Robust Text Mining in Online Social Network Context

Ye Wang

Submitted in total fulfilment of the requirements of the degree of

Doctor of Philosophy

November 2018

Institute for Sustainable Industries & Liveable Cities
VICTORIA UNIVERSITY

Robust Text Mining in Online Social Context

Ye Wang
Principal Supervisor: Prof. Yanchun Zhang

Abstract

Text mining is involved in a broad scope of applications in diverse domains that mainly, but not exclusively, serve political, commercial, medical and academic needs. Along with the rapid development of the Internet technology in recent thirty years and the advent of online social media and network in a decade, text data is obliged to entail features of online social data streams, for example, the explosive growth, the constantly changing content and the huge volume. As a result, text mining is no longer merely oriented to textual content itself, but requires consideration of surroundings and combining theories and techniques of stream processing and social network analysis, which give birth to a wide range of applications used for understanding thoughts spread over the world , such as sentiment analysis, mass surveillance and market prediction.

Automatically discovering sequences of words that represent appropriate themes in a collection of documents, topic detection closely associated with document clustering and classification. These two tasks play integral roles in revealing deep insight into the text content in the whole text mining framework. However, most existing detection techniques cannot adapt to the dynamic social context. This shows bottlenecks of detecting performance and deficiencies of topic models.

In this thesis, we take aim at text data stream, investigating novel techniques and solutions for robust text mining to tackle arising challenges associated with the online social context by incorporating methodologies of stream processing, topic detection and document clustering and classification. In particular, we have advanced the state-of-the-art by making the following contributions:

1. A Multi-Window based Ensemble Learning (MWEL) framework is proposed for imbalanced streaming data that comprehensively improves the classification performance. MWEL ensures that the ensemble classifier is maintained up to date and adaptive to the evolving data distribution by applying a multi-window monitoring mechanism and efficient updating strategy.

2. A semi-supervised learning method is proposed to detect latent topics from news streams and the corresponding social context with a constraint propagation scheme to adequately exploit the hidden geometrical structure as supervised information in given data space. A collective learning algorithm is proposed to integrate the textual content into the social context. A locally weighted scheme is afterwards proposed to seek an improvement of the algorithm stability.

3. A Robust Hierarchical Ensemble (RHE) framework is introduced to enhance the robustness of the topic model. It, on the one hand, reduces repercussions caused by

outliers and noises, and on the other overcomes inherent defects of text data. RHE adapts to the changing distribution of text stream by constructing a flexible document hierarchy which can be dynamically adjusted. A discussion of how to extract the most valuable social context is conducted with experiments for the purpose of removing some noises from the surroundings and efficiency of the proposed.

Doctor of Philosophy Declaration

I, Ye Wang, declare that the PhD thesis entitled *Robust Text Mining in Online Social Network Content* is no more than 100,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work.

Signature  Date 11/11/2018

Preface

This thesis research has been carried out in IT group, Institute for Sustainable Industries & Liveable Cities (the former Centre for Applied Informatics, School of Engineering and Science), Victoria University. The main contributions of the thesis are discussed in Chapters 3 - 5 and are based on the following publications and submissions:

- **Ye Wang**, Hu Li, Hua Wang, Bin Zhou and Yanchun Zhang, "Multi-Window Based Ensemble Learning for Classification of Imbalanced Streaming Data," *in Proceedings of the 16th International Conference on Web Information Systems Engineering (WISE)*, Miami, FL, USA: 78-92, ACM Press, 2017.

- Hu Li, **Ye Wang**, Hua Wang and Bin Zhou, "Multi-window Based Ensemble Learning for Classification of Imbalanced Streaming Data," *World Wide Web, Volume 20, Issue 6, Pages: 15071525*, Springer, 2017.

- **Ye Wang**, Yanchun Zhang, Bin Zhou and Yan Jia, "Semi-Supervised Collective Matrix Factorization for Topic Detection and Document Clustering," *in Proceedings of the 2nd International Conference on Data Science in Cyberspace (DSC)*, Shenzhen, China, Pages: 88-97, IEEE, 2017.

- **Ye Wang** and Yong Quan, Bin Zhou, Yanchun Zhang, Min Peng, "Topic Detection with Locally Weighted Semi-supervised Collective Learning," *in Proceedings of the 18th International Conference on Web Information Systems Engineering (WISE)*, Puschino, Russi: 562-572, ACM Press, 2017.

- **Ye Wang**, Yitong Li, Chunyang Ruan, Hua Wang, Yanchun Zhang, "Robust Topic Detection via Topic Hierarchy," *The 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, (Under review).

- **Ye Wang**, Yitong Li, Hua Wang, Yanchun Zhang, Bin Zhou, Yan Jia, "Understand Corpora Hierarchically: A Robust Document Clustering Method," *IEEE Transactions on Emerging Topics in Computational Intelligence (TETCI)*, IEEE Press, 2019 (Under review).

Acknowledgements

First and foremost, I would like express my heartiest gratitude to my supervisor, Professor Yanchun Zhang who has been a tremendous mentor in my life, for patiently guiding and encouraging me to focus on conducting high level and diverse research and for providing me with great opportunities to grow as a researcher. I would also grateful to Professor Hua Wang for the constructive advice on my research and being my Phd committee member. My thesis would have been impossible without the aid and support of you. I would especially like to thank the families of Professor Zhang and Professor Wang, Dr. Jinli Cao, Dr Lily Sun and PhD candidate Kate Wang. All of you are the family to me in Melbourne.

I would like to thank the chair of my committee, Prof. Yuan Miao, for his advice and support from the beginning of being a PhD candidature. I also like to thank Dr. Jia Rong for her time generously spent on reading my proposal and answering my questions.

I would like to thank my close colleagues, Irena Dzuteska, Dr. Le Sun, PhD candidates Sudha Mani, Supriya Angra, Denish Pandey, Jiahua Du, Jinyuan He. I would also like to thank all the members and visitors of my research group. It was great sharing the group with all of you during last four years. I am also grateful to PhD candidate Jiaying Kou(Alice) and her family, Fan Liu, Yitong Li, Dr. Hui Zheng and Limeng Zhang for those wonderful time we shared together. A very special thanks goes out to Yitong for the hope and assistance you gave in the last stage of this journey.

I acknowledge the China Scholarship Council, the Victoria University for providing me with the financial support and appropriate facilities to pursue this doctoral degree. I also would acknowledge the Spartan from the university of Melbourne where I conducted more than half of my experiments on.

I am profoundly grateful to my family, my parents, grandparents, boyfriend and every member in the big family who have always encouraged me to embark on this wonderful PhD journey and supported me spiritually and financially throughout all these years and my life in general.

And finally, last but by no means least, also to all friends I met in Melbourne, in Australia, especial to my scuba dive instructor Zhengyu Yi, my buddy Cecilia Li, Li Wang. It was fantastic to have the beautiful experience sharing with you in this country.

Ye Wang
Melbourne, Australia
November 2018

Contents

List of Figures

List of Tables

Chapter 1
Introduction

AS a conventional carrier of information and communication, the text has been the most basic paradigm that combines the human understandable sequence of characters, phrases, sentences or even paragraphs diffusing thoughts, ideas and knowledge, while literacy is a milestone for individuals. The forms of text data are manifold, such as news, emails, messages and Internet blogs that are surrounding peoples' daily lives, survey responses, marketing investigation and business reports that are covering social economics, and health records, medical prescriptions and diagnoses that are firmly bound up with public health, which embody abundant natures of natural languages, local dialects, jargon and buzzwords, including the mixture of structured and unstructured data, the inherent high-dimensionality, the high data volume and the context-sensitive semantics expression.

Statistics of May 2018 from Forbes asserted that 90% of the data in the world was generated during the last two years[1] and will be expedited with the further growth of Information and Communications Technology (ICT) industry and Internet of Thing (IoT). Arose in the development of computing science, text mining assists us in obtaining useful content and organising knowledge from information overload effectively, especially in this era of information and big data, where the speed of data generation is predicted as a factor of 10 from the year of 2013 to 2020, reaching 44 trillion gigabytes annually and the quantum of digital universe will accumulate to 163 trillion gigabytes by 2015 [2]. Among them, there is up to approximately 90% unstructured data which is not restricted to pure text, including all formats of official files, text, pictures, XML, HTML, charts,

[1] https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#12981b5460ba
[2] https://www.seagate.com/files/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf

images, audio and video files, that can be preprocessed and interpreted by text mining techniques. In addition, the contents of this tremendous growth have been much more than the volume growth itself, in which the scope of information spreading achieves a higher level in both depth and breadth due to the ease of interaction improved by smart devices over the past five years and further result in a highly complex social context over the Internet. Therefore, the researches to text mining are nonetheless not just for problems of textual content oriented but imperative for challenges of social context.

1.1 Background

In this section, we introduce the framework of text mining, followed by the discussion of the background knowledge of online social network analysis and the challenges of text mining in the social network context.

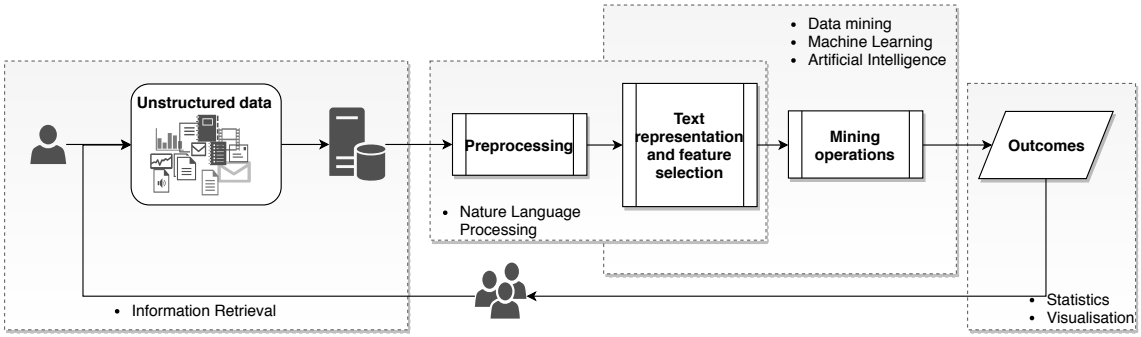


Figure 1.1: An Example of Text mining framework and some confusing terms

1.1.1 Text mining

Text mining seeks to extract meaningful knowledge and associations for domain requirement from vast collections of the unstructured data, transferring them to structured data for further usage, for example, the visualisation. It is roughly equivalent to the knowledge discovery process in which users interact with a document collection and are provided expected information based on a certain step by step process and suitable data mining techniques. Typically, text mining consists of the process of structuring and representing input data, recognising patterns and mining knowledge from structured data,

and evaluation and interpretation of rules and outcomes. A brief framework is depicted in Fig. 1.1 including the following steps.

Information Retrial is the first step that uses locate targeted data that related to their purposes and collect documents forming the input corpus. It can be manually proceeded or automatically crawled from predefined information source with certain formats. Fig. **??** gives two examples. Fig. 1.2a is an news example provided by dataset 20 Newsgroup, No.52726 including the subject, source and other meta information. Fig. 1.2b is an example of tweets message that was collected by specifying user_name and time period. The record only includes lists of tweet_id, time and hashtag message.

```
From: rajan@cco.caltech.edu (Rajan Ranga)
Subject: An external timer
Article-I.D.: gap.1pli7gINNi6b
Organization: California Institute of Technology, Pasadena
Lines: 8
NNTP-Posting-Host: fleming.caltech.edu

I was wondering if anyone knows of a chip that that is similar to
the internal timer 0 on the Intel 80C188?  I want a timer that has
a Maxcount A and B and the output should the same as Intel's timer.  I called
Intel and they told me that they don't make such a chip.  Any suggestions
are welcome.  Thanks in advance.

Rajan Ranga
E-mail: rajan@cco.caltech.edu
```

(a) An exmple of a news

```
id,created_at,hashtags
725451117333598208,2016-04-27 22:26:39,b'UCL'
724448089914765313,2016-04-25 04:00:58,b'Clarridge'
718848324111994880,2016-04-09 17:09:30,b'Espionage'
713095876961415168,2016-03-24 20:11:20,b'foia'
711547837448720385,2016-03-20 13:39:59,b'TSAwaits'
710983111543099392,2016-03-19 00:15:57,b'SunshineWeek'
710633699239272449,2016-03-18 01:07:31,b'TSAwaits'
710512586547077120,2016-03-17 17:06:15,b'TSAwaits'
710502699322617856,2016-03-17 16:26:58,b'TSAwaits'
710496835241189377,2016-03-17 16:03:40,b'TSAwaits'
710494781835747328,2016-03-17 15:55:31,b'TSAwaits'
710491264869748737,2016-03-17 15:41:32,b'TSAwaits'
```

(b) An example of tweets collection

Preprocessing is an integral step to let the natural language understandable to computers and mathematically computable. It always involves the following techniques:

- *Segmentation* is sometimes needed to identify paragraphs and sentences in a large chunk of text.
- *Tokenisation* is the process of breaking long strings of text into small pieces of words, excluding spaces, punctuation and special characters (&, #, %, $, etc.).
- *Normalisation* converts all letters into the same cases and removes punctuation. This is necessary because the upper case and lower case of one letter have different codes in the codding system.
- *Stop word removal* removes words that are meaningless for the evaluation of thse document content, like ‘a’, ‘the’, ‘of’, etc. Stop words usually refer to most common words in a language, for example, there is a fairly complete English Stop Word List for English.
- *Stemmin g and Lemmatization*. Stemming removes tenses and suffix of a word, reducing the words to their stems; while lemmatization identifies the base form of a

word.

- *Pruning* discards terms either rarely appearing or too frequently because terms that rarely appear in a document or terms that appear too frequently do not contribute to identifying the topic of the document.
- *Treating synonyms* identifies two words or terms have the same meaning and if they are synonyms we could replace them by one of them without taking the semantic meaning of the word.

Text Representation encodes documents into a proper representation with language models according to different application requirements, for example, bag of words model, n-gram model, word embedding model.

Feature Selection simplify the model with selecting a subset of relevant features that not only save the time and space complexity of following mining algorithms, but also always enhance the outcomes by reducing variance and overfitting.

Mining Operations This is the kernel stage where we apply algorithms on specific application tasks to discover knowledge. Algorithms involved herein are from fields including data mining, machine learning and even artificial intelligence. Tasks involved in a broad scope of applications in diverse domains that mainly, but not exclusively, serve to political, commercial, medical and academic needs. Regarding academic study, we list the following tasks that have received considerable attention recently:

- *Document clustering and classification* algorithms automatically identify the types of documents and categorize them into different groups with the minimum error so that documents within a group highly similar comparing with others in different groups, in which classification process is the supervised learning, assigning a document to the correct class that has been learned with a training set, while clustering is more likely the unsupervised learning that there is no clue and priori knowledge at all before the corpus arrive. It is a fundamental technique that widely used in web document searching, recommendation system, customer feedback analysis, etc.
- *Topic detection* algorithms monitor the given corpus and reveal latent topics that documents tell with certain sequences of words. It dramatically reduces the labour workload of identifying the relevant text files and summarising the central themes. As an underneath knowledge discovery technique, it has become a major concern

of academic research focuses. Besides being applied in research articles, news and other traditional text files, topic detection has far more application scenarios, such as auxiliary diagnosis and clinical decision in the medical system and trend analysis in the online social network. Also, topic detection is inextricably bound up with document clustering and classification, in which the latent topics always correspond to the boundaries of documents' groups.

