						

Iteration 2								
Id		$\mathbf{V}_2$	Id		$\mathbf{V}_3$	Id		$\mathbf{U}_1$
1	0.1656	0	1	0.1640	0	1	1	0
2	0.1389	0	2	0.2141	0	2	1	0
3	0.1738	0	3	0.3689	0	3	0.9953	0.0047
4	0.0593	0	4	0.2530	0	4	1	0
5	0.2064	0	5	0	0.1133	5	0.9475	0.0525
6	0.0763	0	6	0	0.3236	6	0	1
7	0.1796	0	7	0	0.4123	7	0.0744	0.9256
8	0	0.0800	8	0	0.1509	8	0	1
9	0	0.1409				9	0	1
10	0	0.3157				10	0.0037	0.9963
11	0	0.2019						
12	0	0.2615						

⋮

Iteration 5: converge								
Id		$\mathbf{V}_2$	Id		$\mathbf{V}_3$	Id		$\mathbf{U}_1$
1	0.1613	0	1	0.1743	0	1	1	0
2	0.1405	0	2	0.2201	0	2	1	0
3	0.1642	0	3	0.3561	0	3	0.9901	0.0099
4	0.0847	0	4	0.2494	0	4	1	0
5	0.1839	0	5	0	0.1435	5	0.9414	0.0586
6	0.0964	0	6	0	0.3102	6	0	1
7	0.1690	0	7	0	0.3635	7	0.0819	0.9181
8	0	0.1112	8	0	0.1828	8	0	1
9	0	0.1593				9	0	1
10	0	0.2813				10	0.0082	0.9918
11	0	0.2031						
12	0	0.2450						

Table 5.2: Topic structures of the three 20Newsgroup datasets: TM1, TM2 and TM3

TM1	$C_1: \{\text{rec.sport.baseball, rec.sport.hockey}\}$
	$C_2: \{\text{talk.politics.guns, talk.politics.mideast, talk.politics.misc}\}$
TM2	$C_1: \{\text{comp.graphics, comp.os.ms-windows.misc}\}$
	$C_2: \{\text{rec.autos, rec.motorcycles}\}$
	$C_3: \{\text{sci.crypt, sci.electronics}\}$
TM3	$C_1: \{\text{comp.sys.ibm.pc.hardware, comp.sys.mac.hardware}\}$
	$C_2: \{\text{rec.autos, rec.motorcycles}\}$
	$C_3: \{\text{sci.med, sci.space}\}$
	$C_4: \{\text{talk.politics.guns, talk.politics.mideast}\}$

which subtopic each document belongs. Each subtopic is a category and the relation matrix is a binary matrix with an entry is 1 if the document is from the corresponding subtopic otherwise is 0.

These two relations, i.e., *document-word* relation represented as $\mathbf{R}_{12}$ and *document-category* relation represented as $\mathbf{R}_{13}$ form a tri-type star-structure, where the central type is *document* and two attribute types are *word* and *category*. In this experiment, we use the normalized relation matrices, i.e., $\mathbf{R}'_{12} = \mathbf{D}_r^{-0.5} \mathbf{R}_{12} \mathbf{D}_c^{-0.5}$, where $\mathbf{D}_r$ and $\mathbf{D}_c$ are two diagonal matrices. Each main diagonal entry of $\mathbf{D}_r$ is the sum of each row of $\mathbf{R}_{12}$, and each main diagonal entry of $\mathbf{D}_c$ is the sum of each column of $\mathbf{R}_{12}$. Similar normalization is performed on $\mathbf{R}_{13}$. For these three 20newsgroups datasets, the *document-word* relation alone may be used for document clustering, as it provides the information that documents contain similar words should be labeled in the same cluster; while the *document-category* relation although clearly shows about which documents are in the same subtopic, it provides no indication on which subtopics are related to the same topic or cluster. Therefore, we let the *document-word* relation to be dominant in the clustering process while the *document-category* relation plays a supplementary role. In our experiment, we set $\beta_{12} = 0.9$ and $\beta_{13} = 0.1$.

5.6.2.3 FC-MR vs. HFCM and FCoDok

To see whether FC-MR can make use of the additional *document-category* relation to achieve any improvements on the clustering results, we first compare the document clusters generated by HFCM [51] and FCoDok [85] based on $\mathbf{R}_{12}$ with those generated

by FC-MR based on both $\mathbf{R}_{12}$ and $\mathbf{R}_{13}$. HFCM is a modified fuzzy c-means clustering based on cosine distance, and FCoDok is a fuzzy co-clustering approach. In our experiment, the number of clusters k is set to be equal to the real number of classes for each dataset. For FC-MR on star-structured data, we only produce partitioning of documents and rankings of words and categories. Following the discussion in Chapter 5.3.4, we initially set $\eta_2 = 1$, $\rho_1 = 1/n_2 * k$, $\eta_3 = 1/n_2 * n_3$. For other two approaches, we tried different values of parameters on a grid, and choose those give the best results. In this experiment, we find the following settings allow each approach to perform well: $T_u = 0.001, T_v = 1$ for FCoDok, and $m = 1.02$ for HFCM. For all fuzzy approaches, a truncated partitioning is obtained by assigning each document to the cluster with the largest membership.

Table 5.3 to Table 5.5 give the best clustering results that each approach can achieve among 30 trials with random initializations on the three datasets, respectively. From these contingency tables, it can be seen that with a proper initialization, FC-MR is able to label all the documents correctly on all the three datasets, while FCoDok and HFCM mis-cluster different numbers of documents on different datasets. The mislabeled number of documents by FCoDok and HFCM on TM1 and TM2 are close, but more documents are labeled correctly by FCoDok than HFCM on TM3. Although FCoDok and HFCM make use of the same *document-word* matrix, HFCM treats each document as a vector and use cosine similarity to measure the closeness of two documents, while FCoDok treats document and word as two different object types, and documents are clustered based the ranking of words. There is no explicit similarity measure defined in FCoDok and FC-MR.

To see some more detailed comparison, we plot $\mathbf{u}_1 = \{u_{1,1}, u_{2,1} \dots, u_{500,1}\}$, the membership values of documents in C_1 of TM1 produced by HFCM and FC-MR in Figure 5.2. According to the summation constraints, the memberships of these objects in the other cluster follow $\mathbf{u}_2 = 1 - \mathbf{u}_1$. For TM1, the ground truth partitioning is that the first 198 documents are in C_1 : *rec.sport* and the rest are in C_2 : *talk.politics*. It can be seen from Figure 5.2a that in HFCM, for the 21th document, $u_{21,1} < u_{21,2}$, although the ground truth label of this document is C_1 . At the same time, six documents (labeled by circles) which should be in C_2 are assigned larger memberships in C_1 .

Table 5.3: Contingency table of three fuzzy approaches for TM1

	HFCM		FCoDok		FC-MR	
	Clust 1	Clust 2	Clust 1	Clust 2	Clust 1	Clust 2
Class 1	197	1	196	2	198	0
Class 2	6	293	4	295	0	299

Table 5.4: Contingency table of three fuzzy approaches for TM2

	HFCM			FCoDok			FC-MR		
	Clust 1	Clust 2	Clust 3	Clust 1	Clust 2	Clust 3	Clust 1	Clust 2	Clust 3
Class 1	193	3	3	194	4	1	199	0	0
Class 2	2	188	10	2	190	8	0	200	0
Class 3	20	9	170	21	11	167	0	0	199

Table 5.5: Contingency table of three fuzzy approaches for TM3

	HFCM				FCoDok				FC-MR			
	Clust 1	Clust 2	Clust 3	Clust 4	Clust 1	Clust 2	Clust 3	Clust 4	Clust 1	Clust 2	Clust 3	Clust 4
Class 1	189	4	5	1	194	1	4	0	199	0	0	0
Class 2	3	188	5	3	4	190	4	1	0	199	0	0
Class 3	4	9	180	6	6	6	182	5	0	0	199	0
Class 4	0	6	2	189	1	6	4	186	0	0	0	197

Figure 5.2: Fuzzy memberships of documents with respect to C_1 of TM1 by HFCM and FC-MR. The (horizontal) x-axis denotes the document id ($i = 1, \dots, 500$) and the (vertical) y-axis shows the value of the membership u_{i1} . In (a), small circles label objects that are clustered incorrectly.

These results may indicate that the representation of these documents in terms of word frequency is not sufficient enough for correctly assigning those documents into the right clusters. The document-category relation helps to clarify the uncertainty or correct the misleading information in the document-word relation. For example, according to the document-category relation, it is known that the 21th document possibly should be grouped together with the other 99 documents as they belong to the same category. This is confirmed from Figure 5.2b that when the document-category relation is further incorporated, those previously mis-clustered documents are now assigned reasonable memberships in two clusters which are consistent with the ground truth. Although from the document-category matrix, it can be told that which documents are in the same subtopic, this relation alone delivers no information on the hierarchy structure of the subtopics, i.e. which subtopics should be combined into high-level clusters. Therefore, clustering only based on the document-category relation does not produce any meaningful result, which turn out to be random combinations of those subgroups into the specified number of clusters.

5.6.2.4 Comparison of Accuracy and NMI

Now we compare FC-MR with another two existing multi-type relational data clustering approaches: NMF-MR [31, 91] and SRC [90]. The NMF-MR approach is a non-negative matrix factorization based multi-type relational data clustering as discussed in Chapter 5.5. The SRC approach is a spectral method for multi-type relational data clustering, which minimizes a similar objective as NMF-MR but only consider relations between different types of objects. For each of the three approaches, we tried different weights for the two relations and report the best result of each approach. Specifically, we let $\beta_{12} = [0, 0.1, \dots, 1]$ and $\beta_{13} = 1 - \beta_{12}$.

For each dataset, each approach is run 10 trials. To avoid bad initializations with which the *word-category* relation may dominate in the clustering process to produce random combinations of subtopics, we first run HFCM with random initializations only based on the *document-word* relation, and then use the document fuzzy memberships produced as the initial document partitioning in all the five approaches. Table 5.6 shows the mean and standard deviation of *Accuracy* (%) and *NMI* values (%) over 10 trials.