- *Community detection* algorithms studies the organisation structure of the network that generate the text data. It is more relevant to graph theory that identifies relationships among groups of vertices and edges. In the range of text mining, community detection was introduced to obtain cooperation network through co-author relationship in vast amounts of research papers. Recently, it is also widely used in the areas of biology, social network analysis and the Web [122] where more attentions are not content to discover the underlying mechanism of such systems, but intend to involve in further knowledge mining process. For example, inquiring about the association between emerging topics and grouping communities in the online social network.
- *Sentiment analysis* algorithms discern authors' attitude and standpoint between the lines, for example, how enjoyable a reviewer is to watch the movie and how does a custom satisfy the provided service. Intuitively, human feelings with respect to one entity are complicated more than 'yes' or 'no' and more than one entities may be mentioned together for comparison and demonstration. Therefore, it can be applied to the data at different levels and views, including document-level, sentence-level, aspect-based and comparative sentiment analysis [56]. In both academic and industry community, it is an attractive research area with extensive application prospect, ranging from custom feedback understanding to public opinion spreading prediction, especially with the advent of the Internet and smart devices.
- *Protein interactions and Gene-disease associations* are two typical tasks of biomedical text mining. Protein interactions algorithms identify associations of proteins among protein complexes, a production of protein-protein interactions, and protein-protein interactions, which can help people with comprehension of diseases and pharmaceutical research [9,123]. Gene-disease association benchmarks were devel-

oped to cope with the gene prioritization problem recognising major gene linked to genetic diseases [187].

Outcome Analysis and Visualisation. Last, sometimes there are requirements of simple, straight and declarative presentation of mining outcome as meaningful conclusions and charts, such as the annual report and the decision-making process. For this purpose, tools for further analysis and visualisation such as statistic software and link discovery tool can be used.

This thesis mainly focuses on theories and techniques of two closely related tasks, document clustering and classification and topic detection.

So far, we know that text mining is a broad concept of operations that involves series techniques in steps, but for us, some concepts that are shown in Fig. 1.1 can be confusing due to the sort of overlaps between the goals and research objects. Here, we differentiate them from their intersection with text mining.

- *Information Retrieval (IR)* finds and presents information that relevant to user's need with a specific context, for example answering an input query. It is a hybrid area using machine learning, text mining and natural language processing techniques. In web searching, IR cares more about information interaction through web search engines which is not only text though it is text most of the time.
- *Nature Language Processing (NLP)* eyes on human-computer interaction, processing natural language and generating human understandable text. It has a more concentrated focus on improving computer's human linguistic recognition and organisation capabilities so that automatically NLP system can be provided to users structuring their unstructured input for further intentions, such as being retrieved and being analysed.
- *Data Mining (DM)* is an umbrella term for techniques that reveal relations and patterns from structured data, for example, tables with columns and rows, that can be used for a variety of purposes. Data Structure is the essential difference between data mining and text mining. Therefore, the mining operations of text mining involve data mining algorithms and ideas to a great extent.
- *Machine Learning* is a kind of specific algorithms that can be implemented with computer programs to solve different formalised problems for many domains. It em-

bodies the solution depending on the context of other fields, including data mining and artificial intelligence.

- *Artificial Intelligence (AI)* aims to build cognitive machine or system that can automatically assist human activities. It is somehow the ultimate goal for all developing research and techniques. But fundamental techniques for AI are also techniques developed for the above tasks.c
- *Statistics and Visualisation* generate charts, graphs or diagrams to better demonstrate the result for users. Typically, they are intuitive analysis rather than hidden knowledge mining.

1.1.2 Online Social Network

The advent of the online social network is a product of the mutuality of ICT, smart devices and human natural social network that broaden horizons of Social Network Analysis (SNA) in sociology and anthropology. Online social network is a social structure that constructed by three factors of the Internet, where the subject is network individuals, the object is network information and the carrier is network relations. Network individuals include entities, such as individual users, organisations, and groups, and virtual individual, like a username of a web use. Relationships between network individuals range from real friends, sharing ideas, sending and receive messages to advertising cooperation. Online communities consisting of network individuals are subsets of the whole network relations, in which nodes are closely associated with each other in the inner community while there is lack connection between nodes from different communities. Based on various interactions of network nodes, the network information is spread across the network relations, which in turn affects network nodes' further interpretation reaction. Therefore, facilitated by ICT and smart devices, the social context that reflects the content of human network is dramatically complicated by an invisible hand. According to "We Are Social" company's recent statistics, there are 4.021 billion Internet users and 3.196 billion social media users by 2018 which achieves high rates of growth at 7% and 13% in the past 12 months[3]. Take the news consumption of social media in the US as an example, Pew Research Centre reported that 68% adults are audiences of news on

[3] https://wearesocial.com/uk/blog/2018/01/global-digital-report-2018

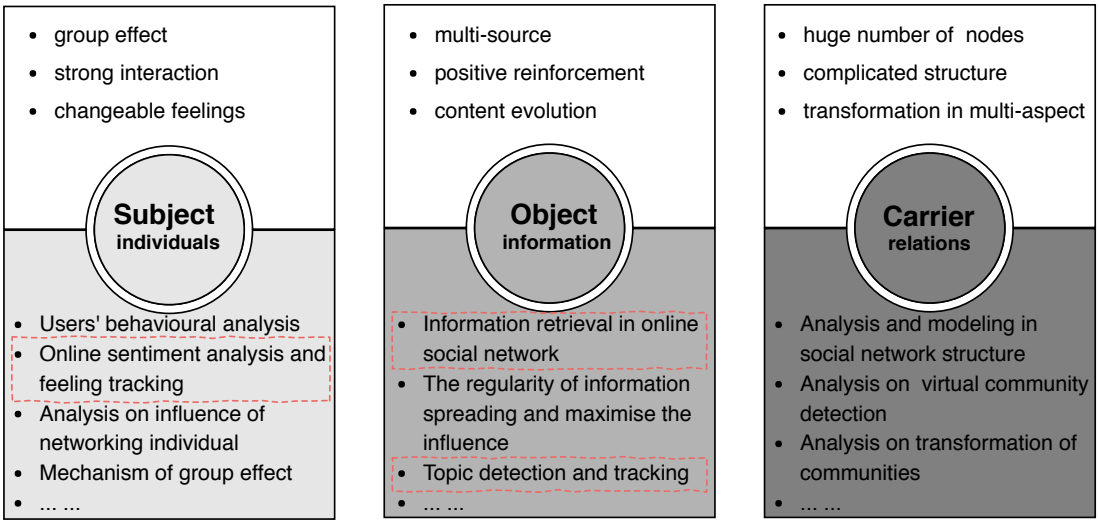


Figure 1.2: Scientific problems in Social Network Analysis

social media, among which up to 70% are under the age of 50[4].

In general, there are following peculiarities of the online social network that are most remarkable in comparison with the traditional social network:

- *Convenience in accessing to information*. What is provided by various smart devices and social network services connected by the Internet is the ease of releasing and receiving information from anywhere at any time.
- *Extremely high speed and the wide range of information spreading*. The information released from a network node will be spread exponentially that opens a way for social network users to express themselves.
- *Low cost of being an influencer*. It is a great opportunity for a social network user to be an influencer through activities in the online social network, playing a crucial role as an opinion leader in the lifecycle of a network event, including emerging, evolving and fading process.
- *Grouping together as virtual community spontaneously*. Online communities will emerge apace and spontaneously because of the reasons mentioned above.

SNA studies almost everything of network individuals, relations, information spread among them and how their interactions work, in which there is a great demand for text

[4] http://www.journalism.org/2018/09/10/news-use-across-social-media-platforms-2018/

mining techniques in selecting and refining features from the network structure because of the massively distributed unstructured data and human natural language. Though extensive and in-depth discussion has been generated in the research and industry community, many open problems to solve. Fig. 1.2 illustrates research directions in terms of the features of the above three factors of the online social network. Since SNA is not the main concern of this thesis, we do not go into greater depth on the features and scientific problems. Despite the requirement of text mining techniques for unstructured data, it can be noticed that there are many overlaps between scientific problems in the two areas, even at a superficial glance (roughly marked with red dashed box).

As text mining to SNA, the landscape of text mining has become increasingly broad with influences of the vast online network and the constantly changing data universe. Fig. 1.3 illustrates an example framework of test mining in association with the related social context. As a result, text mining techniques are no longer merely oriented to textual content itself but requires considering challenges posed by the social context, which we summarise in the following.

1.1.3 Challenges of Text Mining in Social Network Context

1. **Challenges in the vast dynamic data universe**

   The huge volume issue has been emphasised in recent years which seriously impact on the traditional data collecting and processing method. A consensus is achieved that the store-and-process model is no longer suitable for processing such a huge volume of changing data due to the limitation of storage capacity, that means an in-memory computing and abandoning mode, named as stream processing, is widely adopted. However, the existing approaches of information retrieval and mining for the targeted data cannot satisfy the demand of following this dynamic data processing mode well since most of them rely on the massive data storage to discover insightful knowledge what we refer to as data concept or distribution. Therefore, though it is impossible to implement the complete computing and abandoning mode, efficient processing and mining methods capable of dealing quickly changing concept on real-time social network data stream is needed. The second challenge is to appropriately segment data stream to achieve performance enhancing

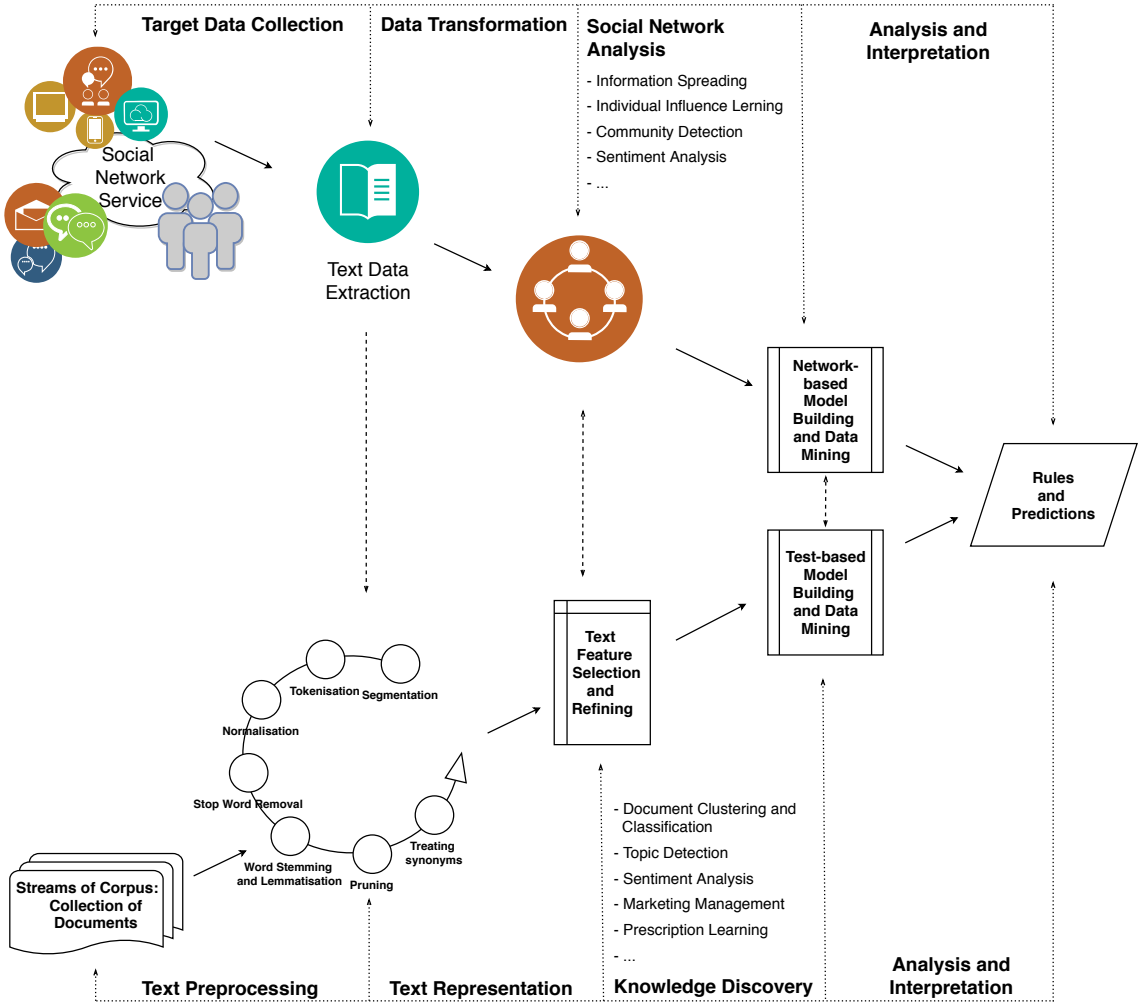


Figure 1.3: Text mining framework in social network context

for further mining operations. It is also hard to tune the update frequency for a particular tracking query without any empirical input. Nevertheless, the empirical knowledge needs long-term observation. A novel adaptive learning mechanism to track fluctuating data distribution is required for content evolving monitoring. Last but not least, it is challenging to represent and make use of the tracked traces of content evolving in the further mining tasks rather than just simply visualise them for th demonstration.

2. **Challenges in combining textual content with the social network context**

Existing text mining techniques have been developed over the years around the

content of unstructured data, text in particular. However, relations between people hidden in the online social network will also provide auxiliary information that is helpful for revealing the truth more precisely. Then the question has drawn much attention that how to leverage the surrounding social information of the targeted text data. For example, absorbing surrounding information to improve the accuracy of topic detection in social media is an emerging direction to improve the traditional topic model. Challenges here are not only lie in the huge volume and complicated structure of the online social network that highly related to challenges mentioned above, but also in some fundamental problems, such as how to formulate the context of social network, including the network individuals and relations, how to effectively select the most valuable network subset representing the really needed context and how to formalise the connection between the text data and corresponding social context. Furthermore, how to keep the form consistency of representation of social context is also a challenge for the auxiliary information obtained from the online social network.

3. **Challenges in achieving robust performance**

   When talking about the system reliability, the more data source we access to, the more varied, uncertain and unbalanced data will be involved for processing. The first challenge is to reduce repercussions of noises and outliers issue that may be aggravated by introducing auxiliary information. Noises and outliers may cause serious bias and errors which are bottlenecks of the existing mining methods. The second challenge is to overcome mutual interference of dynamic data in a specific time period and part of the network structure that will confuse boundaries of text content. The dynamic data is not merely the textual data, but also the relevant data of social context. Therefore, text mining combining with social context has a surging demand for a robust mechanism. Moreover, from the stream processing view, the fluctuating distribution of data stream also requires a performance-oriented method with minimised space and time consumption.

4. **Other challenges**

   A diverse multilingual environment is formed through the Internet, connecting

people from anywhere in the world in convenient and efficient ways, where multiple languages, local dialects, jargon, buzzwords and even emoticons are fused together that poses unprecedented challenges to the preprocessing and representation step of text data. Each language has its unique cultural connotation and melody but it is hardly telling by computers. The most commonly employed language models so far are words based models, such as bag of words model and word embedding model, which directly correlate to the model's capability of processing a particular language. That is to say, an auto multilingual text mining framework is needed. Take the online media as an example, the auto multilingual text mining framework not only applies to detecting news contents, but to analysing opinions and comments of readers as well. For the moment, this problem also can be discussed with challenges in taking advantage of language-neutral network structure. In addition, follow-up challenge of sentiment analysis across multilingual environment and putting intertwined communication in order will come.