Table 5.6: Performance comparison of three multi-type relational data clustering approaches for 20Newsgroups data

Accuracy			
	TM1	TM2	TM3
SRC	100.00 $\pm$ 0.00	80.17 $\pm$ 10.25	86.56 $\pm$ 16.69
NMF-MR	97.91 $\pm$ 6.34	86.66 $\pm$ 12.10	83.43 $\pm$ 11.23
FC-MR	100.00 $\pm$ 0.00	95.00 $\pm$ 15.81	91.27 $\pm$ 11.81

NMI			
	TM1	TM2	TM3
SRC	100.00 $\pm$ 0.00	73.14 $\pm$ 2.79	90.93 $\pm$ 9.74
NMF-MR	93.20 $\pm$ 19.44	77.24 $\pm$ 17.82	77.90 $\pm$ 12.48
FC-MR	100.00 $\pm$ 0.00	93.69 $\pm$ 19.95	91.69 $\pm$ 12.44

Table 5.7: Top 10 words in each cluster of the three newsgroup datasets generated by FC-MR

TM1		TM2			TM3			
<i>C</i> ₁ : <i>rec.sport</i>	<i>C</i> ₂ : <i>talk.politics</i>	<i>C</i> ₁ : <i>comp</i>	<i>C</i> ₂ : <i>rec</i>	<i>C</i> ₃ : <i>sci</i>	<i>C</i> ₁ : <i>comp.sys</i>	<i>C</i> ₂ : <i>rec</i>	<i>C</i> ₃ : <i>sci</i>	<i>C</i> ₄ : <i>talk.politics</i>
game	jews	windows	car	key	mac	bike	space	israel
hockey	israel	file	bike	clipper	drive	car	nasa	israeli
team	people	graphics	dod	chip	card	dod	moon	jews
games	government	files	bmw	keys	scsi	bikes	launch	arab
players	gun	tiff	dog	nsa	apple	list	pat	arabs
ca	state	version	cars	encryption	disk	ride	shuttle	turkish
baseball	children	ftp	insurance	netcom	mb	honda	henry	jewish
season	cramer	site	bikes	phone	software	bmw	orbit	mr
win	fbi	program	audi	des	drives	oil	alaska	armenians
espn	cosmo	dos	motorcycle	government	driver	miles	mission	muslim

It can be seen from Table 5.6 that FC-MR gives the highest *Accuracy* and *NMI* values among three approaches on all the three datasets. Although SRC gives the best results as FC-MR on TM1, its *Accuracy* value on TM3 is much lower than that of FC-MR, and its results on TM2 are worse than both FC-MR and NMF-MR. These results show that our fuzzy approach FC-MR is more favorable than nonnegative matrix factorization based and spectral clustering based approach for multi-type relational data clustering.

5.6.2.5 Ranking of Words and Categories

Together with the documents clusters, we also obtain the word rankings and category rankings. The ranking values of each category in each cluster is shown in the three matrices given in Eq. (5.58) to Eq. (5.60) for TM1, TM2 and TM3, respectively, where each row indicates a category and each column indicates a cluster or a topic. Top 10 words of each cluster of three datasets are also listed in Table 5.7. As attribute types, *category* and *word* can be used for description or interpretation of the document clusters. We can see that those key categories and key words which have large ranking values provide meaningful information to tell what topic each document cluster possibly related to. If we take C_1 of TM1 for example, from its associated key categories named *baseball* and *hockey*, and key words, such as *game*, *hockey*, *team*, *players*, *baseball*, we may guess that documents of this cluster are about sports, possibly on hockey and baseball. Such kind of information can be used as a short summary of the cluster to give a quick sense of the whole document cluster without looking through each of the complete long documents.

$$\begin{array}{c}
 \begin{array}{cc}
 & \begin{array}{cc}
 \textit{rec.sport} & \textit{talk.politics}
 \end{array} \\
 \begin{array}{c}
 \textit{baseball} \\
 \textit{hockey} \\
 \textit{guns} \\
 \textit{mideast} \\
 \textit{misc}
 \end{array} & \left(\begin{array}{cc}
 0.4327 & 0 \\
 0.5673 & 0 \\
 0 & 0.3067 \\
 0 & 0.3506 \\
 0 & 0.3428
 \end{array} \right)
 \end{array}
 \end{array}
 \quad (5.58)$$

$$\begin{array}{c}
 \begin{array}{c}
 \textit{graphics} \\
 \textit{os.ms-windows.misc} \\
 \textit{autos} \\
 \textit{motorcycles} \\
 \textit{crypt} \\
 \textit{electronics}
 \end{array} & \begin{array}{ccc}
 \textit{comp} & \textit{rec} & \textit{sci} \\
 \left(\begin{array}{ccc}
 0.4716 & 0 & 0 \\
 0.5284 & 0 & 0 \\
 0 & 0.4579 & 0 \\
 0 & 0.5421 & 0 \\
 0 & 0 & 0.7018 \\
 0 & 0 & 0.2982
 \end{array} \right)
 \end{array}
 \end{array}
 \quad (5.59)$$

$$\begin{array}{c}
 \begin{array}{c}
 \textit{ibm.pc.hardware} \\
 \textit{mac.hardware} \\
 \textit{autos} \\
 \textit{motorcycles} \\
 \textit{med} \\
 \textit{space} \\
 \textit{guns} \\
 \textit{mideast}
 \end{array} & \begin{array}{cccc}
 \textit{comp.sys} & \textit{rec} & \textit{sci} & \textit{talk.politics} \\
 \left(\begin{array}{cccc}
 0.5265 & 0 & 0 & 0 \\
 0.4735 & 0 & 0 & 0 \\
 0 & 0.3749 & 0 & 0 \\
 0 & 0.6251 & 0 & 0 \\
 0 & 0 & 0.4056 & 0 \\
 0 & 0 & 0.5944 & 0 \\
 0 & 0 & 0 & 0.3530 \\
 0 & 0 & 0 & 0.6470
 \end{array} \right)
 \end{array}
 \end{array}
 \quad (5.60)$$

5.6.3 Experiment on Extended Star-Structured data

In this experiment, we test the performance of FC-MR on relational data with an extended star-structure. Two groups of document datasets are used. Each dataset is described by the *document-document* relation denoted as $\mathbf{R}_{11}$ and *document-term*

relation denoted as $\mathbf{R}_{12}$. The first group contains three document datasets which are used in Chapter 3. The second group is the five Cora datasets that have been used in Chapter 4. For Cora datasets, the *document-document* relation is obtained as the citation information among papers. Since there is no *document-document* relation directly available for the first group of datasets, we follow [31] to simulate this relation based on the ground truth classes. More specifically, two documents are linked or the corresponding entry in $\mathbf{R}_{11}$ is 1 with a probability of 0.2 if these two documents are in the same class; or the two documents are linked with a probability of 0.1 if they are in different classes.

We compare FC-MR with other three approaches including NMF-MR: the non-negative matrix factorization based multi-type relational data clustering, iTopicModel [17]: a generative model-based approach, which consider both text and structure information for documents, and LinkFCM: the fuzzy c-means with links which is proposed in Chapter 4. It has been shown in Chapter 4 that LinkFCM outperforms HFCM and Fk -Parts, and the quadratic regularization based variant is better than the fuzzifier based one. Here we compare FC-MR with quadratic regularization based LinkFCM (i.e., LinkFCMq). The SRC approach is not compared here as it does not consider document-document relation. Following the guide line discussed earlier, we find the initial setting $\eta_1 = 1$, and $\rho_1 = \rho_2 = 1/n_1 * k$, $\eta_2 = 1/n_1 * n_2$ gives good result for FC-MR on all the datasets except tr12. For this dataset, the initial value $\rho_1 = \rho_2 = 0.0256$ is too large, so we adjusted to $\rho_1 = \rho_2 = 0.01$. For both NMF-MR and FC-MR, we tried different values of weights of the two relations for each dataset, i.e., $\beta_{11} = [0, 0.1, \dots, 1]$ and $\beta_{12} = 1 - \beta_{11}$. The best result of each approach is reported.

5.6.3.1 Comparison of Effectiveness

Each approach is run for 10 trials with random initialization. Table 5.8 and Table 5.9 shows the mean and standard *Accuracy* (%) values and *NMI* (%) values, respectively. Bar plot is also provided under each table for easier visualization.

It is seen that on these two groups of datasets, FC-MR performs the best of these four approaches. The performance of LinkFCM on two groups of datasets is not consistent. As shown from the two Tables, LinkFCM achieves the second best results

on five Cora datasets. However, on the other three datasets, LinkFCM is only better than iTopicModel while much worse than FC-MR. The inconsistent in the performance of LinkFCM is possibly due to the different quality of the document-document relation of two groups of datasets. Compared with the document-document relation of the Cora datasets, which records the citation information, the simulated one of the first group of datasets may be more noisy. Because a research paper seldom cite other papers which are in different areas, for Cora datasets, there are usually no links among documents from different classes. However, for the datasets of the first group, two documents are simulated to be linked with a probability of 0.1 even they are in different classes, while two documents are linked with a probability of 0.2 even we known they are from the same class. The probability producing positive links is not very large compared to the probability producing negative or noisy links. This means the document-document relation matrix of the first group of documents contains noisy information with a relative high level. Nevertheless, FC-MR achieves consistently good results on all the datasets. This may indicate that compared with LinkFCM, the formulation of FC-MR is less sensitive to the “false links” in the document-document relation.

It is also observed that comparing with 20newsgroups datasets, the *Accuracy* and *NMI* values for Cora datasets are much lower. This is mainly because that the clusters of 20Newsgroups datasets are well separated since each cluster corresponds to a topic, and all the topics are very different, e.g., *sport* and *politics*; while the Cora datasets consists of a group of subfields, which are related to the same research field. Other than that, for each document in a 20Newsgroups dataset, we make sure that it at least contains two words excluding stop words; while quite a few documents in Cora data have no abstract being recorded, and these documents are not necessary removed during the preprocessing [131]. This means in each Cora dataset, some documents contain no words. Both the overlaps among clusters and less information on the content make Cora datasets more challenging for clustering.

5.6.3.2 Comparison of Efficiency

We now compare the time efficiency of four approaches. Given a relation $\mathbf{R}_{11}$ between each pair of n_1 documents and $\mathbf{R}_{12}$ between n_1 documents and n_2 words, the time

Table 5.8: Comparison of *Accuracy* of four approaches iTopicModel, LinkFCM, NMF-MR and FC-MR, using two groups of document datasets

	iTopicModel	LinkFCM	NMF-MR	FC-MR
Multi5	76.94 $\pm$ 8.45	92.19 $\pm$ 8.50	96.54 $\pm$ 0.18	97.98 $\pm$ 0.00
tr12	54.35 $\pm$ 6.15	61.47 $\pm$ 4.56	65.97 $\pm$ 6.07	77.54 $\pm$ 2.82
re0	37.51 $\pm$ 3.40	38.72 $\pm$ 2.47	49.53 $\pm$ 3.00	76.92 $\pm$ 3.47
DS	29.85 $\pm$ 3.00	42.06 $\pm$ 2.74	41.45 $\pm$ 2.26	44.90 $\pm$ 3.23
HA	34.38 $\pm$ 4.17	47.62 $\pm$ 3.73	45.82 $\pm$ 5.53	49.85 $\pm$ 2.03
ML	39.97 $\pm$ 4.66	60.49 $\pm$ 6.61	49.37 $\pm$ 4.41	66.53 $\pm$ 5.81
OS	50.81 $\pm$ 6.65	64.80 $\pm$ 6.24	52.69 $\pm$ 1.87	67.46 $\pm$ 5.59
PL	33.30 $\pm$ 2.66	43.18 $\pm$ 3.86	34.37 $\pm$ 1.91	43.56 $\pm$ 3.71

Table 5.9: Comparison of *NMI* of four approaches iTopicModel, LinkFCM, NMF-MR and FC-MR, using two groups of document datasets

	iTopicModel	LinkFCM	NMF-MR	FC-MR
Multi5	67.82 $\pm$ 6.09	83.94 $\pm$ 7.36	89.15 $\pm$ 0.53	93.32 $\pm$ 0.00
tr12	47.68 $\pm$ 5.07	62.26 $\pm$ 6.10	62.80 $\pm$ 4.17	72.54 $\pm$ 2.21
re0	28.18 $\pm$ 3.28	32.84 $\pm$ 2.14	50.69 $\pm$ 1.62	70.65 $\pm$ 2.72
DS	17.08 $\pm$ 2.36	35.09 $\pm$ 2.74	27.65 $\pm$ 1.73	37.13 $\pm$ 1.43
HA	16.11 $\pm$ 3.19	39.78 $\pm$ 2.05	35.55 $\pm$ 2.77	43.26 $\pm$ 3.22
ML	21.71 $\pm$ 2.95	47.51 $\pm$ 4.40	27.76 $\pm$ 3.18	50.27 $\pm$ 4.15
OS	16.59 $\pm$ 4.16	30.43 $\pm$ 6.19	15.71 $\pm$ 0.82	37.61 $\pm$ 3.49
PL	17.92 $\pm$ 1.75	30.25 $\pm$ 2.07	26.06 $\pm$ 1.67	37.05 $\pm$ 1.95

(a)

Figure 5.3: Comparison of time efficiency of the four approaches: iTopicModel, LinkFCM, NMF-MR and FC-MR. The time required by each of the approaches on eight datasets are displayed in log(seconds).

complexity of these four approaches are the same $O(kln^{max}n_1)$, where k is the number of clusters, l is the number of iterations, and $n^{max} = \max\{n_1, n_2\}$. But FC-MR is faster and requires less storage than the other three.

Figure 5.3 shows the running time of one trial required by four approaches to produce the results on five Cora datasets. All algorithms are implemented with MATLAB 7.4 on a machine with Dual 2.66GHz Intel Core processors and 2GB RAM. We use the codes of iTopicModel provided by the authors. It is obvious that the two proposed fuzzy approaches LinkFCM and FC-MR are faster than the other two. FC-MR requires the least time of the four approaches on five Cora datasets while LinkFCM is the fastest of all on the four CLUTO datasets. Comparing iTopicModel, NMF-MR is faster.

5.7 Summary

In this chapter, we worked on fuzzy clustering approaches for multi-type relational data mining. The main idea of the proposed approach FC-MR is to estimate the fuzzy memberships of objects in each cluster based on the ranking values of related

objects in that cluster, and the memberships are in turn used for updating the rankings of objects in each cluster. The proposed FC-MR is able to handle relational data with various structures. We use document clustering as a real-world example to illustrate how the the proposed approaches can be applied in various situations according to the requirement of the applications and the availability of relational information in the data. Connections between FC-MR and existing fuzzy approaches show that FC-MR can be seen as generalizations of fuzzy approach for homogeneous relational data and fuzzy co-clustering. Our experimental study on benchmark document datasets show that FC-MR performs consistently better than state-of-the-art multi-type relational clustering approaches and also generally better than the LinkFCM approach proposed in Chapter 4. This demonstrates the great potential of proposed approach for relational data clustering.

Chapter 6

Application: Subtopic-based Multi-document Summarization

6.1 Overview

In this Chapter, we are going to present an useful application of clustering in text summarization. Specifically, we propose a new sentence-extractive based multi-document summarization system SumCR, where a new subtopic-level features *Exemplar* is defined based on the sentence clusters produced by the *Fk*-Parts clustering approach. Due to the possible existence of subtopics, it may be more effective to define features in a subtopic-level. Therefore we use *Fk*-Parts to cluster sentences into groups based on the similarities among pairs of sentences. After clustering, sentences which are similar to each other in content are grouped into the same cluster with each cluster corresponds to a subtopic or topic area. An important reason of using the *Fk*-Parts clustering approach is that together with the partitioning of sentences, the prototype weight provides a measure of representativeness of each sentence in the associated group. Based on the resulting clusters and the prototype weights, the *Exemplar* weights are computed for each of the sentences. To maintain a good performance on various data sets, we score sentence by combining the weights of *Exemplar* and another feature *Position*, which reflects the location of the sentence in the corresponding document. With the combination of subtopic-level feature and document-level feature, SumCR aims to produce a summary with a high relevance as well as a wider coverage and less redundancy. By further considering the closeness to the user-given query, SumCR is extended for task-focused summarization.

We conduct experiments for both generic and task-focused multi-document summarization on the benchmark data DUC2004 and DUC2005, respectively. The summarization results are evaluated use the same ROUGE scores which are used by DUC (Document Understanding Conferences) ¹. The experimental results show that in both tasks, the proposed approach outperforms the popular multi-document summarization system MEAD [132] and the state-of-the-art clustering-based summarization approach SNMF [133]. Our approach also produces better or comparable results with top-10 systems participated in the DUC competition of the corresponding year.

6.2 Background and Motivations

We first give some background knowledge on multi-document summarization and then give the motivations of why apply the *Fk*-Parts clustering approach in the proposed summarization system.

6.2.1 Background

The explosive growing of on-line documents in recent years attracts a lot of interests in multi-document summarization (MDS), which automatically generates a single short summary for a set of documents related to the same topic. According to the content, summaries can be either task-focused or generic. A task-focused, or topic-driven, or query-oriented summary is tailored for the requirement of a particular group of users by extracting information from the documents that is most related to a given topic, and a generic summary tries to present all the main points conveyed by the input documents without any specific external instruction or requirement. A summary can be generated in an extractive way or abstractive way [134–136]. Extractive summarization simply extracts salient information, such as sentences, from the input documents and “put them together” to form summaries, and abstractive summarization creates summaries by synthesizing and rewriting sentences based on the contextual and linguistic understanding. Here we focus our efforts on multi-document summarization through sentence extraction.

¹<http://duc.nist.gov/>

The framework established by [137] is a classic paradigm in extractive summarization that continues to influence the research work on this topic today. Four features are used in it, namely *cue words*, *title words*, *key words* and *sentence location*. The overall score of each sentence is obtained through a linear combination of the weights of the four features. A summary is then formed by choosing a needed number of sentences with the largest overall scores. Quite a lot of subsequent researches on extractive summarization, including some notable summarization system, such as SUMMARIST [138], MEAD [132], and SumBasic [139], follow this framework with improvements made by expanding the set of features and using more sophisticated methods for weighting different features. Other than introducing a number of new features, [138] studied the best positional feature for sentence extraction. [139] studied the usefulness of a particular feature—frequency in their summarization system SumBasic. The MEAD proposed by [132] is one successful system for large scale multi-document summarization where a core feature *Centroid* is derived to measure the centrality of a sentence. The *Centroid* score of each sentence is computed based on its similarity to the centroid, a virtual sentence consisting of important words. Other than following the classic framework, other techniques from related fields are also employed to document summarization. Graph-based methods such as LexPageRank [140], TextRank [141] are proposed to rank sentence using graph ranking algorithms. At the same time, techniques in machine learning area also have been applied for summarization, including supervised approaches, such as [142, 143] and unsupervised approaches, such as Latent semantic analysis [144], Hidden Markov Model [145], Conditional Random Field [146], and clustering-based approaches [133, 147, 148].

Due to the existence of information overlapping and repeating in the documents, simply selecting sentences with high global relevance scores may cause redundancy, i.e. some of those selected sentences may provide the same or similar information. This problem becomes more typical in multi-document summarization. Some researchers began to incorporate explicit ways to reduce the redundancy and enhance the diversity in the summary. [149] proposed the Maximal Marginal Redundancy (MMR) principle to reduce redundancy by selecting sentences to be not too similar to each other. Other than the redundancy problem, different subtopics or concepts may be involved

for describing the same related topic. Traditional statistical features obtained based on the whole input corpus may be unable to capture some of the underlying subtopics, and thus loss some degree of completeness of the summary content. Subtopic-based summarization with clustering technique has been studied as a generalization of MMR [133, 147, 148]. In these approaches, sentences segmented from the document set are first grouped into clusters so that sentences in the same cluster are more similar than those in different clusters. Based on the sentence clusters, the most representative sentence within each cluster is picked to form the summary. Selecting sentence from different clusters may help to increase the diversity and the coverage, and at the same time alleviate the redundancy problem. However, when there is no obvious subtopics in the data set, clustering-based approaches may become less effective, and this causes the degeneration of the quality of summaries produced.

6.2.2 Why Fk -Parts?

There are various clustering approaches, why Fk -Parts is a good choice? Now we highlight several important properties of Fk -Parts which make it more favorable than other clustering approaches with the application of summarization.

- (1) *Fuzzy clustering*: Compared with hard or crisp clustering like k-means, fuzzy clustering is more suitable to be used to generate sentence clusters. In hard clustering, overlap among clusters is not allowed, while in fuzzy clustering, each sentence can belong to more than one clusters with different degrees of belonging. The use of fuzzy membership in clustering is able to capture the structure of the data more naturally because it is very likely that a single sentence may be related to multiple subtopics to different degrees.
- (2) *Similarity-based*: Unlike some other fuzzy clustering approaches which require the sentences to be represented as vectors of a set of terms, Fk -Parts is designed for relational data, i.e. similarity or dissimilarity. This form of inputs enables the overall performance of SumCR to be benefited from the development of a more accurate similarity measure. In other words, any advanced technique which gives a good similarity measure of sentences including statistical models

and other models incorporating high level NLP techniques, can be easily adopted and integrated into SumCR. There are already many studies, such as [133, 150, 151], that reported the success of using good similarity measure to achieve final improvement in clustering and classification.

- (3) *Prototype Weight*: The most distinctive characteristic of Fk-Parts compared with other existing (dis)similarity-based fuzzy clustering is that cluster-based representativeness called prototype weight is generated along with the clusters. When the input data is a similarity matrix, it is unable to define a cluster center as it did in a vector space, therefore, most existing (dis)similarity-based clustering approaches concentrate on obtaining the partition of the data without defining any variable to reflect the representativeness of objects in each cluster, such as SNMF in [133]. In Fk-Parts, we are interested in both partition and cluster-based typicality. To achieve this, fuzzy membership together with the prototype weight are defined. With prototype weights, all sentences are able to represent each cluster to a certain extent. This enables Fk-Parts to capture the cluster structure more accurately hence creates clusters with good qualities. Moreover, according to the definition, prototype weight provides a good measure of within-cluster representativeness of each sentence.

6.3 The Proposed Method: SumCR

In this section, we present the details of the proposed SumCR for multi-document summarization. We first give an overview of the core SumCR system for generic multi-document summarization. It follows by the detailed introduction to each of the components. An extended version of SumCR will also be given for task-focused summarization.

6.3.1 System Overview

The overall system structure of the proposed SumCR is shown in Fig. 6.1. Given a set of documents $\mathcal{D} = \{d_1, d_2, \dots, d_n\}$ related to the same topic, SumCR

produces a short summary for the document set with several procedures given as follows:

Compute the sentence similarity matrix: First, each of the documents is segmented into sentences to get a pool of sentences $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$, where N is the total number of sentences from n documents. After that, the similarity matrix $S_{N \times N}$ recording similarities between every pair of sentences are calculated or estimated with any proper similarity measure techniques.

Calculate the *Exemplar* score: Once the similarity matrix is obtained, we perform the fuzzy clustering *Fk*-Parts to group sentences into subsets or clusters. At the same time, prototype weight of each sentence with respect to a cluster are also produced at the end of the clustering process. The *Exemplar* weight of a sentence considers both the closeness to sentences in the same cluster and the prototype weight of this sentence in the cluster it belongs to.

Calculate the *Position* score: The *Position* score decreases with the sequence of the sentence appears in the document, i.e. the *Position* value of the first sentence is the largest while the last sentence is the smallest.

Form the summary: After the weights of *Exemplar* and *Position* are calculated, we get the final score as the combination of the two. Sentences are ranked with their scores in descendent order and those top ones are selected to form the summary. The pairwise similarities of sentences being selected need to be smaller than a given threshold.