1.2 Research Motivations and Problems

This thesis concerns the robust text mining in the online social network context, mainly focusing on techniques of topic detection and algorithms of clustering and classification with respect to above-mentioned first three challenges. The research problems of this thesis are summarised from the following aspects:

1. **How to identify the changing distribution in the data stream and maintain an up-to-date classifier for imbalanced data stream?**

   Both imbalanced data and streaming data with changing concept are pervasively distributed and appear together in various real applications; however, most existing studies still focused on either the former or the later, because it is hard to detect and react to imbalanced distribution in a time-critical continuous data stream with a space and time efficient method. For the task of classification on an imbalanced data stream, an effective monitoring mechanism of changing distribution in streams is needed to arrange with proper updating scheme for the classifier.

2. **How to combine existing text content representation with the corresponding social context in network individual level to boost the topic detection performance?**

   When giving a document released at a particular time step, the following reactions around this document on a social network service platform, for example, readers, communities, comments, time and area, reposts and mentions, etc. on Twitter can be collected, which we named as the corresponding social context. There are many ways to represent the social context from different levels and aspects. In this work, we concentrate on the individual level of the social context, aiming to find a proper combination between the input corpus and the corresponding social context for topic detection task.

3. **How to extract the inherent geometric structure from the data distribution and maximise the effect?**

   Apart from the word features of a corpus, the interrelations among data points which are often unspectacular and with little causal relations hiding in the geometric structure of the data space also helps to distinguish semantic structure [24]. They usually are not apparent in high-dimensional textual data, as well as the corresponding network context. Additionally, most of them are so weak that hardly be observed and leveraged. Therefore, inherent geometric structure mining and enhancing scheme is required.

4. **How to adaptively detect robust correct topics and implement meaningful document clustering simultaneously?**

   Since the outlier and noises always cause various negative influences to mining tasks, ranging from the low accuracy of clustering results to high information entropy of topic detection result, especially on high-dimensional dynamic data space, including our targeted text content and social context, improving the robustness of model has been the central concern. Moreover, most existing studies assume a pre-defined constant k as the specific number of latent concepts to guide the topic detection and document clustering, which is incompatible with the quickly evolving and changing scenario of topics in the present social context. Thus, an adaptive and robust topic detection method is required. As being closely associated with the

topic detection, we will also discuss the accompanied document clustering at the same time.

1.3 Thesis Contributions

The key contributions of this thesis against the problems mentioned above are listed in this section.

1. A Multi-Window based Ensemble Learning (MWEL) framework for imbalanced streaming data which comprehensively improves the classification performance:
    - A multi-window monitoring mechanism that maintains four windows for the current batch of instances, latest positive instances, sub-classifiers of the ensemble classifier and instances employed to train existing sub-classifiers, respectively.
    - An effective updating strategies for weights corresponding to existing sub-classifiers that keep the ensemble classifier being adaptive to the evolving data distribution.
    - An efficient updating strategies for sub-classifiers to guarantee the ensemble classifier up-to-date when a change of data concept is detected and renewal is necessary.
    - An impartial re-sampling mechanism for both positive and negative instances that generate new training set with an ideal imbalance ratio for updating sub-classifiers.
2. A semi-supervised collective learning method for topic detection combining the text content of the input corpus with the corresponding social context:
    - A constraint propagation scheme that adequately exploits and enhance the hidden geometrical structure, which is naturally very sparse and weak, in the given data space.
    - A user preference representation on the individual-level that formulate the social context surrounding the input corpus.

- A collective non-negative matrix factorization based topic model to combine two parts of the corpus, the textual content and the social context, of the input corpus as a whole.
- A locally weighted scheme to better approximate certain parts of the data matrix in each iteration.

3. A Robust Hierarchical Ensemble (RHE) framework for topic detection via document hierarchy in text corpus and the corresponding social context with severe outliers and noise issue:

   - A robust multiplicative updating rule based non-negative matrix factorization algorithm for document clustering in which an orthonormal constraint is added on the output cluster indicator matrix so that detected topics can be generated at the same time.
   - A top-down hierarchical algorithm to flexibly cluster documents that adapt to the changing distribution of the input corpus and reduce the dependence, achieving a logical and specific interpretation for clusters of the output.
   - A hierarchy design includes the candidature selection policy, the pruning strategy, the verification of outliers and two practical stopping criteria, which also facilitate the robustness of the ensemble framework.
   - A comparative analysis in the objective function level is conducted for the serious outliers and noise issue in complicated social context and high-dimensional text data.
   - A discussion on subset selection of network individuals that represents the really valuable social context for the purpose of removing some noises from the surroundings and efficiency of the proposed.

1.4 Thesis Structure

The organisation of this thesis is shown in Fig. 1.4. Chapter 2 provides introduction for fundamental text mining techniques and review some typical research of topic model and document clustering and classification. The literature reflecting the influence of the

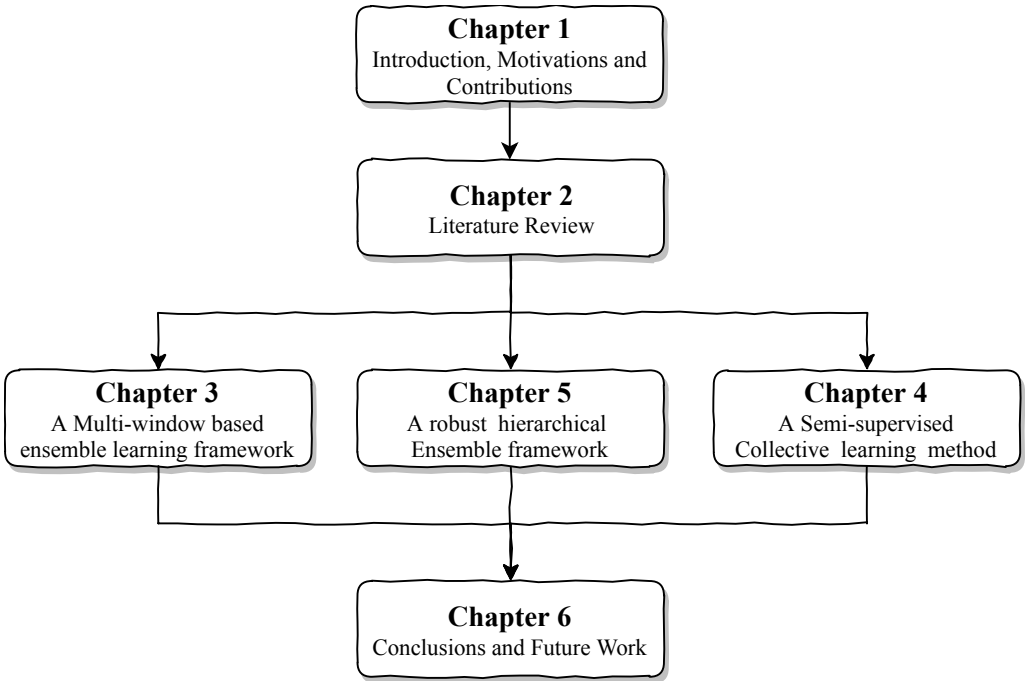


Figure 1.4: The organisation of the thesis

social network services will also be referred to. Chapter 3 presents a multi-window based ensemble learning (MWEL) framework for imbalanced streaming data. Chapter 4 proposes a semi-supervised collective learning method for topic detection combining the text content of the input corpus with the corresponding social context. Chapter 5 proposes a robust hierarchical ensemble framework for topic detection via document hierarchy in text corpus and the corresponding social context with serious outliers and noise issue. Chapter 6 concludes this thesis followed by a discussion of future directions.

Chapter 2
Literature Review

Text mining tasks extract meaningful knowledge and associations for domain requirement from vast collections of the unstructured data, mainly is natural language text. Before and after the advent of ICT and IoT, many efforts have been made to achieve better automatically understanding and adaptively learning from the complicated and irregular unstructured data patterns. In this chapter, we first introduce the key assumptions and the common thought of the text representation. Although many primary techniques and tasks have been mentioned in Subsection 1.1.1, we concentrate on the literature of state-of-the-art topic detection and document clustering and classification methods which are directly relevant to the target of this thesis, rather than review all of them here. To better understand these methods, we will briefly introduce the VSM-based text representation and language models that are widely adopted throughout the real applications as a preliminary. The literature reflecting the influence of the social network services will also be referred to.

2.1 Introduction

Text mining is a content-oriented process of comprehending people's communication. Therefore, two crucial factors determine the scope of text mining, which are the semantic space and the social relation. It should be clarified here that the social relation between subjects of communication has always been there; however, what we emphasise across the thesis is the online social context that can be regarded as an updated version of social relation promoted with the advancement of ICT, IoT and social network services. Intuitive, formulating semantic features over the input corpus is an access point, which results in a long period of development on *Text Representation* and *Language Model*.

The most typically used representation model for text document is the Vector Space Model (VSM) [140], which in some simple case also referred to as Bag of Words (BoW). The target of VSM is formulating a document d by a numerical vector so that the similar-

ity between documents can be equivalent to the similarity between vectors, for example $d = [(t_1, w_1), (t_2, w_2), \cdots, (t_n, w_n)]$, where t_i is a term representing a semantic feature and w_i is the corresponding weight. The weight can be computed with the boolean value (0-1), absolute word frequency, relative word frequency using Term Frequency-Inverse Document Frequency(TF-IDF) or others, in which the TF-IDF is the most commonly used technique. The model recognises a document as a collection by assuming that words are independent of each other and ignoring the order and distance between words. And a vital assumption established for TF-IDF is that The word that can better differentiate text categories are those with large TF-IDF value in documents of this category but with a low TF-IDF value in documents of other categories [196]. These natures lead to weaknesses of VSM that the structural and semantic information of documents cannot be well expressed [80]. Although many improved versions have been proposed based on the original VSM to cope with the weaknesses, some attentions turn to an alternative for encompassing semantic relations between words and utilising more meaningful weighted strategy in the recent decade.

Graph-based text representation model extracts linguistic objects or entities as the vertices of the graph and the semantic relations between objects as the edges forming meaningful representative substructures. The linguistic objects are not only words, phrases, sentences and paragraphs, but also semantic concepts. Chakravarthy et al. firstly proposed two types of domain-independent graph representations, tree and star representation, in which domain knowledge is required to choose words and structure forming the graph for a specific domain, for example, the email system [27]. Later on, the star representation is employed commonly. Choudhary et al. adopted a Universal Networking Language link method to represent sentences of a document with the homogenous representation [36] that based on the assumption of "The more links to and from a universal word, the more important the word is in a document". Hensman afterwards proposed conceptual graph representation methods which include two steps where semantic roles are identified by using the verb lexicon VerbNet and the lexical database WordNet in the first step followed by the second step of construction of conceptual graph with the roles and some heuristic rules [68]. Besides the structure of the graph, efforts of this type of model put forward a Degree Centrality (DEG) to replace the TF-IDF which determines

the relative importance of a vertex by considering many aspects including the order, the location and the frequency of a word [12]. Erkan et al. proposed a text summarisation method LexRank to computing sentence importance in a document or a document collection with the assessment of the Eigenvector Centrality (EVC) of each sentence [54]. EVC is calculated by the weighted sum of the EVCs of its neighbours with the idea that "a central node is connected to other central nodes". Furthermore, Betweenness Centrality (BWC) which motivated by the assumption of "a node is important if it lies in many shortest paths" [142] and Closeness Centrality (CLC) which measures the ability of a vertex to quickly pass information through the graph as the inverse of the sum of short distances between a vertex and all the neighbours [150] are introduced to weight the vertices in the graph of linguistic objects. Although the graph-based text representation models are still very young, they have been applied in many tasks [40,61,65,153,164]. However, since the representation we employed in this thesis is based on VSM, we will not go deeper into the graph-based text representation model in the following section of this chapter.

Recently, an idea of social graph-based text mining was proposed since the mutual interaction between individuals is noticeably affected by the popularity of online social network services where people join to conversations and publish their fresh opinions freely. The social graph identified from the textual communication records may be able to provide us with a new way to reorganise the sequences of the data and discover the logical connections from the sequences. For example, questions and answers during a conversation section may be disordered with the timeline and consecutive dialogues may correspond to different themes, especially conversations among multiple people. Anwar et al. proposed a social graph construction technique associated with an n-gram key-words extraction method to investigate possible criminal scenarios and groups from MSN chat messages and digital logs [8]. Their work inspired us to consider the social context the surrounds the textual content.

The rest of this chapter is organised as follows. We expand reviews on word representation and language model to start the next section, then we transfer to the language model to show how the development of language model promotes the advance of text representation model. Next, we introduce the widely applied topic models and the state-

of-the-art methods of topic detection and document clustering and classification. Last, we review the improvement of methods facilitated by SNA and stream processing.

2.2 Word Representation and Language Model

Word representation model formulates semantic characters of the natural language. As we know that the essential semantic character is the word, word vectors in an article become the first option to project linguistic features. Many works have been developed towards the word representation in the past decades, among which the one-hot representation is the most intuitive one. However, the distributed representation is a more comprehensive alternative to the one-hot representation with more possibility of being improved. No matter what kind of word representation is adopted, the obtained word vectors of the document will be concatenated with some methods to a vector which is the document representation.

2.2.1 One-hot Representation: Syntagmatic Models

Syntagmatic models concern the combinatorial relation between words, focusing on words that exist together in the same text article. With the one-hot representation, a word is formulated to a long vector with all elements valued with 0 except one element valued with 1. The non-zero element is the feature represented this word. For example, the first word "thesis"$= [0,0,0,1,0,0,\cdots,0,\cdots]^T$, and the second word "submission"$= [0,1,0,0,0,0,\cdots,0,\cdots]^T$. The length of the vector is the number of words. Then the words vector will be selected as features to fit the text representation model for further tasks, including topic detection and document clustering. However, the limitations are obvious, for example, high dimensional features and lack of semantic information between words since two vectors are always orthogonal, which will have a strong impact on the performances of further tasks.

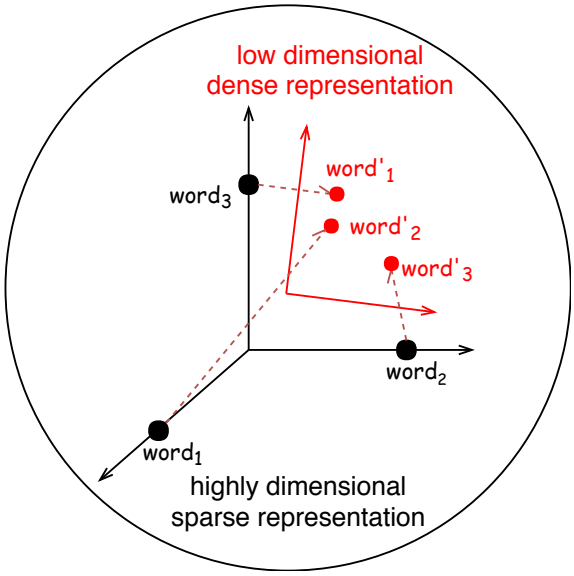


Figure 2.1: An example of comparison between a three dimensional one-hot representation and a two dimensional distributed representation.

2.2.2 Distributed Representation: Paradigmatic Models

Paradigmatic models emphasise the substitutional relation between words, focusing on words that have the similar semantic function for the context but do not have to occur in the same text region. The distributed representation is believed to be a lower-dimensional dense vector that represents a word with its semantic information. For example, the word "thesis"=$[0.2, 1.6, 0.3, -0.5, 0, \cdots]^T$ and "submission'=$[0.4, -0.3, 1.2, 1, 0.7 \cdots]^T$", where each feature may not be necessary to correspond a word existing or not, then the semantic similarity between these two words can be easily calculated. It was first proposed in [69] based on the distributional hypothesis "You shall know a word by the company it keeps." [58] which clarified the earlier opinion "the complete meaning of a word is always contextual" [57]. The word "distributed" is different from the "distributional" [1], we will explain the difference with the following different categories of methods. The only thing is that since methods of distributed representation have the same fundamental hypothesis, the main steps are the same which are:

1. Find a way to specify the context.
2. Find an expression for the relation between the targeted word and its context.

[1] http://www.marekrei.com/blog/26-things-i-learned-in-the-deep-learning-summer-school/

Normally, three categories of these methods are recognised as follows.

1. *Matrix-based distributed representation* is also referred to as distributional semantic models or distributional representation directly [11]. This kind of methods depends on a "word-context matrix" where the rows are words in the corpus, and the columns represents different contexts. Basically, it is still a count-based method beginning with the generation of a contextual matrix. Three methods are commonly used to generate the contextual matrix, which are "words - document matrix" [92], "words - words co-occurrence matrix" [132] and "words - n-grams co-occurrence matrix" [151]. Then the values of the contextual matrix are further calculated on the number of the co-occurrence with some techniques, including TF-IDF and P-MI (Pointwise Mutual Information). So far, the contextual matrix is still a high-dimensional and sparse matrix that does not satisfy the requirement of a dense matrix; therefore the dimensionality reduction is always conducted here, for example, the Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF), Principle Component Analysis (PCA) and Canonical Correlation Analysis (CCA).
2. *Clustering-based distributed representation* is also referred to as distributional clustering [133]. This kind of methods using clustering algorithms to automatically group words according to their occurrence in particular grammatical structure with other words. Brown algorithm is a classic clustering algorithm based on n-gram model and Markov chain for words clustering [88, 113]. It only considers the relationship between two adjacent words and outputs a binary tree representing whether the two words share a cluster label. The similarity between the two words depends on the common clusters. Lin et al. proposed a scalable phrase clustering method using the simple k-means algorithm later [101].
3. *Neural network-based distributed representation* is also referred to as distributed representation directly or word embedding [14]. It draws many attentions recently because it can express complicated context by using the flexible neural network learning algorithms in a lower dimensional space, i.e. each concept can be represented by many neurons, and each neuron can also participate in the representation of many concepts. Recently, this kind of learning methods goes further by taking

advantages of Deep Learning techniques. Recently, Mitra et al. presented a model combining distributed representation with local representation to train deep neural networks for the document ranking task and achieve a higher performance than the solo models do [116]. However, the disadvantages are lying two aspects, which are the training process of word embeddings is complicated and can hardly be interpreted. The training process of word embeddings is always accompanied by the training of the language model for an input corpus. Next, we will briefly introduce the language models.

2.2.3 Language Model

A Language model is an umbrella term that describes a variety of mathematical models for formulating, analysing and creating the text with natural language. Essentially language model is used to judge whether the input text is reasonable and comprehensible. It plays a vital role in tasks of information retrieval, machine translation, speech recognition, etc.

Statistical Language Model

Assume there is a sequence S containing m words $[w_1, w_2, \cdots, w_m]$, statistical language model estimates the probability distribution $P(S = w_{1:m})$ over S of $[w_1, w_2, \cdots, w_m]$ to represent the likelihood that S is a sentence, calculated as:

$$\begin{aligned} P(S = w_{1:m}) =& P(w_1)P(w_2|w_1)P(w_2|w_1, w_2) \\ & \cdots P(w_i|w_1, w_2, \cdots, w_{i-1}) \cdots P(w_m|w_1, w_2, \cdots, w_{m-1}) \\ =& P(w_1) \prod_{i=2}^{m} P(w_i|w_{1:(i-1)}) \end{aligned} \tag{2.1}$$

However, if too many words accrue in the text, it is hardly to estimate $P(w_i|w_1, w_2, \cdots w_{i-1})$ because of sparsity. Alternatively, we can simplify it with an n-gram model that only considers n previous words in the front of w_i, known as the n-step transition probability of a Markov chain:

$$P(w_i|w_1, w_2 \cdots w_{i-1}) \approx P(w_i|w_{1+i-n}, w_{2+i-n} \cdots, w_{i-1}) \tag{2.2}$$

Then how can we know the value of $P(w_i|w_1, w_2 \cdots w_{i-1})$? Normally, we assume that words in the text is subject to a polynomial distribution θ and use the maximum likelihood estimation and Lagrange multiplier to estimate the value of θ.

When $n = 1$, it is called the unigram model and $P(S = w_{1:m}|\theta) = \prod_{i=1}^{m} P(w_i)$ which indicates that words are independent from each other without any semantic and ordering information. It is called the bigram model and trigram model when $n = 2$ and $n = 3$. Under the n-gram model, the conditional probability normally computed with the frequency counting:

$$P(w_i|w_{1+i-n}, w_{2+i-n} \cdots w_{i-1}) = \frac{count(w_{1+i-n}, w_{2+i-n} \cdots, w_{i-1}, w_i)}{count(w_{1+i-n}, w_{2+i-n} \cdots, w_{i-1})} \tag{2.3}$$

where $count(w_{1+i-n}, w_{2+i-n} \cdots w_{i-1})$ is the times that words sequence $[w_{1+i-n}, w_{2+i-n}, \cdots, w_{i-1}]$ appears in the text. Intuitively, a larger n will keep more ordering and semantic information of the text, however, it also leads to a more severe sparsity issue with Eq. 2.3. Therefore, the trigram model and other methods for smoothing purpose are usually adopted in real application. For example, to attenuate the noise caused by those co-occurrences that happen rarely or never, the Golbal Vector Model, which is based on the "words - words co-occurrence matrix" where the time of words w_i and w_j co-occurring in a corpus is x_ij, is added a weight function $f(x_ij)$ into the cost function with properties: $1) f(0) = 0$; $2) f(x)$ is non-increasing and relatively small to x_ij so that both rare and frequent co-occurrences will not be overweighted [132].

Neural Network Language Model

Neural network language model stared from the following idea:

- The problem of estimating $P(w_i|w_{1:(i-1)})$ within a corpus containing a vocabulary $\mathbf{V}$ of length $|\mathbf{V}|$ can be seem as a problem of multi-class classification and the number of class is m. It can be formulated with follow:

$$P_{k \in |\mathbf{V}|} \left(label(w_i) = k | h_i = label(w_{1:(i-1)}) \right) = f_k(w_{1:(i-1)}, \alpha) \tag{2.4}$$

  where, $label(w_i)$ is the predicted class label of word w_i; $h_i = label(w_{1:(i-1)})$ is the previous information of w_i and $f_k(w_{1:(i-1)}, \alpha)$ is a classification function which es-

timate how probable the w_i is the k-th word in the vocabulary with the constrain $\sum_{k=1}^{\mathbf{V}} f_k(w_{1:(i-1)}, \alpha) = 1$, where α is a parameter.

The process of optimising a classifier can be coped with many methods of machine learning, among which, the neural network model draws much attention. Xu et al. first introduced the neural network model to the bigram language model [**?**]. Bengio et al. formally proposed a neural network language model based on n-gram model [13, 14] becoming the representative work in this research direction. Typically, when we know the first i words $[w_1, w-2, \cdots, w_{i-1}]$, we estimate the i-th word w_i with a neural network language model consisting of the following three layers:

- *Input layer* maps $[w_1, w-2, \cdots, w_{i-1}]$ as word embeddings $[e_1, e-2, \cdots, e_{i-1}]$. In this layer, a word w_i is transformed to a dense real-valued vector e_{w_i} by $e_{w_i} = Mv_i$, where v_i is the one-hot representation of a word w_i and $M \in \mathbb{R}^{t \times |\mathbf{V}|}$ is a t-dimensional word embedding matrix. Each column of M corresponds to a real-valued vector of a word, we denote it as e_{w_i}. This procedure embeds a word into the continuous semantic space and reduces the dimension of it from $|V|$ to t.
- *Hidden layer* employs different types of neural networks, such as the Feed-forward Neural Network (FNN), the Recurrent Neural Network (RNN) and many other manifolds to compute a representation of linear distributional features of previous information h_t. FNN requires a fixed size for input vector; therefore, word embeddings from the input layer will be formed as a long x vector in successive and $h_t = tanh(b_1 + WX)$, where W is the input-to-hidden weights matrix and b_1 is another output biases.
- *Output layer* uses a classifier, $y_t = softmax(Oh_t + b_2)$, maps the values of h_t to a vector $y_t \in \mathbb{R}|\mathbf{V}|$ that represents a probability distribution in which the j-th elements is the posterior probability that the t-th word is the j-th word in $\mathbf{V}$, where $O \in \mathbb{R}^{|v| \times t'}$ is the hidden-to-output weights matrix and b_2 is another output biases. O can also be seen as another word embedding matrix, where each row is a new word embedding.

Neural network language model has been a research hotspot in recent years. Many efforts has been taken with different concerns, for example how to train word embeddings [38, 198], how to improve the basic n-gram model [111] and introducing deep learning

method to neural network language model [85, 117, 159].

The performance of language models depends on the input text and the training model and it will affect the performances of further tasks and applications. On the contrary, methods designed for other tasks are tightly associated with the input text representation.

Next, we focus on the primary techniques of topic detection and document clustering and classification tasks, starting with their common foundation, the topic model.

2.3 Topic Model

Generally speaking, a topic model discovers and represents the latent topics hidden across the input corpus by grouping related words without any supervision. It holds a basic assumption that words indicating similar topics will appear in a document frequently. It is easily understood that topics are represented with a sequence of related words. With the trained topic model, a document can be assigned to each topic with a certain probability and can be categorised to a set of documents with the same probability. Hence, except conducts tasks of topic detection and document classification and cluster, a topic model is also capable of outputting a representation of the text in the topic space, so that the dimensionality of features space can be reduced. Fig. 2.2 illustrates the connection between functions of a topic model.

One thing needs to be especially pointed out is that most topic models, to the best of our knowledge, are based on BoW model with a global view, looking at words distributions and documents distributions across the entire corpus, which also ignore the ordering of words and semantic information among them, although distributed representation of words has been introduced to topic models in recent three years. The work in this thesis is also based on BoW model, but integrating word embeddings with the latent topic model will be our future direction. We will explain it in Chapter 6.

In the following, we introduce previous works for topic model from two aspects, in terms of two categories of methods, the *probabilistic distribution* and the *linear algebra*, for topic detection.

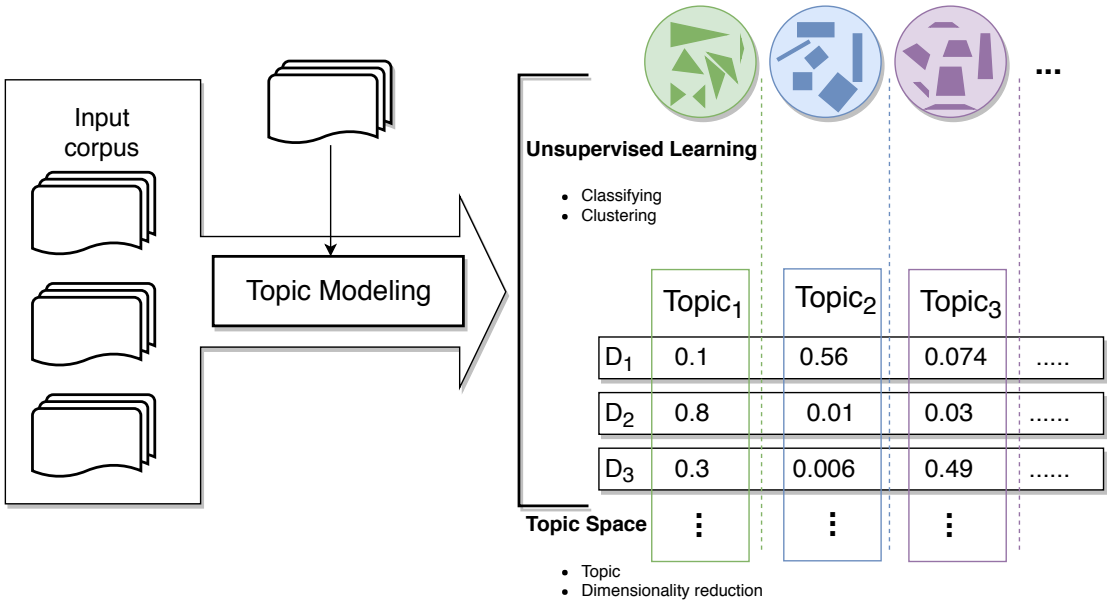


Figure 2.2: The Functions of A Topic Model

2.3.1 Methods for Topic Detecion

Non-probabilistic Methods

To discover the latent topic space, Latent Semantic Indexing (LSI), also referred to as Latent Semantic Analysis (LSA) in some works, explores the basic assumption of the topic model and describes the similarity between documents with a latent space representation. The corpus, a collection of documents, is represented as a words-documents matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ under VSM as we introduced above. Then, how to discover the latent space

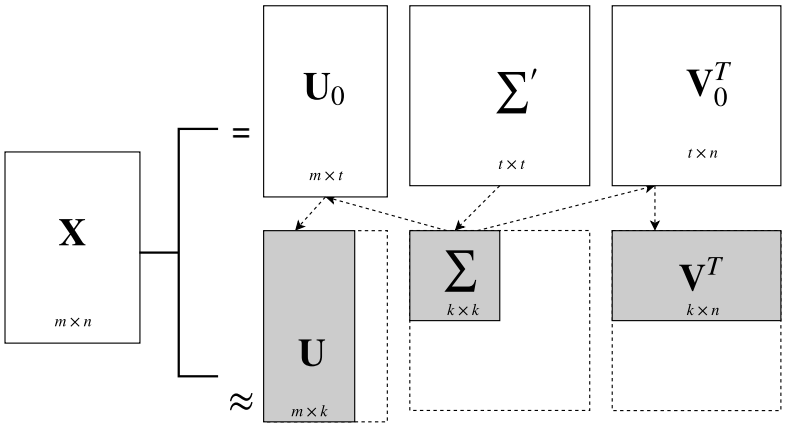


Figure 2.3: A schematic of the SVD based LSI

representation from the original space representation $\mathbf{X}$? The answering is using the dimension reduction techniques on $\mathbf{X}$. The first generation of LSI methods [41, 51, 91] are Singular Value Decomposition (SVD) methods that decompose the original rectangular matrix into the product of three matrices as $\mathbf{X} = \mathbf{U_0}\sum_0 \mathbf{V_0}^T$, where matrices $\mathbf{U_0} \in \mathbb{R}^{m\times t}$, $\mathbf{V_0} \in \mathbb{R}^{t\times t}$ have orthogonality on columns, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$; $\sum_0 \in \mathbb{R}^{t\times n}$ is the diagonal matrix of singular values and t is the rank of $\mathbf{X}$. The theoretical background is from a mathematical proof that any matrix can be decomposed perfectly by using no more factors than the smallest dimension of the original matrix. However, differing from the traditional SVD, LSI emphases a smaller size of $\mathbf{U}$ and adopted a reduced SVD on $\mathbf{X}$ so that an approximation of $\mathbf{X}$ will be obtained as $\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{U}\sum\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{m\times k}$, $\sum \in \mathbb{R}^{k\times k}$ and $\mathbf{V} \in \mathbb{R}^{k\times n}$, and $k \ll (m, n)$ is the pre-defined topic number, which can be implemented simply by deleting coefficients in the diagonal matrix. Fig. 2.3 shows a schematic of the SVD based LSI.