6.3.2 Feature, Score and Selection

Now, we give the definition of each of the features used in SumCR for sentence scoring and also present how to select sentences based on the final scores.

Figure 6.1: An overview of the proposed SumCR summarizer.

6.3.2.1 Exemplar: A subtopic-based feature

Based on the fuzzy membership $u_{ci} \in U_{K \times N}$ and prototype weight $v_{cj} \in V_{K \times N}$ produced by Fk-Parts, two possible ways might be used to calculate the *Exemplar* weight, the one based on truncated clusters and the one based on the original fuzzy clusters.

For the truncated case, we assign each sentence i to a cluster with the largest membership, i.e.

$$x_i \in \mathcal{C}_k, \text{ with } k = \arg \max_f u_{fi} \quad (6.1)$$

and we obtain the prototype weight of this sentence in the associated cluster $Pw_i = v_{ki}$. After that we calculate the sum of similarities between this sentence to all other sentences in the same cluster, i.e.

$$Sm_i = \sum_{j \neq i, j \in \mathcal{C}_k} s_{ij} \quad (6.2)$$

where sentence j is in the same cluster $\mathcal{C}_k$ as sentence i . For a sentence i , both Sm_i and Pw_i reflect how central this sentence is in the related cluster. We hybridize these two measures to get *Exemplar* as below

$$Exemplar_i = (1 - \alpha) \times Pw_i + \alpha \times Sm_i \quad (6.3)$$

where α is the hybrid parameter. To make effective use of both Pw and Sm , α is set in $(0, 1)$.

By truncating, some detailed information of the partition may be lost. To maintain those information, the *Exemplar* score can be calculated based on the fuzzy assignment U as follows:

$$Smf_i(c) = \sum_j u_{ci} u_{cj} s_{ij} \quad (6.4)$$

$$Pwf_i(c) = u_{ci} v_{ci} \quad (6.5)$$

$$Exemplar_i = \max_c \{(1 - \alpha) \times Smf_i(c) + \alpha \times Pwf_i(c)\} \quad (6.6)$$

Normally, the results obtained with two strategies are close. We use the fuzzy one in the experiments of this paper.

6.3.2.2 Position

We have just shown that the feature *Exemplar* plays a critical role in capturing subtopic-based representativeness. However, the effectiveness of this feature depends on the existence of the subtopic structure in the data set. In other words, when the data set does not contain obvious subtopics, such a cluster-based feature may become less effective. Therefore, to maintain a good performance on different data sets and further improve the quality of the summary, we also consider another feature, *Position*, which has been shown to be a good indication of the significance of a sentence. Early investigation on sentence position was made in [152], which shows that a very large portion of topic sentence came as the first sentence of the paragraph. [137] found sentence location is the best individual feature compared with other three word-level features. Sentence position is also used as a default feature in MEAD. Here we adopt a simple weighting scheme for *Position*, i.e. the score of *Position* decreases as the sentence becomes farther from the beginning of the document. Specifically, we use Eq. (6.7) to assign the *Position* score to each sentence:

$$Position_i = \frac{1}{\sqrt{i}} \quad (6.7)$$

where i is the sentence index in the document it appears.

6.3.2.3 Score for generic summarization

After both scores of *Exemplar* and *Position* being calculated, we combine them to obtain the overall *Significance* (*Sig*) score of each sentence as

$$Sig_i = w_e \times Exemplar_i + w_p \times Position_i \quad (6.8)$$

where $w_e, w_p \geq 0$ are the combination parameters. The weights of each feature are scaled into $[0, 1]$ before combination.

6.3.2.4 Score for Task-focused summarization

The SumCR that has been presented so far is for generic summarization. To extend it for task-focused summarization, we need to further consider the information contained in the user-specified query when extracting sentences. A simple and widely used way is to consider the similarity between each sentence i and the query denoted as $SimToQ_i$ as a feature and add it into the feature set. In such a way, our task-focused significance score $SigQ_i$ of each sentence i is given as below

$$SigQ_i = w_e \times Exemplar_i + w_p \times Position_i + w_s \times SimToQ_i \quad (6.9)$$

where $w_e, w_p, w_s \geq 0$ are the respective combination parameter of each feature.

6.3.2.5 Selection Criteria

Although redundancy problem is expected to be alleviated by using subtopic-based feature with the clustering technique, a simple reranker may still be helpful in the final selecting of sentences to form the summary. We use the default reranker in MEAD in our experiment, which sorts the sentences by *Sig* (or *SigQ*) score computed in Eq. (6.8) (or Eq. (6.9)) in a descendent order, and successively decides whether to add the next sentence to the summary. The sentence is added to the summary if the current summary length does not exceed the required limitation and the similarities between the candidate sentence to the previous selected ones are all below a given threshold, e.g. we set this threshold to 0.5 in our experiment.

6.4 Experimental Results

In this section, we carry out experimental studies to evaluate the proposed approach compared with other existing summarization systems. Both generic and task-focused multi-document summarization are conducted on DUC (Document Understanding Conferences)² datasets. Our experimental study mainly focus on two aspects, namely the effectiveness of *Exemplar* and the positive contribution of *Position* in SumCR.

6.4.1 Systems Evaluated

We first compare the proposed SumCR with two most related approaches MEAD and SNMF. Two important features used by MEAD are *Centroid* and *Position*. Compared to the subtopic-level feature *Exemplar* used in SumCR, which captures different aspects of the same topic and can be calculated based on sentence similarities measured with any advanced techniques, *Centroid* in MEAD does not capture subtopics and is defined based on the “bag of words” sentence representation. Another approach SNMF only relies on cluster-based feature to rank sentences. More comparison is given between SumCR and several other systems, such as supervised HybHSum, model-based HIERSUM, graph-based DrS (-G and -Q) and Top 3 DUC participated systems. The lowest ranked human expert and baselines provided by DUC are also compared. A description of each system can be found in Table 6.1.

Here for SumCR, we simply use $K = 3$ for all the topics of DUC2004 and DUC2006, and $K = 4$ for all the topics in DUC2005. We set $\alpha = 0.4$ (parameter in Eq. (6.3)) in all the experiments. The results of SNMF, HybHSum, HIERSUM, and DrS-G and DrS-Q are quoted from the corresponding original studies, where confidence interval is not given. For HybHSum and HIERSUM, only the results on DUC2006 have been reported in the original papers. The results of SNMF are reported in [133] on DUC2005 and DUC2006 where two types of sentence similarity are tested, i.e., the bag-of-words representation based (keyword) and semantic analysis based (SLSS). We implemented

²In the last few years, DUC <http://duc.nist.gov/> has been established as a system evaluation competition for researchers to compare the performance of different summarization approaches on common datasets.

Table 6.1: Summarization systems

System ID	Description
SumCR-G, SumCR-Q	Proposed approach
MEAD [132]	A centroid-based approach
SNMF [133]	A clustering-based approach
HybHSum [153]	A probability model-based approach
DrS-G, DrS-Q [154]	A graph-based approach
HIERSUM [155]	A hierarchical LAD-style model
Human-letter	The worst human performance provided by DUC
System-number	Top 3 automatic systems in DUC competition
Baseline	The baseline system used in DUC

Table 6.2: Summary of the summarization tasks and datasets

	Generic	Task-focused	
	DUC2004	DUC2005	DUC2006
Number of topics	50	50	50
Number of documents of each topic	10	25 ~ 50	25
Data Source	TDT	TREC	AQUAINT
Summary Length	665 bytes	250words	250words

SNMF on DUC2004 to get its results on this dataset with keyword-based similarity as we used in SumCR. For SumCR and DrS, “-G” refers the version for generic summarization and “-Q” for query-focused summarization.

6.4.2 Data Sets and Preprocessing

Document Understanding Conferences (DUC) have been a forum for researchers in text summarization since 2001. Different tasks are assigned every year for researchers taking part into the DUC competition. The performance of different approaches for the same task are compared on the same benchmark datasets with common evaluation metrics. We use DUC2004 for generic summarization and DUC2005, DUC2006 for task-focused summarization. A summary of the tasks and datasets is given in Table 6.2. For each topic, documents are segmented into sentences and each narration section in the user profile in DUC2005 and DUC2006 is treated as a single query sentence. Since the focus of this paper is not on sentence similarity measure, we use the simple statistical vector space model to present each sentence as a vector. Cosine

coefficient based on the tf-isf (term frequency-inverse sentence frequency) weighting [156] is calculated as the pairwise sentence similarity. The indexing process is done by *Rainbow* toolkit [130] with default settings, i.e. stop-words removing but no stemming and feature selection.

6.4.3 Evaluation Metric

The quality of a summary is evaluated with the ROUGE toolkit developed by [157], which is adopted by DUC for automatic summarization evaluation. ROUGE measures summary quality by counting the overlapping units such as n-gram, word sequences and word pairs between the candidate summary and a reference summary (summaries). The former is often referred as an automatically generated summary while the latter is normally produced by human experts. ROUGE-N is an n-gram recall measure computed as

$$ROUGE_N = \frac{\sum_{S \in \{ref\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ref\}} \sum_{gram_n \in S} Count(gram_n)} \quad (6.10)$$

where n represents the length of the n-gram, $Count_{match}(gram_n)$ stands for the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries and $Count(gram_n)$ is the number of n-gram in the reference summaries. Here we report the mean value as well as 95% confidence interval over all topics of the recall scores of ROUGE-2 and ROUGE-SU4 (skip-bigram plus unigram) [158].

6.4.4 Comparison of Results

We first compare SumCR with MEAD and SNMF. Table 6.3 shows the ROUGE scores of three approaches on three datasets and Table 6.4 gives the improvements of SumCR with respect to SNMF and MEAD respectively on each dataset. From these two tables, it can be seen that SumCR performs consistently much better than the other two approaches for both generic and task-focused summarizations. Since both SumCR and MEAD use the *Position* feature, the improvement of SumCR compared with MEAD is attributed to *Exemplar*. Specifically, this indicates that the subtopic-level feature *Exemplar* in SumCR possibly works better than *Centroid* in MEAD in improving the summary quality. With the same keyword-based similarity, the results

Table 6.3: Comparison of SumCR, MEAD and SNMF

DUC2004		
System ID	ROUGE-2	ROUGE-SU4
SumCR-G	0.0965 [0.0871 - 0.1059]	0.1364 [0.1278 - 0.1453]
MEAD	0.0930 [0.0852 - 0.1011]	0.1319 [0.1253 - 0.1384]
SNMF (keyword)	0.0840 [0.0756 - 0.0916]	0.1266 [0.1187 - 0.1342]
DUC2005		
System ID	ROUGE-2	ROUGE-SU4
SumCR-Q	0.0700 [0.0620 - 0.0780]	0.1251 [0.1170 - 0.1330]
MEAD	0.0688 [0.0624 - 0.0795]	0.1232 [0.1155 - 0.1310]
SNMF (SLSS)	0.0604 [-]	0.1230 [-]
SNMF (keyword)	0.0571 [-]	0.1145 [-]
DUC2006		
System ID	ROUGE-2	ROUGE-SU4
SumCR-Q	0.0906 [0.0827 - 0.0988]	0.1437 [0.1364 - 0.1514]
SNMF (SLSS)	0.0855 [-]	0.1398 [-]
SNMF (keyword)	0.0830 [-]	0.1319 [-]
MEAD	0.0732 [0.0657 - 0.0808]	0.1235 [0.1159 - 0.1313]