We interpret the decomposition as follows. Each element x_{ij} in $\mathbf{X}$ is the value of i-th feature of document d_j. After the SVD decomposition on $\mathbf{X}$, u_{il} is to what degree that word w_i is belong to this category of words cw_l. v_{jp} is to what degree that the document d_j is relevant to a topic T_p, and the middle matrix $\sum$ indicates the correlation between categories of words and topics, which also indicates the relevant dependence between the column $U_{\cdot p}$ in $\mathbf{U}$ and the column $V_{\cdot p}$ in $\mathbf{V}$. Therefore, with SVD, we can obtain the categories of similar words and the categories of documents simultaneously. Moreover, the topic T_p can be represented by sequentially ordering words in U_l, where $p \leftrightarrow l$. The documents-topics matrix $\mathbf{V}$ can be used for the computation of similarity between documents, for example using the cosine distance, and the further clustering and classification. After some unnecessary noises are filtered out through SVD, the approximated matrix $\hat{\mathbf{X}}$ can also be regarded as a purer words representation of the corpus. How to conduct the SVD is a mature technology so that we do not say too much here.

LSI has attracted much attention once it was presented. It was employed by Dumais et al. to a manuscripts-reviewing system to analyse the research interests and reviewers' interests, and assign relevant papers to reviewers [52]. Bradford combined the technique of entity extraction and LSI presenting an algorithm to generate and display an initial estimate of nodes and links relevant to a chosen topic and applying it to generate Graphs

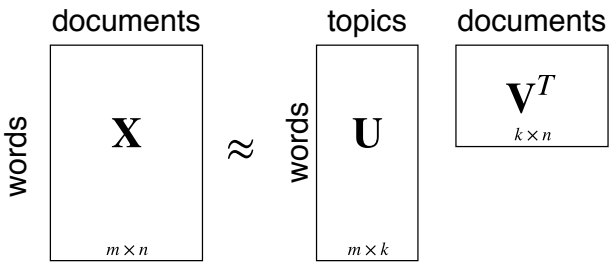


Figure 2.4: A schematic of the NMF

of terrorist networks [22]. LSI can also be used in short text. Yang et al. proposed a hot topic detection approach for Chinese microblogging data, in which LSI was used to map a microblogging text vector to low-dimensional feature vector space [184].

Although LSI exhibited impressive results on a number of textual document, drawbacks of SVD are noticeable. For example, the decomposing high dimensional input matrix is very time-consuming for SVD, and the negative value obtained by SVD is challenging to interpret. Therefore, an improved LSI method, Non-negative Matrix Factorisation (NMF), was proposed afterwards.

NMF decomposes the giving non-negative matrix $\mathbf{X}$ to two non-negative matrix $\mathbf{U} \in \mathbb{R}^{m \times k}$ and $\mathbf{V} \in \mathbb{R}^{n \times k}$ satisfying $\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$. The non-negativity constraints only allow the representation additive $x_{ij} \approx \left(\mathbf{U}\mathbf{V}^T\right)_{ij} = \sum_{a=1}^{k} u_{ia} v_{ja}$, which is in contrast to SVD and PCA (Principal Components Analysi) where negative subtractions are allowed [93]. This property also makes it output a parts-based representation that is easy to interpret [75], and be considered as an unnormalized probability distributions over topics [154]. The problem here is how to find the proper two matrix $\mathbf{U}$ and $\mathbf{V}$. To measure the distance of the approximation $\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$ and conduct optimisation, two distance functions are commonly used:

- *Euclidean distance between* $\mathbf{X}$ *and* $\mathbf{U}\mathbf{V}^T$

$$f_1 = \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 = \sum_{i,j} \left(x_{ij} - \sum_{a=1}^{k} u_{ia} vja \right)_F^2 \tag{2.5}$$

- *Divergence between* $\mathbf{X}$ *and* $\mathbf{U}\mathbf{V}^T$

$$f_2 = \mathcal{D}\left(\mathbf{X} \| \mathbf{U}\mathbf{V}^T\right) = \sum_{i,j} \left(x_{ij} \log \frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij} \right), \text{ where } \mathbf{Y} = \{y_{ij}\} = \mathbf{U}\mathbf{V}^T \tag{2.6}$$

Here, the Frobenius norm is always used, but other norms, for example ℓ_1-norm [82, 89, 119] and $\ell_{2,1}$-norm [76, 87, 191], are used recently. Fig. 2.4 graphically shows the NMF. On the right hand, column $U_{\cdot a}$, $a \in [1, k]$ represents the a-th topics and elements u_{ij} is the correlation between the word w_i and the topic T_a; while v_{ja} is the correlation between the document d_j and the topic T_a. Only two matrices are produced by NMF, so the similarity between words is sacrificed.

Comparing to SVD_based LSI, NMF owns advantages of easy interpretation and less time-consuming. Moreover, NMF has probabilistic property to some extent. Many attentions from research and industry communities have turned to NMF recently, and many improvements for manifold NMF have proposed which lies in not only text mining area, but also the image processing [102, 171] and bioinformatics [161, 170, 185] etc. For topic detection, many efforts with NMF have been done. Vaca et al. utilised the time sequence proposing a time-based factorization algorithm for topic discovery to monitoring news [168]. Choo et al. proposed a semi-supervised NMF based topic model involving user interactions and user's prior knowledge in topic modelling processes [33]. Suh el. al proposed an ensemble framework using residual matrix obtained from multiple times of NMF iteratively to explore deeper semantic similarity and discrimination among words to represent different topics better [157]. Du et al. applied the divide-and-conquer strategy on NMF and proposed a fast algorithm for analysing large-scale data sets [50]. The topic hierarchy started to get more attention because it can help the model discover the potential relations between topics and adapt to the inherent distribution of the corpus, relaxing the condition of the pre-defined number of topics. Chen et al. proposed a fast progressive EM algorithm through hierarchical latent tree analysis [31]. Tu et al. proposed a hierarchy NMF approach to detect and tracking the evolving process of the topic hierarchy in a text stream. However, this method needs an extra cluster step to determine the topic number [166]. Through mining the hidden geometrical structure of the original data space, Cai and Liu et al. proposed graph regularized NMF and constrained NMF for image representation [24, 102]. Besides, NMF has been widely applied on the community detection task [105, 109]. In addition, NMF methods for clustering and online data stream emerged also, which will be reviewed in the rest of this chapter.

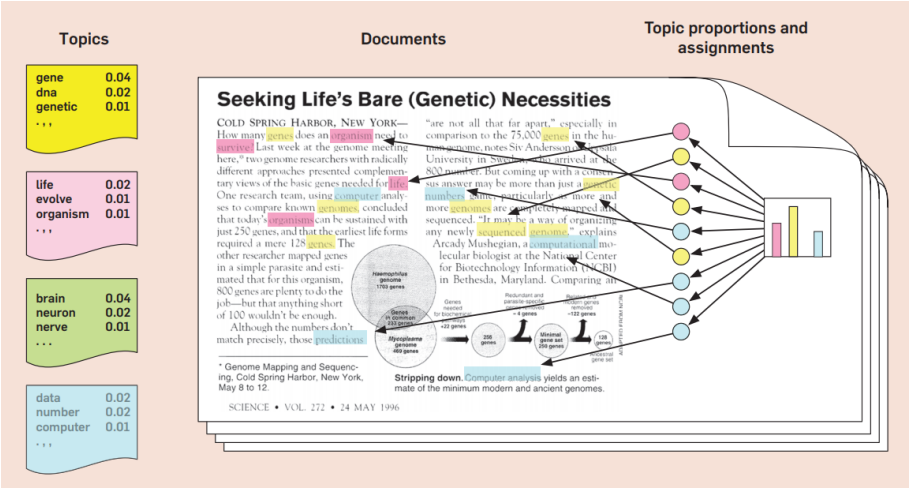


Figure 2.5: An illustration of the assumption of probabilistic topic model [18]

Probabilistic Methods

Different from non-probabilistic LSI, probabilistic topic methods allow a model to extract sets of term probability distributions from the corpus and can be applied to new documents. Since the intuition behind probabilistic methods is that multiple topics may appear in one article [18], they always suppose that there is a jointly probability between the words and the documents which are observed and there are also conditional probabilities across the latent topics that describe the jointly probability with the marginal probability of a document. Fig. 2.5 give an illustration of the assumption of probabilistic topic model [18] where we can find that a document consists of a mixture of topics, and each topic consists of a collection of words. There are two models have to mention here, Probability Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA).

PLSA [72] introduced probabilistic distributions to LSI (LSA); therefore, some researches treat it as an alternative to NMF rather than a real probabilistic method [18]. Ding et al. proved that PLSI and NMF with the divergence-based objective function are equivalent since they optimise the same objective function [44]. But on the other hand, PLSA is generally considered a particular case of LDA. So we review them together in this thesis. Fig. 2.6 graphically demonstrates their idea, where grey nodes are observed variables, and white nodes are hidden variables or parameters. Rectangles denote the replication of the inside elements and the bigger number in the bottom right corner denotes the number of times to replicate, which also corresponds to the n documents within

the corpus and m_j words in the document d_j. To explain the process, we have to clarify that both of them are reversed generation process of documents we observed, rather than finding topics in an existing collection of documents.

The process of PLSA is as follow:

1. Choose a document d_j with the probability $P(d_j)$;
2. Choose a topic T_a to present in d_j with conditional probability $P(T_a|d_j)$;
3. Generate a word w_i in terms of T_a with conditional probability $P(w_i|T_a)$.

What we can observe in the corpus is the word and the documents (d_j, w_i), while T_a is a hidden parameter. According to the manipulation between joint probability and conditional probability, we have:

$$P(d_j, w_i) = P(d_j)P(w_i|d_j) = \sum_{a=1}^{k} P(w_i|T_a)P(T_a|d_j) \tag{2.7}$$

where, $P(d_j)$, $P(T_a|d_j)$, and $P(w_i|T_a)$ are parameters of PLSI model and $P(d_j)$ can be ob-

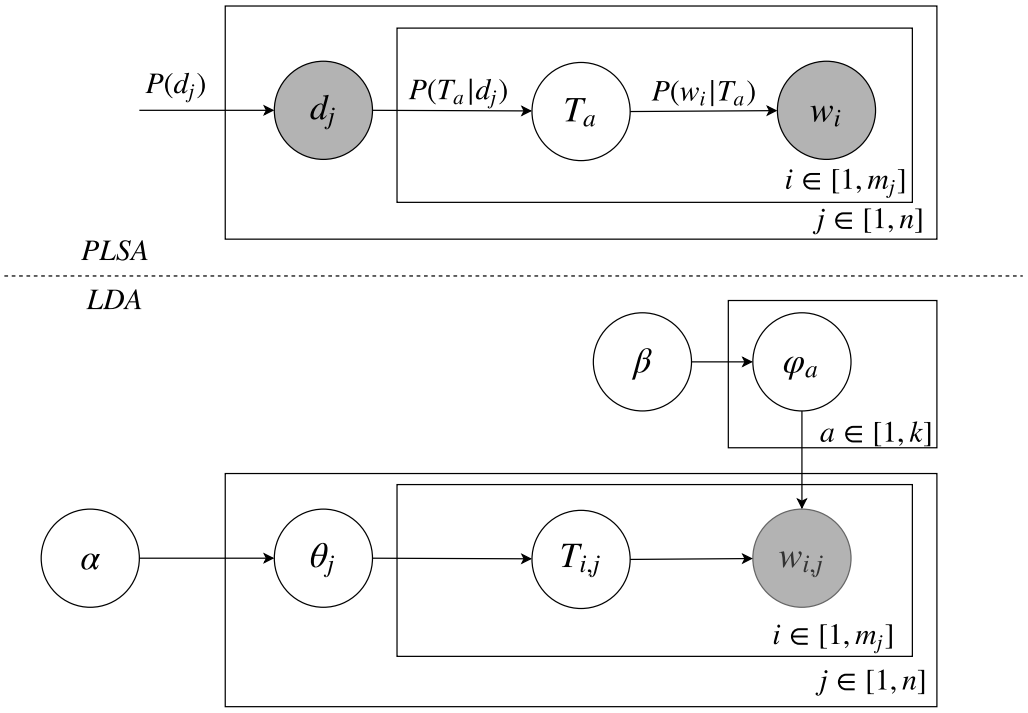


Figure 2.6: The graphical models for PLSA (up) and LDA (down)

served directly from the input corpus. PLSI provides multinomial distributions with parameters $P(T_a|d_j)$ and $P(w_i|T_a)$ over topic space and vocabulary, and trains parameters with the expectation-maximization algorithm (EM). We are not going to the technique details here, but talking about another interpretation of the formula of the jointly probability of PLSI as follow:

$$P(d_j, w_i) = \sum_{a=1}^{k} P(w_i|T_a)P(T_a)P(d_j|T_a) \tag{2.8}$$

Which has a quite similar structure with the LSI model $\mathbf{X} \approx= \mathbf{U}\sum\mathbf{V}^T$ where the probability of the topic $P(T_a)$ corresponds to the diagonal singular matrix of probabilities of correlation between categories of words and topics; conditional probability $P(d_j|T_a)$ corresponds to the documents-topics matrix $\mathbf{V}$; and the conditional probability $P(w_i|T_a)$ corresponds to the words-categories of words (i.e. topics to some extent) matrix $\mathbf{U}$. The full deduction can be referred in [44]. Hence, like LSI, PLSA can only discover topics for documents that in the corpus, but do noting for a new documents because it does not involve a parameter to model $p(d_i)$. And, the overfitting issue raises naturally because the number of parameters completely equals to the number of documents in the whole corpus.

Blei et al. proposed LDA as a more far more flexible model for topic detection by adding Dirichlet priors for the document-topic and word-topic distributions. Dirichlet distribution is a distribution over distributions that each variable drew from the Dirichlet distribution is a parameter of the multinomial distribution. The premise here is that LDA assumes that topics are specified before any document has been generated [18]. In Fig. 2.6, the generative process of LDA is:

1. For a document d_j, randomly choose a distribution θ_j over topics from a dirichlet distribution $Dir(\alpha)$: $\theta_j \sim P_{Dirichlet}(\theta_j|\alpha)$;
2. For a word in this document $w_{i,j}$, select a topic $T_{i,j}$ from the distribution θ_j: $T_{i,j} \sim P_{Multinomial}(T_{i,j}|\theta_j)$;
3. Randomly select a word distribution φ_a over vocabulary for the topic $T_{i,j}$ from another dirichlet distribution $Dir(\beta)$: $\varphi_a \sim P_{Dirichlet}(\varphi_a|\beta)$;
4. Randomly choose a word for this document $w_{i,j}$ from the distribution φ_a of the topic

$T_{i,j}$: $w_{i,j} \sim P_{Multinomial}(w_{i,j}|\varphi_a, T_{i,j})$.

Where α is a super-parameter of θ and β is a super-parameter of φ. Since we used m to denote the size of the vocabulary of the whole corpus already, m_j denotes the number of words in d_j and a word in document d_j is denoted as $w_{\cdot,j}$ here. This process perfectly reflects the assumption in Fig. 2.5 that each document exhibits topics in different distribution and each word in each document is drawn from one of these topics. Beside, LDA chooses a word distribution φ_a for each topic $T_a, a \in [1, k]$. This process can be formulated with the following jointly probabilities of the hidden and observed variables with parameters α and β:

$$P(\varphi_{1:k}, \theta_{1:n}, T_{1:n}, w_{1:n}|\alpha, \beta) = \prod_{a=1}^{k} P(\varphi_a) \prod_{j=1}^{n} P(\theta_j) \left(\prod_{i=1}^{m_j} P(T_{i,j}|\theta_j) P(w_{i,j}|\varphi_{1:k}, T_{i,j}) \right) \tag{2.9}$$

And, algorithms based on Gibbs Sampling or Variational Bayesian can be used to optimise above function.

To date, comparing to PLSA, LDA can be generalized to new document since the input collection of documents are the training set for the Dirichlet distribution of documents-topics and words-topics distributions. The words-document representation of a new document can be projected to the low-dimensional topic space.

With many efforts to improve and modify PLSA and LDA, probabilistic topic models have become a core tool for the analysis of text mining. Steyvers et al. developed idea of LDA and proposed probabilistic author-topic models that generate documents with a two-stage stochastic process: authors-topics and topics-words, where the words in a multi-author paper are assumed to be the result of a distribution of each authors topic distribution [155]. Nallapati et al. proposed Pairwise-Link-LDA and the Link-PLSA-LDA models in [118]. The Pairwise-Link-LDA model combines the LDA and Mixed membership stochastic block models [2] (another work in bioinformatics of LDA's author) for jointly modelling text and citations in the topic model framework focusing only on the single domain of blog data. However, they found that Pairwise-Link-LDA model is computationally expensive because of the operation conducted for every pair of documents. They proposed another model, Link-PLSA-LDA model, combines the LDA and PLSA models into a single graphical model. Tang et al. presented three topic models

for simultaneously modeling papers, authors, and publication venues which and further integrated them into the random walk framework for academic search [160]. Ramage et al. proposed a topic model for the credit assignment problem on social bookmarking websites that constrains Latent Dirichlet Allocation by defining a one-to-one correspondence between LDAs latent topics and user tags, named labelled LDA [136]. Andrzejewski and Zhu added partial supervised information, called topic-in-set knowledge, to the latent topic model which can encourage LDA to recover topics relevant to user interests [7]. Foulds et al. inherited the idea of LDA but replaced the Dirichlet priors with a tractable class of conditional random field (CRF) models over continuous random variables consists of a flexible probabilistic programming framework for designing custom topic models [59]. To the universal assumption of the topic model that words are generated independently, Xie et al. proposed a Markov Random Field (MRF) regularised Latent Dirichlet Allocation (LDA) model to take advantage of the rich similarity relationships among words by encouraging words labelled as similar to share the same topic label. As a result, the topic assignment of each word by their model is affected by the topic labels of its correlated words [180]. Chou et al. proposed a method to estimate context-aware sentiment value for concepts on Chinese ConceptNet using LDA to generate a topic for each context [34]. LDA model is also used in image data recently, for example, a nonlinear compressed sensing-based LDA Topic (NCSLT) model was proposed to classify polarimetric synthetic aperture radar (PolSAR) images [66]. Recently, Pham et al. introduced LDA to generate the weighting attribute for the objects links to better find the most relevant information within large-scale heterogeneous information networks [134]. Pavlinek et al. proposed text classification method for very small labelled datasets which uses LDA topic model to extract features then passed forward to a self-training algorithm [128]. Hong et al. applied LDA to an internet topic evaluation model to explore the intrinsic links among the news and extracted effective topics [74]. LDA was also applied to the topic hierarchy. Paisley et al. proposed a nested hierarchical Dirichlet process for a hierarchical topic model which allows each observed word to follow its own path to a topic node in the tree [121]. On the other hand, probabilistic methods also have drawbacks, for example, Choo et al.'s experiments indicated that the convergence of LDA is random, which means it basically gives the user no control over the algorithm process. Besides,

there are also concerns about the less consistency from multiple runs and the complexity in the formulation and the implementation of LDA comparing to NMF [33].

2.3.2 Document Clustering and Classification

Finding proper groups for documents in a corpus is a widely studied problem with many applications, for examle, the information retrieval, document summarisation, document organization and document classification. It is a product of jointing clustering and classification of machine learning and topic model and text representation of text mining.

Specific to text mining domain, one commonly adopted direction is based on matrix factorisation, including SVD [120], NMF [183] and concept factorisation [23]. They can be divided into two categories according to the method. One of them utilise the topic model to reduce the text representation from the words space to the topic space, then conducts other clustering and classification algorithm on the new topic space-based topic representation. This paradigm is also called the feature selection on text since the dimensionality reduction can effectively lower the noise of similarity measure and magnify the semantic associations in the underlying data space [1]. [24] proposed an NMF based data representation algorithm to encode the geometrical information by employing the idea of the spectral clustering. [23] proposed a locally consistent concept factorisation by using graph Laplacian to smooth the document-to-concept mapping so that concepts with respect to the intrinsic manifold structure can be extracted as soft labels and documents associated with the same concept can be well clustered. Both of the two methods adopted a k-nn clustering after the dimensionality reduction. So did [107, 171].

Another paradigm directly corresponds to the result of the reduced text representation on topic space of the topic model. For example, documents are clustered by examining topic distribution vector θ with LDA and intuitively a document d_j is assigned to cluster a if $a = argmax_j \theta_j$ [106]. Also In NMF, the output documents-topics matrix $\mathbf{V}$ can be regarded as a cluster indicator and each element $v_a j$ represents a value that similar to the probability indicating that document d_j belongs a-th cluster, $a = argmax_j v_{ij}$ [174, 183]. [183] simply added a requirement that the Euclidean length of each column vector in matrix $\mathbf{U}$ is 1 to make the factorisation solution $\mathbf{V}$ and $\mathbf{U}$ unique. Many works [46, 87, 192] improved the NMF for more rubust performance of document clustering so far. LDA-

basd document clustering algorithms also move forwards. Papanikolaou et al. proposed a Subset Labeled LDA to tackle the scalability issues for labelled LDA [136], making it appropriate for extreme multi-label classification tasks [124].

Making use of the topic hierarchical structure is also a research centre that distinguishes document clustering and classification from flat clustering. Kuang and Park proposed a fast rank -2 NMF method to hierarchical cluster document to a binary tree [90]. The brilliance of this fast rank -2 NMF method from NMF's perspective is that it applied an two block coordinate descent framework to solve the optimisation in NMF as a non-negative least squares, that can be highly paralleled in implementation and performed extremely fast. Yang et al. proposed a hierarchical attention network mechanism for document classification, in which a document vector is progressively built by aggregating important words into sentence vectors and then aggregating important sentences vectors to document vectors [186].

Recently, distributed representation model was introduced to Document Clustering and Classification task. Yoon Kim trained the convolutional neural networks (CNN) on pre-trained word vectors for sentence-level classification tasks [84]. Zhang et al. applied CNN on characters and found the conclusion that when trained on large-scale textual datasets, deep CNN does not require the knowledge of words [193], which is a further finding of previous researches that CNN does not require the knowledge about the syntactic or semantic structure of a language [64, 84].

2.4 Influence of social network service and stream processing

The advent of social network service brings many problems to topic detection and document clustering and classification, such as how to catch the ever-changing concepts in text stream and how to properly represent the short and sparse text. In the following, we introduce some typical techniques and algorithms with respect to the topic detection and clustering and classification.

Online topic models concerns about catching the temporal information and the lifecycle of a topic, i.e. the emerging, evolving and fading process. We have mentioned that various social network services sped up and complicated this process consider-

ably. Therefore, developing the online topic detection that aims at the continuously data spreading among online social network is very necessary.

An online NMF approach was proposed to detect latent factors and track their evolution while the data evolve [25]. Online NMF (ONMF) can automatically and incrementally update the latent factors and thus track the changes. Chou et al. proposed an incremental probabilistic latent semantic indexing (IPLSI) method for event detection in a continuous stream of multi-sources documents [35]. Above mentioned hierarchical NMF method [166] extended ONMF to generate topics in a hierarchical structure and dynamically adjust the hierarchy with the evolving process. IPLSI alleviates the threshold-dependency problem to some extent and simultaneously maintains the storyline of the latent semantics for developing events. AlSumait et al. proposed an online-LDA model to capture the thematic patterns and identify emerging topics of text streams and their changes over time automatically and continuously [5]. This method directly modified the LDA in an online fashion to incrementally build an up-to-date model. Kasiviswanathan et al. proposed an Online ℓ_1-dictionary learning algorithm to detect novel documents, or saying carrier of novel topics, from a voluminous stream of textual documents and an ℓ_1-penalty was adopted to compute the reconstruction error for the dictionary instead of the squared loss [82]. Except had been applied on documents stream, this method was texted on tweets stream as well. Twitter-LDA is a model designed for short tweets to discover topics from a Twitter sample representation [194]. Sasaki et al. extended Twitter-LDA by adding an online inference considering a sequence of tweets rather than assuming tweets are independent and exchangeable [141]. Saha et al. proposed an online NMF framework to rapidly analyse the information content in online social media streams with a temporal regularisation that captures the emerging trends [139]. Spina et al. learnt tweets similarity function with all types of Twitter signals and applied a clustering algorithm on the previously learned similarity function for reputation monitoring on Twitter [152]. For bursty topic detection, Xie et al. proposed a sketch-based topic model integrated with dimension reduction techniques based on hashing to achieve scalability on real-time detection [181]. Mirończuk et al. published a detailed survey about the state-of-the-art elements of text classification [114]. Classification and clustering techniques for data stream are widely used in text data [4, 6, 26, 127, 167].

Chapter 3

A Multi-window Based Ensemble Learning for Imbalanced Data stream

Imbalanced data stream is pervasively distributed in various real applications, including the Internet news, the email conversations, the capital flows and the trade transactions. Although it has attracted much attention from data mining and machine learning research community in recent years, most studies still focused on either imbalanced data or streaming data. However, the more serious problem is that both imbalanced data and streaming data are always appear together in practice which brings big challenge of catching and detecting the imbalanced distribution in time-critical continuous data stream. In this chapter, we propose a multi-window based ensemble learning framework for the classification of imbalanced streaming data. Four types of windows are defined to store the current batch of instances, the latest rare instances, sub-classifiers of the ensemble classifier and instances employed to train existing sub-classifiers. The ensemble classifier consists of a set of up-to-date sub-classifiers, which are combined with a majority weight voting scheme to predict the labels for newly arriving instances. Training the new sub-classifier is only necessary when the concept drift is detected. Extensive experiments on real datasets covering five different application scenarios and synthetic datasets generated following three distributions demonstrate that the proposed multi-windows method can efficiently and effectively classify imbalanced streaming data with outstanding performances across comprehensive evaluation criteria compared to baseline approaches.

3.1 Introduction

CLASSIFICATION is one of the most important problems in the fields of data mining and machine learning, and has attracted much attention in recent years. In conventional methods of solving the classification problem, a classifier is generally trained on a dataset that does not change over time. As such, the dataset is assumed to have all the information required to learn the underlying concepts. However, in many real-world

scenarios, including spam filtering [42], credit card fraud identification [172], intrusion detection [126], and web-page classification [175], datasets are not static. Typically, the dynamic data forms a continuous data stream following an emerging in-memory computing paradigm where a process-once-arrival strategy is applied and the process are time-critical. When dealing this kind of data, we assume that new instances may arrive one by one or batch by batch, and incoming instances must be classified within a finite period with finite resources. Regardless of whether new instances arrive incrementally or in batch, only data received prior to the time step t can be used to train the classifier and predict the instances arriving at the time step t+1. In other words, the classifier can only be trained on an incomplete portion of the information.

The problem of class imbalance is very common in streaming data. For instance, in spam filtering, the amount of spam is usually much less than the amount of normal mails; the fraud is usually the minority comparing with normal customers in credit card fraud identification, and intrusions are not common compared with normal actions. Thus, class imbalance is a vital issue of data streams that cannot be ignored when dealing with real-world problems. Because we are more interested in rare class instances, these are usually denoted as positive instances and instances of the majority are denoted as negative instances.

In an earlier stage of development, classification of streaming data and imbalance problems were studied separately; however, increasing attention has been dedicated to addressing these two problems together in recent year. Most researches combined models designed for streaming and imbalanced data in a relatively simple manner [70], which are almost equivalent to address these two problems separately, and few of them took an insight look into these two problems together. In this chapter, we analyse the twin problems in a novel way and propose a method using multi-window ensemble learning for classification of imbalanced streaming data. Main contributions of this chapter are as follows:

- A multi-window framework is proposed to record the current batch of instances, selected positive instances, and the ensemble classifier along with the corresponding instances used to train each sub-classifier. The framework enables us to accumulate the latest positive instances, and enhance the usefulness of positive instances ac-

cordingly. Furthermore, concept drift in data streams can be detected by the error rate together with similarities between the current window and the history windows used to train each sub-classifier.

- A novel ensemble learning mechanism is designed to classify the incoming instances. The weight of each sub-classifier is kept updating by the latest classification error rate of new instances and the window similarity. New instances are classified using weighted majority voting. Adjusting the weight according to the error rate can improve the classification accuracy, and weight adjustment based on window similarity can solve the reoccurring concept drift issue [70] to a certain extent.

- Extensive experiments on both synthetic datasets and real datasets in different application domains are conducted, from which optimal parameters for each dataset are obtained and analysed. The proposed approach is demonstrated to generally outperform baseline methods.

To simplify the model, we focus only on single-label two-class classification problems, where each instance can belong only to a single class and only two classes are considered. Single-label two-class classification tasks are found in many real-world applications, for example, email messages can be classified into spam and normal messages; credit card users can be labeled as fraudulent and regular users. Moreover, multi-label classification problems can be divided into several single-label two-class classification problems [145]. Multi-class problems can also be subjected to divide and conquer strategies [78, 104, 138]. In addition, we mainly focus on low dimension problems because high dimension problems can always be reduced, as discussed in [148].

The remainder of this chapter is organized as follows. Works related to streaming data classification and imbalanced classification are presented in Subsection 3.2. Subsection 3.3 proposes the multi-window based ensemble learning approach including the multi-window framework, the updating policies and algorithms. The experimental evaluation and discussion on both real and synthetic datasets are demonstrated in Subsection 3.4, followed by a summary in Subsection 3.5.

3.2 Related Work

Classification of streaming data has attracted much attention for some time, and has been widely studied, particularly with problems related to concept drift. Meanwhile, numerous researchers have focused on the imbalanced data classification problem. In recent years, an increasing number of scholars have addressed problems involving either class imbalance or concept drift. We briefly review these works in the following.

3.2.1 Classification of streaming data

Features among newly incoming data may vary over time, which causes sudden changes in the latent generation function of existing data. When a classifier suffers such a non-stationary contexts, concept drift may happens to this streaming dataset at anytime. We can roughly divide previous methods for concept drift on data stream classification into three groups which is adaptive based learning, training set modified learning and ensemble learning.