Table 6.4: Improvement of SumCR with respect to MEAD and SNMF

DUC2004		
	ROUGE-2	ROUGE-SU4
MEAD	3.76%	3.41%
SNMF(keyword)	14.88%	7.74%
DUC2005		
	ROUGE-2	ROUGE-SU4
MEAD	1.74%	1.54%
SNMF (SLSS)	15.89%	1.71%
SNMF(keyword)	22.59%	9.26%
DUC2006		
	ROUGE-2	ROUGE-SU4
MEAD	23.77%	16.36%
SNMF (SLSS)	5.96%	2.79%
SNMF(keyword)	9.16%	8.95%

Table 6.5: More comparisons of SumCR with state-of-the-art approaches

DUC2004				
System ID	ROUGE-2		ROUGE-SU4	
Human-G	0.0854	[0.0730 - 0.0979]	0.1291	[0.1188 - 0.1398]
SumCR-G	0.0965	[0.0871 - 0.1059]	0.1364	[0.1278 - 0.1453]
System-65	0.0920	[0.0830 - 0.1000]	0.1331	[0.1256 - 0.1404]
DsR-Q	0.0872	[0.0780 - 0.0960]	0.1290	[0.1209 - 0.1364]
System-104	0.0857	[0.0782 - 0.0934]	0.1294	[0.1231 - 0.1361]
System-35	0.0835	[0.0741 - 0.0932]	0.1286	[0.1215 - 0.1362]
Baseline	0.0640	[0.0555 - 0.0732]	0.1029	[0.0961 - 0.1101]
DUC2005				
System ID	ROUGE-2		ROUGE-SU4	
Human-H	0.0887	[0.0780 - 0.1011]	0.1487	[0.1378 - 0.1606]
DsR-Q	0.0771	[0.0734 - 0.0808]	0.1337	[0.1303 - 0.1373]
System-15	0.0727	[0.0657 - 0.0795]	0.1318	[0.1247 - 0.1392]
System-17	0.0719	[0.0632 - 0.0805]	0.1299	[0.1214 - 0.1383]
SumCR-Q	0.0700	[0.0620 - 0.0780]	0.1251	[0.1170 - 0.1330]
System-10	0.0699	[0.0623 - 0.0776]	0.1253	[0.1186 - 0.1322]
Baseline	0.0402	[0.0330 - 0.0481]	0.0870	[0.0769 - 0.0971]
DUC2006				
System ID	ROUGE-2		ROUGE-SU4	
Human-A	0.1032	[0.0911 - 0.1151]	0.1679	[0.1601 - 0.1764]
System-24	0.0957	[0.0877 - 0.1040]	0.1552	[0.1473 - 0.1628]
System-15	0.0912	[0.0835 - 0.0994]	0.1474	[0.1403 - 0.1547]
HybHSum	0.0910	[-]	0.1510	[-]
SumCR-Q	0.0906	[0.0827 - 0.0988]	0.1437	[0.1364 - 0.1514]
System-12	0.0899	[0.0819 - 0.0978]	0.1475	[0.1396 - 0.1549]
DsR-Q	0.0899	[0.0857 - 0.0943]	0.1427	[0.1391 - 0.1464]
HIERSUM	0.0860	[-]	0.1430	[-]
Baseline	0.0495	[0.0419 - 0.0575]	0.0979	[0.0896 - 0.1064]

of clustering-based SNMF is much worse than those of SumCR. The performance of SNMF is improved considerably when semantic similarity is used instead of the keyword-based one.

Other than these two approaches, we compare SumCR with more other approaches in Table 6.5. For each year, there are over 30 systems have participated in DUC competition, and here we only compare with the top 3 systems. From this table, it is observed that although based on a simple similarity measure, SumCR-G achieves quite good results compared with those state-of-the-art approaches. SumCR-G produces the best result on DUC2004 for generic summarization, where the improvements of SumCR-G with respect to other approaches are significant. In task-focused summarization on DUC2005 and DUC2006, the performance of SumCR-Q ranks at 4 or 5th among all compared automatic summarizers. It can be seen that the results of SumCR-Q on DUC2005 is comparable to the 3rd best system participated in DUC2005. On DUC2006, the result of SumCR-Q is slightly better than DsR-Q and HIERSUM, and is close to the result of HybHSum, which needs to be trained by manually generated summaries.

Here, all the results of SumCR-Q are produced based on a simple similarity measure and the query information is only incorporated in a naive way. The keyword-based similarity used in SumCR-Q fails to capture the true similarity between two sentences which consist of different words but convey similar meanings, therefore, many approaches such as System-15 and System-17 in DUC2005 use semantic similarity. The effectiveness of semantic similarity over keyword-based similarity also has been illustrated through the results of SNMF in Table 6.3. In task-focused summarization, the way of query information extraction from the task description directly affects the final summary. In System-15 participated in DUC2006, query terms are selected with a particular functional words list and stop-word list. Since a query may only contain a small number of words, query expansion [159] is one way to reserve and enrich the information contained in the original query. The results of SumCR-Q shown here is obtained without applying any query expansion or any term selection. We believe that SumCR-Q has the potential to achieve further improvements in its performance on task-focused summarization by incorporating sophisticated methods to make use of the query information in a more effective way.

6.4.5 Results with Different w_p

To see how two features contribute to the final results, Table 6.6 to Table 6.8 show the mean values of ROUGE scores of SumCR on three datasets by combining *Exemplar* with *Position* to various degrees. The ROUGE scores of keyword-based SNMF on each of the datasets are also listed in these three tables for the convenience of comparison.

When $w_p = 0$, the *Position* feature is totally not considered in SumCR. Comparing the results of SumCR with this setting on three datasets with those of SNMF, it shows that SumCR performs much better than SNMF on DUC2005 and DUC2006 and slightly worse on DUC2004. The difference in the performance of SumCR with $w_p = 0$ and SNMF indicates that the subtopic-level feature *Exemplar* of SumCR is more effective than the cluster-based feature used in SNMF.

From Table 6.6 to Table 6.8, it shows that on DUC2004, gradually increasing the combination weight of *Position* with w_p from 0 to 1 always leads to improvement in the results. For DUC2005, the subtopic feature alone is not effective enough. Improvement is achieved by incorporating *Position*, and the best result comes with w_p around [0.4, 0.6]. Further emphasis on *Position* with $w_p > 0.6$ does not help to improve the overall performance. However, for DUC2006, it can be seen that *Exemplar* alone is quite effective but improvement still can be achieved by combining with *Position* to a proper extent, e.g. $w_p = 0.2$. These results show that scoring sentences by taking into consideration of different types of features, e.g. *Exemplar* and *Position*, usually gives a more consistent good performance. However, even each individual feature is well defined, finding the proper combination weight for each feature is still not trivial. More efforts are needed in the future to work towards a sophisticated solution for this problem.

6.5 Summary

We have proposed a new extractive approach SumCR for multi-document summarization where clustering is used to discover subtopics. The core part of SumCR is the weighting of the *Exemplar* feature based on the clustering result of Fk-Parts. The Fk-Parts is a similarity-based soft clustering approach. A unique property of Fk-Parts

Table 6.6: Results of SumCR-G on DUC2004 with $w_e = 1$ and various w_p

w_p	ROUGE-2	ROUGE-SU4
0.0	0.0802	0.1250
0.2	0.0856	0.1272
0.4	0.0900	0.1315
0.6	0.0917	0.1334
0.8	0.0942	0.1343
1.0	0.0965	0.1364
SNMF	0.0840	0.1266

Table 6.7: Results of SumCR-Q on DUC2005 with $w_e = w_s = 1$ and various w_p

w_p	ROUGE-2	ROUGE-SU4
0.0	0.0668	0.1213
0.2	0.0681	0.1226
0.4	0.0700	0.1251
0.6	0.0696	0.1260
0.8	0.0693	0.1240
1.0	0.0695	0.1231
SNMF	0.0571	0.1145

Table 6.8: Results of SumCR-Q on DUC2006 with $w_e = w_s = 1$ and various w_p

w_p	ROUGE-2	ROUGE-SU4
0.0	0.0884	0.1419
0.2	0.0906	0.1437
0.4	0.0857	0.1380
0.6	0.0826	0.1342
0.8	0.0788	0.1300
1.0	0.0769	0.1273
SNMF	0.0830	0.1319

that differentiate it from others is that it produces the prototype weight of each sentence together with the sentence clusters. The prototype weight reflects within-cluster representativeness. Therefore, it provides an accurate measure of the relevance of each sentence in a subtopic it is most related to. With *Fk*-Parts, our proposed summarizer SumCR maintains the advantage of conveniently integrating any sentence similarity measuring techniques, but also makes use of the prototype weight for producing a more effective subtopic-level feature *Exemplar*. Other than that, we incorporated the document-level feature *Position*, which enables SumCR to work consistently better than approaches using only cluster-based features.

Among over 30 DUC participated systems and several other state-of-the-art summarization systems, our method gives the best result in generic summarization on DUC2004 and ranks 4th or 5th on DUC2005 and DUC2006 for task-focused summarization. We also see the great potential of SumCR-Q from several aspects, for example: (1) in query-based summarization, SumCR-Q has not incorporated query expansion, query term selection or any other query processing techniques; (2) instead of using a more sophisticated similarity, SumCR-Q at moment only uses keyword-based similarity measure. We therefore believe that in the future, the performance of SumCR-Q possibly would be further improved by incorporating some advanced query processing techniques and sophisticated sentence similarity measures.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Data clustering is a crucial technique in different data mining applications. Clustering can offer many potential benefits to a wide-range of fields including data mining, pattern recognition, information retrieval, bioinformatics, and business intelligence. Different requirements and challenges need to be taken into consideration in designing clustering algorithms applied to different tasks. In this thesis, we focused our study on clustering of document data. Other than the text included in each of the documents, other types of data may also be available and can be used to assist document clustering.

We provided a literature review of representative existing approaches of document clustering as well as typical data formats that used by clustering algorithms. Based on the analysis of challenges in clustering of document datasets with various properties and the limitations of existing approaches, we discussed why we need a new fuzzy relational clustering framework for document data analysis. We found that relation-based data representation provides a more flexible and effective way than the vector-based format for description of document data possibly from multiple angles. We have also conducted detailed studies on some existing clustering models that are relevant to our work, such as (fuzzy) k-medoids, fuzzy c-means (FCM), and matrix-factorization based multi-type relational data clustering. We have discussed their application domain, advantages, and disadvantages, and concluded that, relation-based fuzzy clustering is a more suitable framework for document categorization and analysis.

With this objective in mind, we first consider a basic situation where only the pairwise document relation is available. This document-document relation could be derived in different ways. It could be content-based document similarity, or the link structure among documents, or derived with combinations of different types of data. We formulated the partitioning fuzzy clustering *Fk*-Parts to generate document clusters with the pairwise document relation matrix. This approach is a *k*-medoids like approach that produces not only fuzzy clusters but also weighted medoids or representative objects for each cluster. Compared with existing (dis)similarity-based fuzzy clustering, where no medoid is defined or each cluster is represented by one medoid, *Fk*-Parts has been shown to be more effective and informative with the new strategy of using multiple weighted medoids for cluster representation. This approach has been successfully applied in an automatic multi-document summarization system, where summaries are generated with subtopic-level relevance defined based on the sentence clustering results produced with *Fk*-Parts.