In adaptive based learning, trainers are improved to self-adaptive the concept drift in streaming data. Decision Tree provides us an intuitive perspective to analyse the entire dataset. Hulten et al. proposed the Concept-adaptive CVFD (Very Fast Decision tree) method [78] based on VFDT(Very Fast Decision Tree) [49] to keep classifier up-to-date by add the ability to detect and respond the underlying concept change in data arriving process with a sliding window by computing new split attributes and comparing them with the old. Thereafter, many works based on CVFD turned up, for example, Bifet and Gavalda proposed HWT(Hoeffding Window Tree), which advocates recomputing split attributes once new item arrives, instead of waiting to the count of items equals to the size of the window, by which the former can conquer the slowly adaptive of the later [17]. At the same time, the authors also developed the HAT (Hoeffding Adaptive Tree) to capture concept drift quickly by the adaptive window technique. Moreover, the kNN algorithm was improved by Alippi and Roveri [**?**], who manage the situations with and without concept drift separately.

While in the training set modified learning, the data or the relative weights are updated to fit the target concept instead of retraining the exist trainer, which gained a

wider application. Typical methods consist of windowing techniques and weighting techniques. Windowing techniques train the classifier by the size of the latest incoming data window. The easiest way for windowing techniques is fix the window size which is difficult to decide with different data category in different discipline. Thus, a self-adaption method FLORA3 was presented to fit the present concept [179]. Also, Game et al. considered the dramatic increasing of error rate as the indication of concept drift [3]. When this happened, the latest arriving items will be used to renew the classifier. And almost at that time, Bifet and Gevalda proposed two approaches to compute the size of the window, of which ADWIN (adaptive window) calculates the similarity between data items within two windows that are big enough, but has a certain drawback of large computation quantity; therefore the authors give another method named ADWIN2 with the designing of a time and memory efficient data structure [16]. Whereas, weighting techniques balance the concept drift by weight data items belongs to different classes. In Alippi and Roveri's work, all data items are weighted by their similarity compared with current underlying concept [3]. This way, the items more similar to the concept will get more weight and others allocated with a relatively low weight will help the concept drift absorbed by the classifier during the data arriving progress.

Ensemble learning integrates the results from multiple sub-classifier, aims to better the classification performance of single classifiers. SEA (Streaming ensemble algorithm) trains the classifier in each non-overlapping window instances [156]. Newly trained classifiers are added to the classifier-list before reach its upper limit; otherwise, the worst classifier to the current window is replaced by the new one. Wang et al. proposed a method weighting each existing sub-classifiers to classify the incoming window of instances [172]. The weights are determined by the results of last window. By contrast, D-WM trains and adds a new sub-classifier to the classifiers list as well as assigns the weight 1 to it when an insistence is misclassified by the existing ensemble classifier and the weights of misclassified sub-classifiers will be reduced [86]. Thus, when the weight less than a certain threshold after certain iterations, the sub-classifier will be eliminated from the list. The third weighting method is Learn++ which always trains new sub-classifier on the test set of newly arrival window of instances and adding it to the ensemble classifier, assigning the corresponding error rate to the new classifier as its weight[135]. Based on

it, Elwell and Polikar proposed Learn++.NSE to deal with the non-stationary problem, which classify the newly arrival instances with all existing sub-classifiers and adjusts their weights according to their error rates, in spite of new or old [53].

3.2.2 Classification of imbalanced datasets

As discussed above, when dealing with imbalanced datasets, normally we are more concerned with the positive class that fewer instances belong to, named as the minority class here. Therefore, the minority class must be emphasized. Previous works can be divided into three categories depending upon the level on which the method targets [29, 67]. The first type eyes on the data-level, which seeks to balance the sizes of classes by data sampling techniques, which are either increasing or decreasing the number of minority or majority instances. For example, the classic SMOTE method, which creates synthetic minority instances rather than duplicating instances, avoiding over-fitting problem of minority instances [67]. In this case, the minority instances are simply oversampled. Random under-sampling method under-sampled the majority instances by randomly deleting some of them to fit the pre-setting imbalance ratio. And, heuristic under-sampling methods focus on discovering instances which make no difference to the prediction result with heuristic rules, then delete some of them according to the imbalance ratio. For example, Tomek-line is used to locate redundant majority instances [163]. The second type works on the algorithm level attempting to improve the performance of a specific model. For example, Liu et al. proposed the CCPDT (Class Confidence Proportion Decision Tree) [103]. Moreover, ensemble learning methods are considered to be a hopeful method to handle imbalanced issue, with which Chawla et. al. proposed SMOTEBoost algorithm [30] and Sun et al. proposed series methods of AdaC1, AdaC2 and AdaC3 [158]. Overall, the issue of imbalance has been highly considered in recent year, however, there are still more open problems [28] waiting for in-depth study and solution. For example, how can we correctly identify the data distribution and how can we evaluate the identified data distribution?

3.2.3 Classification of Imbalanced streaming data

A more practical model is data with imbalanced feature in streams arrive over time which has attracted more attentions from the research community in recent year. The boundary definition (BD) approach was proposed to build classifiers based on boundary instances, which are easily misclassified or hardly differentiated[100]. The approach divides the majority instances into a correctly classified set and a misclassified set. Random under-sampling is performed on them simultaneously to guarantee distribution consistency after under-sampling. Eleftherios et al. proposed a method maintaining two windows to record the positive and negative instances separately for multi-label stream classification problem [182]. To save the space, only indexes of data instances are recorded in the windows. Later on, a learning framework for the online imbalanced classification problem, including an imbalanced class detector, a concept shift detector, and an online learner, was proposed [173]. The researchers also proposed the so-called oversampling-based online bagging (OOB) and under-sampling-based online bagging (UOB) methods to improve predictive accuracy. In the chapter, concept drift was simplified to a change of the imbalance ratio among classes. Thereafter, the authors proposed the sampling-based online bagging (SOB) method, which aims at maximizing the G-mean value to balance different classes by adjusting the parameter λ of the poisson distribution in online bagging [173]. Recently, a selective re-training approach based on clustering has been proposed [190]. In this chapter, a new sub-classifier is trained on newly arrival data when there is not any concept drift. It compares AUC measurement between the new classifier and the old to determine whether or not updating the ensemble classifier with the newly trained sub-classifier. When there is a sign of concept drift, those wrongly classified instances will be blended into previous training sets of existing sub-classifiers respectively with a certain probability to re-train new sub-classifiers, forming new ensemble classifier. This method is computationally complex and time consuming since it trains sub-classifiers for all data instances and makes comparisons of performances between the new and the old classifiers.

3.3 Framework Design

In this section, we first give a formal definition of the problems, followed by the proposed framework, including the design of multi-window mechanism, strategy of updating classifiers and weights of each sub-classifiers, and the entire algorithm description. The notations frequently used in this chapter are summarized in Tab. 3.1.

3.3.1 Problem Definition

Assume that, at time step t, the existing instances form a data sequence $D = \{(X_0, l_0), (X_1, l_1), ..., (X_t, l_n)\}$ in chronological order, where X_i is a d-dimensional vector and corresponds to a class label l_i . All class labels constitute a label sequence $L = \{l_i, L_2, \cdots, l_n\}$. In a two-class classification task, $l_i \in \{+, -\}$, where '+' and '-' represent positive (minority) and negative (majority) classes, respectively. For simple classification problem, training a single classifier might be sufficient; however, the goal here is to train a set of sub-classifiers on existing instances constituting an ensemble classifier under a strategy that we will describe later so that the classifier can as accurately as possible to predicate the label of the incoming instance X_{t+1} at time step $t = 1$; and, once X_{t+1} is predicated, we are aware of the true label of X_{t+1}, denoted by l_{t+1}. If the data is unstable, for example the data distribution changes over time, $p(l_1, l_2, \cdots, l_n|D_{t+1}) \neq p(l_1, l_2, \cdots, l_n|D_t)$, we say the concept drift occurs in the data stream. In addition, we say the dataset is imbalanced if the ratio of the instances' quantities in different classes exceeds a predefined threshold θ, for example $\#\{(X_i, l_i)|l_i =' +'\}/\#\{(X_j, l_j)|l_j =' -'\} > \theta, i, j = 0, 1, \cdots, n$. Therefore,

Table 3.1: Summary of notations

Symbols	Description
X_i	An data instance or an data sample
l_i	Class lable of instance X_i
WB	Window of batch containing certain number of instances
WM	Window of minority containing certain number of positive instances
WC	Window of classifier containing certain number of sub-classifiers
WI	Window of instances that used to train the corresponding sub-classifiers
M_b	The size of WB
max_{WM}	The size of WM
max_{WC}	The size of WC

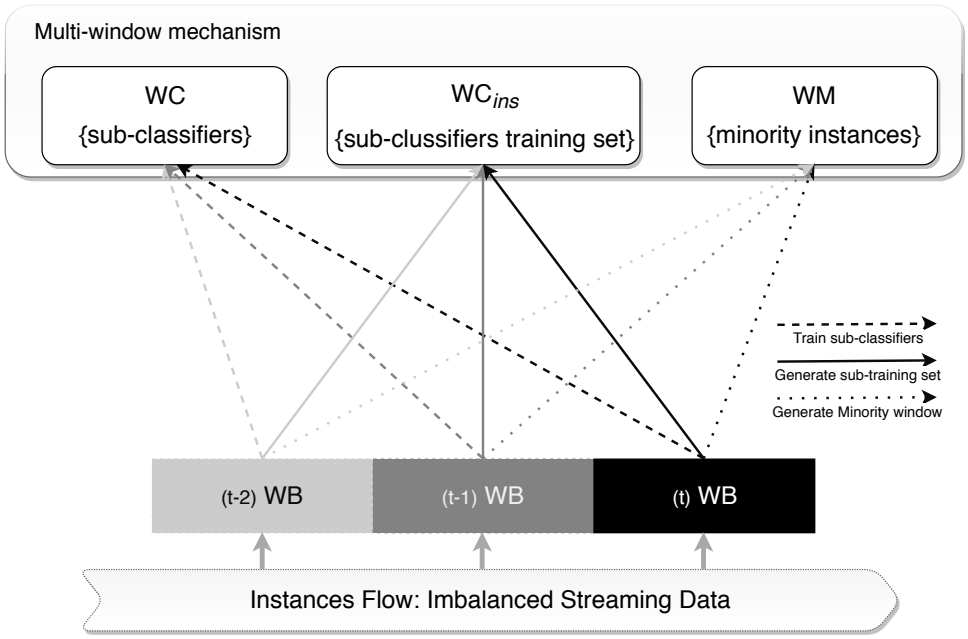


Figure 3.1: Multi-window mechanism.

the task in this section is to build an ensemble classifier that keep self-updating on an imbalanced data stream with possible concept drift issue and obtain acceptable results.

In our study, we adopt a window approach or batching approach, where the optimal window size is pre-calculated and fixed by experiments. We denote the sliding window size as M_b. The sliding window keeps moving forward and the windows on the data stream are non-overlapping and in chronological order. Suppose that, n instances arrive as a sequence at time step t and are divided into n/M_b windows or bathes according to their sequences. Then, each window containing M_b instances will be the input of our method. When $M_b = 1$, the batching approach is equivalent to incremental learning, where a single instance is processed at a time.

3.3.2 Multi-window Mechanism

The multi-window mechanism is shown in Fig. 3.1 where four types of windows, WB, WC, WI and WM, are kept. Before we start to describe our multi-window ensemble learning algorithm in detail, we must determine the size of the sliding window (WB). An overly small window cannot adequately represent the class characteristics, and may result in a classifier with poor generalization. However, an overly large window size

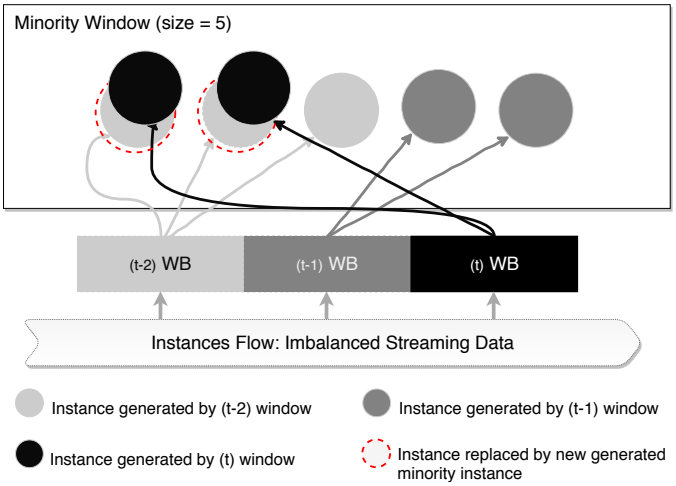


Figure 3.2: An example of updating minority window.

will require too much acquisition time and computing resources, which are restricted in practical applications. For instance, a window may not have acquired sufficient instances within the limited time available after which a prediction is expected. To our knowledge, it is lack of theoretical guidance for determining an optimal window size. Meanwhile, the data distribution also restricts a universal window size. Therefore, for datasets from different scenarios, the most reasonable choice is training an optimal window size through experiments and we will discuss it in Section 3.4.

Algorithm 3.1: Update the WM

Input: WM, WB and size limit max_{WM} of WM
Output: Updated minority window WM'

for *each instance* $wb \in WB$ **do**
  if $classLabel(wb) = '+'$ **then**
    if $|WM| < max_{WM}$ **then**
      $WM_{|WM|} \leftarrow wb$
    else
      for $i = 0, 1, \cdots, max_{WM} - 2$ **do**
        $WM_i \leftarrow WM_{i+1}$
      end
      $WM_{max_{WM}-1} \leftarrow wb$
    end
  end
end

In circumstance of imbalanced data, positive instances are always sparse, or even be

absent from some of the sliding windows. As a result, classifiers trained on these windows may be unable to represent the positive class. Therefore, we use a minority window (WM) to record the newly arrived minority or positive instances. Minority window has also been used in the BD approach [100], although the WM employed in BD consumed excessive time and resources; moreover, early acquired minority instances ran the risk of becoming outdated due to the concept drift happened along the stream. For this issue, an updating strategy on minority window has been proposed in REA approach [32]. In detail, REA fix the minority window size and add minority instances which are similar to the current positive instances into the WM. However, it is time consuming to select the closest instances when the window size is large. Moreover, the closest instances may not be of the same concept. Consequently, we alternatively utilize a simple substitution strategy herein with fixed size, adding minority instances into WM before reaching its upper size limit; otherwise, the oldest instance will be replaced by the newest. Thus, the WM always represents up-to-date positive instances over time. We show an example of updating minority window in Fig. 3.7 and the algorithm is summarized in Algorithm 3.1. Our experiment results show that it is reasonable to set the size of WM in the range of $[0.5, 1]$ times that of the size of WB.

In addition, we use a classifier window (WC) to record the requisite number of newly trained sub-classifiers and their corresponding weights WC_{weight}, as well as the windows of instances WI used to train each sub-classifier. The size of WC is also determined in advance through experiments and fixed. Newly trained sub-classifiers will be added to WC before reaching the predefined upper bound; otherwise, the oldest sub-classifier will be replaced. Moreover, the corresponding weights will also be updated. We summary how to update WC in Algorithm 3.2, where $errorRate = 1 - accuracy$ and give an example in Fig. 3.3. We will further explain when and how to update WC in the next subsection.