In another approach LinkFCM, we consider to incorporate relational information into the popular fuzzy *c*-mean clustering. While fuzzy *c*-means produces clusters of objects with vector-based data, the LinkFCM approach considers information both in the form of vector-based data and pairwise relation in clustering. Other than data-derived relation, we also consider user-specified relation or pairwise constraints. When the relation matrix is constructed from given pairwise constraints, LinkFCM on one hand groups those documents which are similar in content into the same cluster, and on the other hand produces clusters of documents without conflicting with those specified constraints as much as possible, i.e., try to label two documents as the same cluster if they are specified to be “must-link” and label two documents as different clusters if they are specified to be “cannot-link”.

Both *Fk*-Parts and LinkFCM only generate document clusters at the end of clustering. To handle more general cases where the document dataset is described by multiple types of data from different angles or views, we proposed the third approach FC-MR. This is a fuzzy approach for clustering of multi-type relational data with various structures. We treat document data as multi-type relational data, where each relation matrix encodes one type of data or one view. FC-MR produces clusters for

each type of objects based on relations among them. Other than producing clusters of each type, FC-MR also generates the relative representativeness of objects of each type with respect to each cluster. The main idea of FC-MR is to estimate the fuzzy memberships of objects in each cluster based on the ranking values of their related objects in that cluster, and the memberships are in turn used for updating the rankings of objects in each cluster.

Fig. 7.1 shows the input and output of each of the proposed three approaches. Fig. 7.2 shows the connections between the proposed approaches and existing ones. It is shown that when only the pairwise document relation is given, FC-MR is equivalent to Fk -Parts; when only a relation between document and one other object type is given, FC-MR is simplified to a fuzzy co-clustering approach. Unlike co-clustering, which is specifically designed for high dimensional data, LinkFCM is able to handle data with any dimensions. This difference indicates that FC-MR may be more effective for high dimensional data clustering while LinkFCM has a more broad application domain.

Through our experimental studies, we observed the following results:

- Using multiple representative objects or medoids with different weights is more powerful than using a single medoid for capturing clusters with various structures. This is shown from the improvement made by Fk -Parts compared to traditional k -medoids approaches in document clustering.
- Making use of additional information in the clustering process is usually helpful to improve the final clustering results. This is indicated through the comparison between content-based result and content+link based result produced by the same clustering algorithm.
- Considering different types of information in the formulation of clustering is usually more effective than applying an existing clustering approach on pre-transformed data. This is shown from the results that treating text and links as two types of data in clustering, LinkFCM and FC-MR perform better than existing approaches such as HFCM on a combined data matrix.

Figure 7.1: Comparison of input and output of the three proposed approaches: Fk -Parts, LinkFCM and FC-MR.

- The quality of information contained in a relation matrix directly affects the clustering result. This is shown that even with the same vector-based data, the performance of LinkFCM is better when a relation matrix derived based on the true class labels is used instead of a data-derived relation matrix.
- The proposed fuzzy approach is more effective and efficient than matrix factorization based and model-based approaches in the literature.
- Subtopic-level feature defined based on sentence clustering results of Fk -Parts is effective for producing good quality of summaries.

To conclude, our work on fuzzy document clustering aims to effectively make use of multiple types of data together to improve the document clustering results, provide interpretive information to describe each of the document clusters produced, and maintain a low complexity.

Figure 7.2: Connections between the proposed approaches and existing (fuzzy) clustering approaches.

7.2 Future Work

In this section, we discuss some possible future work based on the fuzzy relational clustering framework that we have proposed. We are mostly interested in several directions as below:

- Integration of domain knowledge. Our studies on clustering in this thesis mainly fall into the classic domain of clustering, i.e. unsupervised learning. Since unsupervised learning approaches do not require labeled data in the learning process, they can be used to handle a large amount of unlabeled data without expensive human labor. This is the main advantage of unsupervised learning compared with supervised learning, which rely on a training dataset labeled by human experts. However, the reduced human effort usually has the trade off of the effectiveness in the performance. Supervised learning approaches typically achieve better results than unsupervised ones. Taking both the effectiveness and the cost into consideration, researchers proposed the so-called “semi-supervised

learning”, i.e., learning with a small amount of supervision together with a large unlabeled dataset. In Chapter 4, we provided theoretical analysis to show that the LinkFCM approach with user-provided “must-link” and “cannot-link” relationships of a small number of pairs of objects is a constrained fuzzy c-means or semi-supervised fuzzy c-means. Our experimental results demonstrated the usefulness of such kind of supervision in improving the clustering results. Since pairwise constraints can be naturally treated as a particular type of relation, it can be readily integrated into FC-MR. In the future, we may consider other forms of domain knowledge or supervision and incorporate it into our approaches.

- **Cluster Ensemble.** Like most partitioning clustering approaches, the three proposed approaches are sensitive to initialization and need to be specified the number of clusters k . In other words, when starting with different initializations or setting k to different values, our proposed approaches possibly give different clustering results even for the same dataset. Some studies in the literature such as [160–164], work on good initialization strategies, and some studies under the topic of cluster validity, work on techniques to find the true number of clusters in a dataset. However, most of these approaches are designed to be effective for some specific type of clustering methods, and may require a high computational cost. A recent technique called cluster ensemble, which is inspired from classifier ensemble, is shown to be promising for handling the sensitivity issues in initialization and the number of clusters. Cluster ensemble takes the clusters produced by a clustering algorithm such as k-means with different initializations and different k values as base clusters. A consensus process is then performed to produce a final partitioning of the dataset. Along with a better stability and robustness, we expect the final clustering results of the proposed approaches to be further improved by using cluster ensemble techniques.
- **Online and incremental algorithms.** So far, we only developed algorithms in a batch mode for a given dataset. To handle incremental data and online data, online versions of the proposed algorithms need to be further developed.

- **Convergence of algorithms.** Currently, the algorithms with guaranteed convergence is obtained with the constrained optimization. Considering the complexity of the solutions obtained in this way, the simple clipping strategy is applied to deal with negative values and it works well in practice. However, convergence is not guaranteed using clipping. It is a good future direction to look for other possible ways in the light of the Fixed-point Iteration Method to solve the problem with simple algorithms with guaranteed convergence.
- **Other Applications.** In this thesis, we applied our proposed approaches for document categorization. In the future, we may explore more applications of our approaches, including Web related applications, such as social network mining, biological data analysis, such as protein complex mining, pattern recognition, such as image pattern discovery, and so on.

Author's Publication List

International Journals

- (1) Jian-Ping Mei and Lihui Chen, "A Fuzzy Framework for Multi-type Relational Data Clustering", *IEEE Trans. Fuzzy Systems*, Accepted and available online through IEEE Explore.
- (2) Jian-Ping Mei and Lihui Chen, "SumCR: A New Subtopic-Based Extractive Approach for Text Summarization", *Knowledge and Information Systems*, Accepted and available online through SpringerLink.
- (3) Jian-Ping Mei and Lihui Chen, "Fuzzy Relational Clustering Around Medoids: A Unified View", *Fuzzy Sets and Systems*, vol. 183, pp. 44-56.
- (4) Jian-Ping Mei and Lihui Chen, "Fuzzy Clustering with Weighted Medoids for Relational Data", *Pattern Recognition*, vol. 43, pp 1964-1974, 2010.

International Conferences

- (1) Jian-Ping Mei and Lihui Chen, "Document clustering around weighed medoids", in *Proc. ICICS 2011*.
- (2) Jian-Ping Mei and Lihui Chen, "An Enhanced Fuzzy C-means Clustering Using Relational Information", in *Proc. FSKD'11*, pp 1090-1094.
- (3) Jian-Ping Mei and Lihui Chen, "Fuzzy clustering approach for star-structured multi-type relational data", in *Proc. FUZZ-IEEE 2011*, pp 2500-2506.
- (4) Jian-Ping Mei and Lihui Chen, "EC-MMR: Revised Exemplar-Based Clustering with Automatic Parameter Estimation Techniques" in *Proc. ICICS 2009*, pp. 1-5.

- (5) Jian-Ping Mei and Lihui Chen, "Entropy-based robust fuzzy clustering of relational data", In *Proc. IEEE SMC 2008*, pp. 385-390.

Under Review

- (1) Jian-Ping Mei and Lihui Chen, "Incorporation of Relational Information into Fuzzy c-Means", *Pattern Recognition*, Revision.

References

- [1] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley, 1990.
- [2] G. Salton, “Cluster search strategies and the optimization of retrieval effectiveness,” in *The SMART Retrieval System* (G. Salton, ed.), pp. 223–242, N. J.: Prentice Hall Englewood Cliffs, 1971.
- [3] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, “Scatter/gather: A cluster-based approach to browsing large document collections,” in *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 126–135, 1992.
- [4] O. Zamir and O. Etzioni, “Web document clustering: A feasibility demonstration,” in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 46–54, 1998.
- [5] S. O. C. Carpineto, G. Romano, and D. Weiss, “A survey of web clustering engines,” *ACM Computing Surveys*, vol. 41, no. 3, pp. 17:1–17:38, 2009.
- [6] I. S. Dhillon, “Co-clustering documents and words using bipartite spectral graph partitioning,” in *Proceedings of the 7th SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 269–274, 2001.
- [7] 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, no. 10, pp. 1279–1296, 2004.

REFERENCES

- [8] F. Wang, T. Li, and C. Zhang, “Semi-supervised clustering via matrix factorization,” in *Proceedings of SIAM International Conference on Data Mining*, pp. 1–12, 2008.
- [9] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, “Cluster analysis and display of genome-wide expression patterns,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, pp. 14863–14868, 1998.
- [10] A. Ben-Dor, R. Shamir, and Z. Yakhini, “Clustering gene expression patterns,” *Journal of Computational Biology*, vol. 6, no. 3/4, pp. 281–297, 1999.
- [11] J. Xu, H. Xiong, S. Y. Sung, and V. Kumar, “A new clustering algorithm for transaction data via caucus,” in *Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 551–562, 2003.
- [12] F. Giannotti, C. Gozzi, and G. Manco, “Clustering transactional data,” in *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 175–187, 2002.
- [13] M. Ester, H. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.
- [14] J. Sander, M. Ester, H. Kriegel, and X. Xu, “Density-based clustering in spatial databases: The algorithm gdbscan and its applications,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 169–194, 1998.
- [15] J. Heer and E. Chi, “Identification of web user traffic composition using multi-modal clustering and information scent,” in *Proceedings of SIAM ICDM Workshop on Web Mining*, pp. 51–58, 2001.
- [16] Q. Mei, D. Cai, D. Zhang, and C. Zhai, “Topic modeling with network regularization,” in *Proceeding of the 17th international conference on World Wide Web*, pp. 101–110, 2008.

REFERENCES

- [17] Y. Sun, J. Han, J. Gao, and Y. Yu, “iTopicModel: Information network-integrated topic modeling,” in *Proceedings of the 9th IEEE International Conference on Data Mining*, pp. 493–502, 2009.
- [18] G. Salton and M. J. McGill, *Introduction to Modern Retrieval*. McGraw-Hill Book Company, 1983.
- [19] S. M. Eissen, B. Stein, and M. Potthast, “The suffix tree document model revisited,” in *Proceedings of the 5th International Conference on Knowledge*, pp. 596–603, 2005.
- [20] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988.
- [21] P. Merlo, J. Henderson, G. Schneider, and E. Wehrli, “Learning document similarity using natural language processing,” *Linguistik online*, vol. 17, no. 5, pp. 99–115, 2003.
- [22] S. Shehata, F. Karray, and M. S. Kamel, “An efficient concept-based mining model for enhancing text clustering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1360–1371, 2010.
- [23] A. Hotho and S. S. G. Stumme, “Wordnet improves text document clustering,” in *Proceedings of Semantic Web Workshop, the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 541–544, 2003.
- [24] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou, “Exploiting wikipedia as external knowledge for document clustering,” in *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 389–396, 2009.
- [25] H. Chim and X. Deng, “A new suffix tree similarity measure for document clustering,” in *Proceedings of the 16th International Conference on World Wide Web*, pp. 121–129, 2007.

REFERENCES

- [26] Y. Wang and K. M., “Use link-based clustering to improve web search results,” in *Proceedings of the 2nd International Conference on Web Information Systems Engineering*, pp. 115–124, 2001.
- [27] M. Halkidi, B. Nguyen, I. Varlamis, and M. Vazirgiannis, “Thesus: Organizing-web document collections based on link semantics,” *The VLDB Journal*, vol. 12, pp. 320–332, 2003.
- [28] 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, no. 1, pp. 19–45, 2002.
- [29] R. Bekkerman, R. El-Yaniv, and A. McCallum, “Multi-way distributional clustering via pairwise interactions,” in *Proceedings of the 22nd International Conference on Machine Learning*, pp. 41–48, 2005.
- [30] A. P. Singh and G. J. Gordon, “Relational learning via collective matrix factorization,” in *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 650–658, 2008.
- [31] B. long, Z. Zhang, and P. S. Yu, “A general framework for relation graph clustering,” *Knowledge and Information Systems*, vol. 24, pp. 393–413, 2010.
- [32] A. Banerjee, S. Basuy, and S. Meruguz, “Multi-way clustering on relation graphs,” in *Proceedings of the SIAM Internatinoal Conference on Data Mining*, 2007.
- [33] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Prentice Hall, 1988.
- [34] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy — The Principles and Practice of Numerical Classification*. San Francisco: W. H. Freeman, 1973.
- [35] B. King, “Step-wise clustering procedures,” *Journal of the American Statistical Association*, pp. 86–101, 1967.

REFERENCES

- [36] Y. Zhao and G. karypis, “Hierarchical clustering algorithms for document datasets,” *Data Mining and Knowledge Discovery*, vol. 10, pp. 141–168, 2005.
- [37] A. B. Y. Linde and R. Gray, “An algorithm for vector quantizer design,” *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.
- [38] J. A. Hartigan, *Clustering Algorithms*. New York: John Wiley & Sons, 1975.
- [39] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.
- [40] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton, “Competitive learning algorithms for vector quantization,” *Neural Networks*, vol. 3, no. 3, pp. 277–290, 1990.
- [41] T. Kohonen, “The self-organizing map,” *Proceedings of The IEEE*, no. 9, pp. 1464–1480, 1990.
- [42] I. S. Dhillon and D. S. Modha, “Concept decompositions for large sparse text data using clustering,” *Machine Learning*, vol. 42, pp. 143–175, 2001.
- [43] B. Schölkopf, A. Smola, and K. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [44] M. Girolami, “Mercer kernel based clustering in feature space,” *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 780–784, 2002.
- [45] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, pp. 338–353, 1965.
- [46] M. Roubens, “Pattern classification problems and fuzzy sets,” *Fuzzy Sets and Systems*, vol. 1, pp. 239–253, 1978.
- [47] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
- [48] R. K. F. Hppner, F. Klawonn and T. Runkler, *Fuzzy Cluster Analysis*. John Wiley & Sons, 1999.

REFERENCES

- [49] S. Miyamoto and M. Mukaidono, “Fuzzy c-means as a regularization and maximum entropy approach,” in *Proceedings of the 7th IEEE International Conference on Fuzzy Systems.*, pp. 86–92, 1997.
- [50] S. Miyamoto and K. Umayahara, “Fuzzy clustering by quadratic regularization,” in *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 1394–1399, 1998.
- [51] M. E. S. Mendes and L. Sacks, “Evaluating fuzzy clustering for relevance-based information access,” in *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 648–653, 2003.
- [52] R. Krishnapuram and J. M. Keller, “A possibilistic approach to clustering,” *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 98–110, 1993.
- [53] A. Massone, L. Studer, and F. Masulli, “Possibilistic clustering approach to trackless ring pattern recognition in rich counters,” *International Journal of Approximate Reasoning*, vol. 41, pp. 96–109, 2006.
- [54] F. Masulli and S. Rovetta, “Soft transition from probabilistic to possibilistic fuzzy clustering,” *IEEE Transactions on Fuzzy Systems*, vol. 14, pp. 516–526, 2006.
- [55] M. Filippone, F. Masulli, and S. Rovetta, “Applying the possibilistic c-means algorithm in kernel-induced spaces,” *IEEE Transactions on Fuzzy Systems*, vol. 18, pp. 572–584, 2010.
- [56] M. Barni, V. Cappellini, and A. Mecocci, “Comments on a possibilistic approach to clustering,” *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 393–396, 1996.
- [57] 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.
- [58] W.-C. Tjhi and L. Chen, “Dual fuzzy-possibilistic co-clustering for categorization of documents,” *IEEE Transactions on Fuzzy Systems*, vol. 17, pp. 532–543, 2009.

- [59] T. Mitchell, *Machine Learning*. New York: McGraw Hill, 1997.
- [60] P. Cheeseman and J. Stutz, “Bayesian classification (AutoClass): Theory and results,” in *Advances in Knowledge Discovery and Data Mining* (U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds.), AAAI Press/MIT Press, 1996.
- [61] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, “Clustering on the unit hypersphere using von mises-fisher distributions,” *Journal of Machine Learning Research*, vol. 6, pp. 1345–1382, 2005.
- [62] M. Steyvers and T. Griffiths, “Probabilistic topic models,” in *Latent Semantic Analysis: A Road to Meaning* (T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, eds.), pp. 424–440, Laurence Erlbaum, 2006.
- [63] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22th Annual International SIGIR Conference on Research and Development in Information Retrieval*, pp. 50–57, 1999.
- [64] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [65] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” in *Proceedings of the National Academy of Sciences of the United States of America*, pp. 5228–5225, 2004.
- [66] Z. Guo, S. Zhu, Y. Chi, Z. M. Zhang, and Y. Gong, “A latent topic model for linked documents,” in *Proceedings of the 32th Annual International SIGIR Conference on Research and Development in Information Retrieval*, pp. 720–721, 2009.
- [67] L. Hagen and A. B. Kahng, “New spectral methods for ratio cut partitioning and clustering,” *IEEE Transactions on Computer-Aided Design*, vol. 11, pp. 1074–1085, 1992.

REFERENCES

- [68] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [69] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: analysis and an algorithm,” *Advances in Neural Information Processing Systems*, pp. 849–856, 2002.
- [70] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, “A min-max cut algorithm for graph partitioning and data clustering,” in *Proceedings of the 11th IEEE International Conference on Data Mining*, pp. 107–114, 2001.
- [71] I. S. Dhillon, Y. Guan, and B. Kulis, “Weighted graph cuts without eigenvectors: a multilevel approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1944–1957, 2007.
- [72] G. Karypis and V. Kumar, “A fast and high quality multilevel scheme for partitioning irregular graphs,” *SIAM Journal on Scientific Computing*, vol. 20, pp. 359–392, 1999.
- [73] D. Lee and H. S. Seung, “Learning the parts of objects by nonnegative matrix factorization,” *Nature*, vol. 401, pp. 788–791, 1999.
- [74] W. Xu, X. Liu, and Y. Gong, “Document clustering based on non-negative matrix factorization,” in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 267–273, 2003.
- [75] D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Proceedings of Neural Information Processing Systems*, pp. 556–562, 2000.
- [76] S. Z. Li, X. W. Hou, and H. J. Zhang, “Learning spatially localized parts-based representation,” in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 207–212, 2001.

REFERENCES

- [77] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.
- [78] W. Xu and Y. Gong, “Document clustering by concept factorization,” in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 202–209, 2004.
- [79] C. Ding, T. Li, W. Peng, and H. Park, “Orthogonal nonnegative matrix tri-factorizations for clustering,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 126–135, 2006.
- [80] C. Ding, T. Li, and M. Jordan, “Convex and semi-nonnegative matrix factorizations for clustering and low-dimension representation,” Tech. Rep. LBNL-60428, Lawrence Berkeley National Laboratory, 2006.
- [81] D. Zhang, Z.-H. Zhou, and S. Chen, “Non-negative matrix factorization on kernels,” in *Proceedings of the 9th Pacific Rim international conference on Artificial intelligence*, pp. 404–412, 2006.
- [82] D. Cai, X. He, X. Wu, and J. Han, “Non-negative matrix factorization on manifold,” in *IEEE International Conference on Data Mining*, pp. 63–72, 2008.
- [83] I. Dhillon and D. M. S. Mallela, “Information-theoretic co-clustering,” in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 89–98, 2003.
- [84] A. Banerjee, I. Dhillon, J. Ghosh, S. Meruguand, and D. S. Modha, “A generalized maximum entropy approach to bregman coclustering and matrix approximation,” in *Proceeding of the 10th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pp. 509–514, 2004.
- [85] K. Kummamuru, A. Dhawale, and R. Krishnapuram, “Fuzzy co-clustering of documents and keywords,” in *Proceedings of the 12th IEEE Internatinoal Conference on Fuzzy Systems*, vol. 2, pp. 772–777, 2003.

- [86] M. M. Shafiei and E. E. Milios, “Latent dirichlet co-clustering,” in *Proceedings of IEEE International Conference on Data Mining*, pp. 542–551, 2006.
- [87] Q. Gu and J. Zhou, “Co-clustering on manifolds,” in *Proceeding of the 15th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pp. 359–368, 2009.
- [88] 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,” in *Proceeding of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 41–50, 2005.
- [89] B. Long, X. Wu, Z. Zhang, and P. S. Yu, “Unsupervised learning on k-partite graphs,” in *Proceeding of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 317–326, 2006.
- [90] B. Long, Z. Zhang, X. Wu, and P. S. Yu, “Spectral clustering for multi-type relational data,” in *Proceedings of the 23th International Conference on Machine Learning*, pp. 585–592, 2006.
- [91] 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, no. 10, pp. 1459–1474, 2010.
- [92] Y. Sun, Y. Yu, and J. Han, “Ranking-based clustering of heterogeneous information networks with star network schema,” in *Proceeding of the 15th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pp. 797–806, 2009.
- [93] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, “Constrained k-means clustering with background knowledge,” in *Proceedings of the 8th International Conference on Machine Learning*, pp. 577–584, 2001.
- [94] S. Basu, M. Bilenko, and R. J. Mooney, “A probabilistic framework for semi-supervised clustering,” in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 59–68, 2004.

REFERENCES

- [95] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, “Distance metric learning, with application to clustering with side-information,” in *Advances in Neural Information Processing Systems*, pp. 505–512, 2003.
- [96] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, “Semi-supervised graph clustering: A kernel approach,” in *Proceedings of International Conference on Machine Learning*, pp. 457–464, 2005.
- [97] X. Wang and I. Davidson, “Flexible constrained spectral clustering,” in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 563–572, 2010.
- [98] N. Grira, M. Crucianu, and N. Boujemaa, “Active semi-supervised fuzzy clustering,” *Pattern Recognition*, vol. 41, pp. 1834–1844, 2008.
- [99] H. Frigui and C. Hwang, “Fuzzy clustering and aggregation of relational data with instance-level constraints,” *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1565–1581, 2008.
- [100] T. Kohonen, *Self-Organizing Maps*. Springer-Verlag, 2001.
- [101] R. Freeman and H. Yin, “Self-organising maps for hierarchical tree view document clustering using contextual information,” in *Proceedings of the Third International Conference on Intelligent Data Engineering and Automated Learning (IDEAL’02)*, pp. 123–128, 2002.
- [102] G. Jones, A. M. Robertson, C. Santimetvirul, and P. Willett, “Non-hierarchical document clustering using a genetic algorithm,” *Information Research*, vol. 1, no. 1, 1995.
- [103] M. Cowgill, R. Harvey, and L. Watson, “A genetic algorithm approach to cluster analysis,” *Computers Mathematics with Applications*, vol. 37, pp. 99–108, 1999.
- [104] X. Cui, T. E. Potok, and P. Palathingal, “Document clustering using particle swarm optimization,” in *Proceedings of IEEE Swarm Intelligence Symposium*, pp. 185–191, 2005.

- [105] T. Yang, R. Jin, Y. Chi, and S. Zhu, “Combining link and content for community detection: a discriminative approach,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 927–936, 2009.
- [106] H. Wang, J. Yuan, and Y.-P. Tan, “Combining feature context and spatial context for image data mining,” in *Proceedings of the 11th IEEE International Conference on Data Mining*, 2011.
- [107] A. Baraldi and P. Blonda, “A survey of fuzzy clustering algorithms for pattern recognition – part 1,” *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 29, no. 6, pp. 778–785, 1999.
- [108] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Academic Press, 2006.
- [109] B. Mandhani, S. Joshi, and K. Kummamuru, “A matrix density based algorithm to hierarchically co-cluster documents and words,” in *Proceedings of the 12th International Conference on World Wide Web*, pp. 511–518, 2003.
- [110] B. Larsen and C. Aone, “Fast and effective text mining using linear-time document clustering,” in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 16–22, 1999.
- [111] Y. Zhao and G. karypis, “Empirical and theoretical comparisons of selected criterion functions for document clustering,” *Machine Learning*, vol. 3, no. 55, pp. 311–331, 2004.
- [112] A. Strehl and J. Ghosh, “Cluster ensembles—a knowledge reuse framework for combining multiple partitions,” *Journal on Machine Learning Research*, vol. 3, pp. 583–617, 2002.
- [113] S. Zhong and J. Ghosh, “A comparative study of generative models for document clustering,” in *SDW Workshop Clustering High-Dimensional Data and Its Applications*, 2003.

REFERENCES

- [114] R. T. Ng and J. Han, “Efficient and effective clustering methods for spatial data mining,” in *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 144–155, 1994.
- [115] R. Krishnapuram, A. Joshi, O. Nasraoui, and L. Yi, “Low-complexity fuzzy relational clustering algorithms for web mining,” *IEEE Transactions on Fuzzy Systems*, vol. 9, pp. 595–607, 2001.
- [116] J.-S. Zhang and Y.-W. Leung, “Approximate clustering in very large relational data,” *International Journal of Intelligent Systems*, vol. 21, pp. 817–841, 2006.
- [117] K. Lang, “NewsWeeder: learning to filter netnews,” in *Proceedings of the 12th International Conference on Machine Learning*, pp. 331–339, 1995.
- [118] I. S. Dhillon, J. Fan, and Y. Guan, “Efficient clustering of very large document collections,” in *Data Mining for Scientific and Engineering Applications* (R. G. G. Kamath and R. Naburu, eds.), Kluwer Academic, 2001.
- [119] B. Schölkopf, A. J. Smola, and K. R. Müller, “Non-linear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, pp. 1299–1319, 1998.
- [120] R. J. Hathaway, J. W. Devenport, and J. C. Bezdek, “Relational duals of the c-means clustering algorithms,” *Pattern Recognition*, vol. 22, pp. 205–212, 1988.
- [121] R. J. Hathaway and J. C. Bezdek, “NERF c-means: Non-Euclidean relational fuzzy clustering,” *Pattern Recognition*, vol. 27, pp. 429–437, 1994.
- [122] R. N. Davé and S. Sen, “Robust fuzzy clustering of relational data,” *IEEE Transactions on Fuzzy Systems*, vol. 10, pp. 713–727, 2002.
- [123] H. Frigui, C. Hwang, and F. C.-H. Rhee, “Clustering and aggregation of relational data with application to image database categorization,” *Pattern Recognition*, vol. 40, pp. 3053–3068, 2007.

REFERENCES

- [124] N. Grira, M. Crucianu, and N. Boujemaa, “Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration,” in *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 867–872, 2005.
- [125] H. Frigui and C. Hwang, “Fuzzy clustering and aggregation of relational data with instance-level constraints,” *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1565–1581, 2008.
- [126] S. Miyamoto, M. Yamazaki, and W. Hashimoto, “Fuzzy semi-supervised clustering with target clusters using different additional terms,” in *Proceedings of IEEE International Conference on Granular Computing*, pp. 444–449, 2009.
- [127] Y. Kanzawa, Y. Endo, and S. Miyamoto, “Some pairwise constrained semi-supervised fuzzy c-means clustering algorithms,” in *Proceedings of the 6th International Conference on Modeling Decisions for Artificial Intelligence*, pp. 268–281, 2009.
- [128] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, “Automating the construction of internet portals with machine learning,” *Information Retrieval*, vol. 3, no. 2, pp. 127–163, 2000.
- [129] D. Zhang, F. Wang, C. Zhang, and T. Li, “Multi-view local learning,” in *Proceedings of the 23rd national conference on Artificial intelligence*, pp. 752–757, 2008.
- [130] A. K. McCallum, “Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering,” 1996.
- [131] S. Zhu, K. Yu, Y. Chi, and Y. Gong, “Combining content and link for classification using matrix factorization,” in *Proceedings of the 30th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 487–494, 2007.
- [132] D. R. Radev, H. Jing, M. Stys, and D. Tam, “Centroid-based summarization of multiple documents,” *Information Processing and Management*, vol. 40, pp. 919–938, 2004.

REFERENCES

- [133] D. Wang, T. Li, S. Zhu, and C. Ding, “Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization,” in *Proceedings of the 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 307–314, 2008.
- [134] I. Mani, *Automatic Summarization*. John Benjamins, 2001.
- [135] H. Jing, “Sentence reduction for automatic text summarization,” in *Proceedings of 6th Applied Natural Language Processing*, pp. 310–315, 2000.
- [136] K. Knight and D. Marcu, “Summarization beyond sentence extraction: a probabilistic approach to sentence compression,” *Artificial Intelligence*, vol. 139, pp. 91–107, 2002.
- [137] H. P. Edmundson, “New methods in automatic extracting,” *Journal of the Association for Computing Machinery (ACM)*, vol. 16, pp. 264–285, 1969.
- [138] E. Hovy and C. Y. Lin, “Automated text summarization in SUMMARIST,” in *ACL’1997 Workshop on Intelligent Scalable Text Summarization*, 1997.
- [139] A. Nenkova and L. Vanderwende, “The impact of frequency on summarization,” tech. rep., Microsoft Research, MSR-TR-2005-101, 2005.
- [140] G. Erkan and D. R. Radev, “LexPageRank: Prestige in multi-document text summarization,” in *Proceedings of Empirical Methods in Natural Language Processing*, pp. 365–371, 2004.
- [141] R. Mihalcea and P. Tarau, “TextRank: Bringing order into texts,” in *Proceedings of Empirical Methods in Natural Language Processing*, pp. 404–411, 2004.
- [142] J. Kupiec, J. Pedersen, and F. Chen, “A trainable document summarizer,” in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 68–73, 1995.

- [143] M. R. Amini and P. Gallinari, “The use of unlabeled data to improve supervised learning for text summarization,” in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 105–112, 2002.
- [144] Y. Gong and X. Liu, “Generic text summarization using relevance measure and latent semantic analysis,” in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 19–25, 2001.
- [145] J. M. Conroy and D. P. O’leary, “Text summarization via hidden markov models,” in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 406–407, 2001.
- [146] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen., “Document summarization using conditional random fields,” in *Proceedings of the 20th international joint conference on Artificial intelligence*, pp. 2862–2867, 2007.
- [147] T. Nomoto and Y. Matsumoto, “A new approach to unsupervised text summarization,” in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 26–34, 2001.
- [148] R. M. Aliguliyev, “A novel partitioning-based clustering method and generic document summarization,” in *Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, pp. 626–629, 2006.
- [149] J. G. Carbonell and J. Goldstein, “The use of MMR, diversity-based reranking for reordering documents and producing summaries,” in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998.
- [150] A. Moschitti, “Syntactic and semantic kernels for short text pair categorization,” in *Proceedings of the 12th European Chapter of the Association for Computational Linguistics*, (Athens, Greece), pp. 576–584, 2009.

REFERENCES

- [151] X. Quan, G. Liu, Z. Lu, X. Ni, and L. Wenying, “Short text similarity based on probabilistic topics,” *Knowledge and Information Systems*, vol. 25, pp. 473–491, 2010.
- [152] P. Baxendale, “Machine-made index for technical literature-an experiment,” *IBM Journal of Research Development*, vol. 2, pp. 354–361, 1958.
- [153] A. Celikyilmaz and D. Hakkani-Tur, “A hybrid hierarchical model for multi-document summarization,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 815–824, 2010.
- [154] F. Wei, W. Li, Q. Lu, and Y. He, “A document-sensitive graph model for multi-document summarization,” *Knowledge and Information Systems*, vol. 22, pp. 245–259, 2010.
- [155] A. Haghighi and L. Vanderwende, “Exploring content models for multi-document summarization,” in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 362–370, 2009.
- [156] J. L. Neto, A. D. Santos, C. A. Kaestner, and A. A. Freitas, “Document clustering and text summarization,” in *Proceedings of the 4th International Conference on Practical Applications of Knowledge Discovery and Data Mining*, pp. 41–55, 2000.
- [157] C. Y. Lin and E. Hovey, “Automatic evaluation of summaries using n-gram co-occurrence statistics,” in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 71–78, 2003.
- [158] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004*, pp. 25–26, 2004.

REFERENCES

- [159] L. Zhao, L. Wu, and X. Huang, “Using query expansion in graph-based approach for query-focused multidocument summarization,” *Information Processing and Management*, vol. 45, pp. 35–41, 2009.
- [160] 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, 2002.
- [161] X. Z. Fern and C. E. Brodley, “Random projection for high dimensional data clustering: A cluster ensemble approach,” in *Proceedings of the 20th International Conference on Machine Learning*, pp. 186–193, 2003.
- [162] A. Topchy, A. K. Jain, and W. Punch, “Clustering ensembles: models of consensus and weak partitions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1866–1881, 2005.
- [163] K. Punera and J. Ghosh, “Soft cluster ensembles,” in *Advances in Fuzzy Clustering and its Applications* (J. V. de Oliveira and W. Pedrycz, eds.), John Wiley & Sons, 2007.
- [164] A. Gionis, H. Mannila, and P. Tsaparas, “Clustering aggregation,” *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, 2007.