3.3.3 WC updating strategy

Notice that, though it is very important for ensemble classifier to increase the inside diversity to enhance its robustness and performance. However, training a new sub-classifier on instances with little concept drift only increases computational cost rather than the diversity. Therefore, our approach does not update or add a new classifier to

WC in each time step. In other words, unless we find that the existing classifier can not works well on the instances of current WB, we train new classifier for this WB. Then the problem comes to how to measure whether the classification result is good or not. The most common and simple metric is accuracy; however, accuracy is not always reliable for imbalanced data. For example, given 99 normal email messages and 1 spam in the dataset, the classifier can treat all 100 messages as normal to obtain an accuracy as high as 99%. Therefore, it is necessary to evaluate the classification result with respect to each class, including class of minority, simultaneously under imbalanced circumstance. In this study, we use the precision of both the minority and majority classes to evaluate the classification performance. We can see the difference between accuracy and precision with

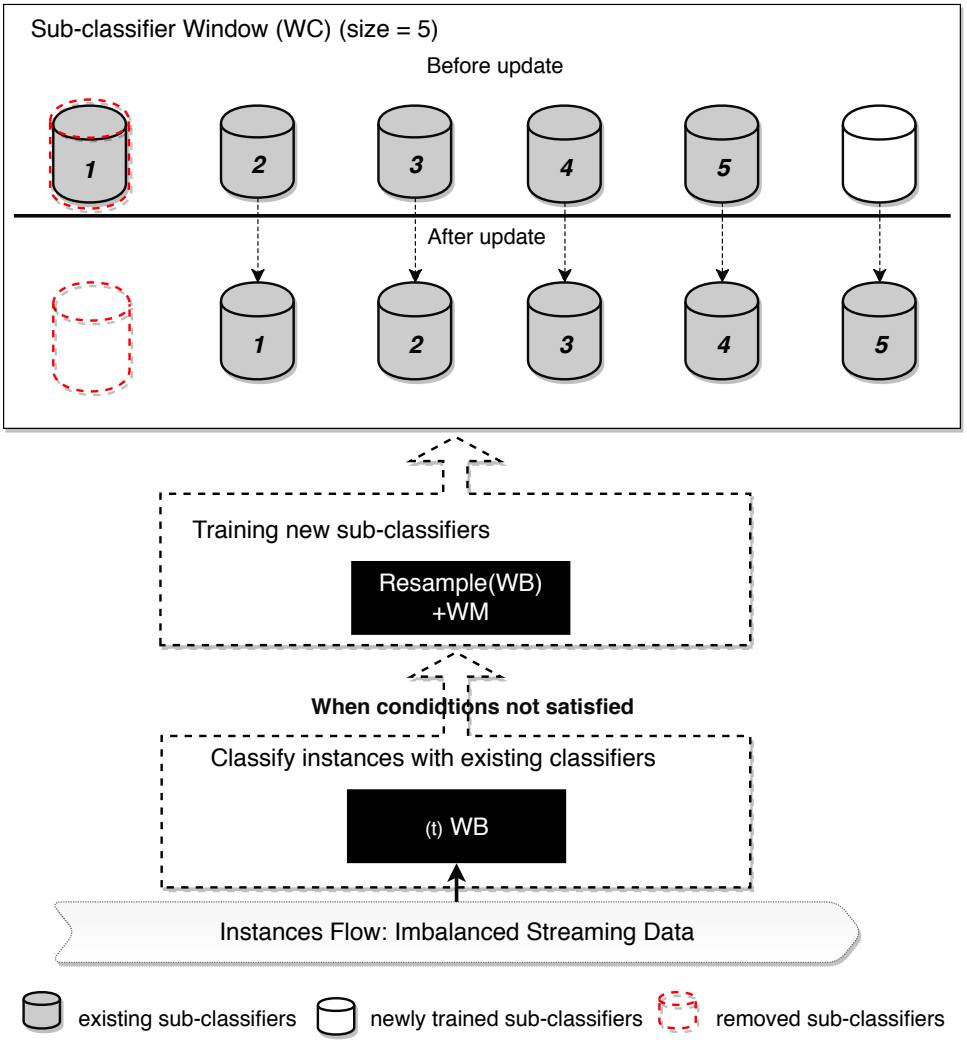


Figure 3.3: An example of updating classifier window.

Algorithm 3.2: Update the WC

Input: WC, WB, WI, new classifier c', list of weights $Weight$ corresponding to sub-classifiers, $errorRate$ and size limit max_{WC} of WC
Output: Updated classifier window WC', updated list of weights $Weight'$ and updated window of instances WI'

if $|WC| < max_{WC}$ **then**
  $WC_{|WC|} \leftarrow c'$
  $WC_{weight_{|WC|}} \leftarrow 1 - errorRate$
  $WI_{|WC|} \leftarrow WB$
else
  for $i = 0, 1, \cdots, max_{WC} - 2$ **do**
    $WC_i \leftarrow WC_{i+1}$
    $WC_{weight_i} \leftarrow WC_{weight_{i+1}}$
    $WI_i = WC_{ins_{i+1}}$
  end
  $WC_{(max_{WC})-1} \leftarrow c'$
  $Weight_{(max_{WC})-1} \leftarrow 1 - errorRate$
  $WI_{(max_{WC})-1} \leftarrow WB$
end

Tab. 3.2. Accuracy is the ratio of correctly predicted instances to the total instances which can be computed as:

$$accuracy = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \tag{3.1}$$

While precision concerns about the ratio of correctly positive instances with respect to all predicted positive instances. Then to minority and majority, $precision_{min}$ and $precision_{maj}$ are as follows respectively:

$$precision_{min} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{3.2}$$

$$precision_{maj} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Negative}} \tag{3.3}$$

Table 3.2: Confusion Matrix

	Predicted Class		
Actual Class		Positive	Negative
	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

If the precisions of both the majority class $Precision_{maj}$ and minority class $Precision_{min}$ of the existing classifier are less than 0.5 which indicates that the old classifier is worse than a random guess, we believe that new classifier is needed. But it is notable that considering the possibility of imbalance issue, we train new classifier on resmapled instances which are not same to original instances in this WB. It will be elaborated later. Then, if the $Precision_{maj}$ and $Precision_{min}$ of the newly trained classifier are both greater than 0.5, we add it into the WC. In addition, its $weight$ is set to $1 - errorRate$, and instances in current WB used to train this new classifier are saved in the WI. Otherwise, if the new classifier fails in the precision test, it will be discarded.

3.3.4 Multi-window Ensemble Learning

The overall processing conducted by our Multi-window ensemble learning (MWEL) is shown in Algorithm 3.3. When the first window arrives, WC and WM are initialised with empty sets; and the first classifier is trained on the first window. If the $precision_{maj}$ and $precision_{maj}$ satisfies the conditions mentioned above, (Algrithm 3.3 line 7), we add this classifier to WC as the first sub-classifier and set its weight to 1. WI is also be filled with instances in this window. Later on, for the following sliding windows, whether or not training or adding a new sub-classifier to WC is determined by the WC updating strategy described in Subsection 3.3.3. If WC is considered to be updated with a new classifier c trained on current WB, instances used to train old sub-classifiers and classifiers' weights are necessary to update together as shown in Algorithm 3.2. Whether WC is updated or not, we always update WM according to rules described in Subsection 3.3.2. As show in line 35 of Algorithm 3.3, the WM is updated for current WB before next window of instances arrives. Here, we simplified the procedure saving only the latest minority instances for the space and time efficiency rather than keeping all minority instances [148] or select the nearest instance to the current window [3]. The primary procedure is described in Algorithm 3.1.

In particular, when the next window arrives at time step t, it become to the current window denoted by WB_t. We must update the weights of all sub-classifiers in WC before we use them on WB_t to fit the concept in the window since the window. We firstly compute and normalize the similarity between the current window and existing windows in

Algorithm 3.3: Multi-window ensemble learning (MWEL)

Input: Instances $D = \{(X_0, l_0), (X_1, l_1), \cdots, (X_n, l_n)\}$ and size limits M_b of WB, max_{WC} of WC, max_{WM} of WM
Output: WM, WC, WI, $Weight$ and predicted labels $L = l_0, l_1, \cdots, l_{M_b}$

Initialise: $WB = \{\}$, $WM = \{\}$, $WC = \{\}$, $WI = \{\}$ and $Weight_i = 1, i = 0, 1, \cdots, max_{WC}$
for *each sequencial instance in* D **do**
  $WB \leftarrow getBatchOfInstances(max_{WB})$
  if $|WC|$ *is empty* **then**
    $c \leftarrow trainClassifier(WB)$
    $precision_{min}, precision_{maj}, errorRate, L \leftarrow classify(WB, c)$
    if $precision_{min} > 0.5$ *and* $precision_{maj} > 0.5$ **then**
      $WC = WC \cup c$ with Algorithm 3.2
      $Weight_1 = 1$
    else
      $WB' \leftarrow$ resample instances in WB using Algorithm 3.4
      $WB' = WB' \cup WM$
      $c' \leftarrow trainClassifier(WB')$
      $precision_{min}, precision_{maj}, errorRate, L \leftarrow classify(WB, c')$
      if $precision_{min} > 0.5$ *and* $precision_{maj} > 0.5$ **then**
        $WC = WC \cup c'$ with Algorithm 3.2
        $Weight_1 = 1$
      end
    end
  else
    for *each window of instances in* WI **do**
      compute $Weight_i$ for each subclassifier in WC with Eq. 3.4
    end
    $precision_{min}, precision_{maj}, errorRate, L \leftarrow classify(WB, WC)$
    if $precision_{min} < 0.5$ *or* $precision_{maj} < 0.5$ **then**
      $WB' \leftarrow$ resample instances in WB using Algorith 3.4
      $WB' = WB' \cup WM$
      $c' \leftarrow trainClassifier(WB')$
      $precision_{min}, precision_{maj}, errorRate, L \leftarrow classify(WB, c')$
      if $precision_{min} > 0.5$ *and* $precision_{maj} > 0.5$ **then**
        $WC = WC \cup c'$ with Algorithm 3.2
      end
    end
  end
  $WM \leftarrow updatethe$WM using Algrithm 3.1
end

WI to modify the weights of sub-classifiers in WC based on the following observation:

Observation 3.1. *The greater the similarity between window* W_i *and window* W_j*, the closer are*

the corresponding concepts within them. Therefore, those sub-classifiers trained on windows more similar to the current window should be given larger weights as follows.

$$Weight_i = \frac{Weight_i}{(1 - sim(WB, WI_i) + \epsilon)}, i = 1, 2, ..., max_{WC} \tag{3.4}$$

Here, $Weight_i$ is the weight of the ith sub-classifier in WC, $sim(WB, WI_i)$ is the similarity between the current window and WI_i, and ϵ is a small constant. Moreover, this observation can be used to target reoccurring concept drift because an existing sub-classifier corresponding to a reoccurring concept will obtain a larger weight, which is expected to provide better results.

For an instance Ins_k in WB, the predicted class label is determined by a majority weighted voting scheme. The output value of the majority weight voting scheme can be a soft label representing to what extent Ins_k belongs to a class or an exact value of 0 or 1 indicating Ins_k should belong to a class or not. We adopt the second method that output the hard class label for each instance. Using $WC_i(Ins_k)$ to denote the classification of new instances using the sub-classifiers in c_i we have the following formulation:

$$l_{Ins_k} = \begin{cases} 1, & \sum_{i=1}^{|c|} Weight_i \cdot c_i(Ins_k) > 0.5 \\ 0, & \text{else} \end{cases} \tag{3.5}$$

Algorithm 3.4: Resample

Input: WB, expected imbalance rate γ
Output: sampled instances S

$truePositive, trueNegative, falsePositive, falseNegative \leftarrow classify(WB)$
while $\frac{|Minority|}{|Majority|} < \gamma$ **do**
    $sampleMinority \leftarrow SMOTE(falseNegative)$
    $sampleMajority \leftarrow randomUnderSample(trueNegtive)$
end
$S = truePositive \cup falsePositive \cup sampleMinority \cup sampleMajority$

After classification of the current WB with existing sub-classifiers in WC, we compare the predicated label with the true label of each instance in WB not merely to judge

whether new sub-classifier is needed, but also see whether there is an imbalance issue. If one or both of $precision_{min}$ and $precision_{maj}$ is less than 0.5, an imbalance issue is very likely to exist within WB, and all instances in WB will be resampled (Algorithm 3.3, line 11 and 26). The resampling is presented in Algorithm 3.4. Correctly classified positive instances (Ture Positive) remain unchanged because they are not likely to be helpful in increasing the precision, while wrongly classified positive instances (False Negative) are over-sampled to increase their quantity when used to train a new sub-classifier. Here, SMOTE [29] is used to oversample minority instances; while random under-sampling will be applied on correctly classified majority instances (True Negative) to decrease the quantity of majority. The iteration of resampling stops when the imbalance ratio gradually decreases to the predefined threshold. Finally, correctly classified minority instances, wrongly classified majority instances, oversampled wrongly clas-sified minority instances, and under-sampled correctly classified majority instances consist of the new training set. New classifier will be trained on it and tested on original instances of WB. If $precision_{min}$ and $precision_{maj}$ of this new classifier satisfy the condition that both greater than 0.5, we add it to WC with Algorithm 3.2. Note that the new classifier will receive the largest weight by default on the basis of the following observation.

Observation 3.2. *Considering the influence of time factor, the latest classifier is more likely to represent the concept of the current window, and, therefore, this classifier deserves a larger weight.*

In general, the majority weight voting method is reasonable and comprehensive method by integrating the factors of time, similarity between instances and accuracy and precision of classifiers.

3.4 Experiment and Evaluation

We evaluated our approach on both real-world and synthetic datasets using a variety of metrics. Experiments were initially conducted to obtain optimal window sizes for the different datasets. The results of the proposed method were compared with those of two existing approaches.

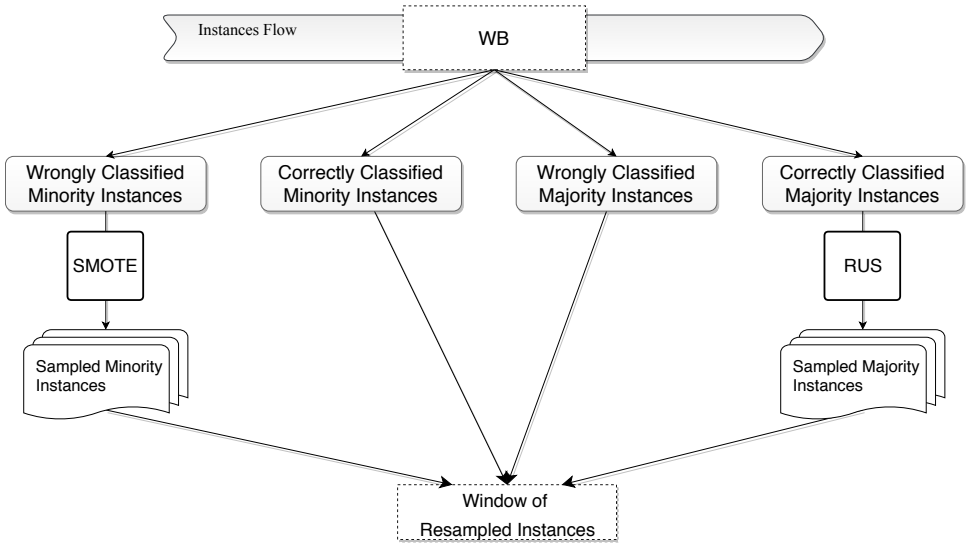


Figure 3.4: The Resampling procedure.

Table 3.3: Dataset Statistics

ID	Name	# of Instances	# of Positive Instances	# of Negative Instances	# of Attributes	Positive Index	Negative Index	Imbalance Rate
ele	Elec	45,312	19,237	26,075	8	1	2	1:1.35
for	Forest	286,048	2747	283,301	54	4	2	1:103
air	Airlines	539,383	240,264	299,119	7	2	1	1:1.24
mus	Mush	8,124	3,936	4,188	23	1	2	1:1.06
th1	Thyroid1	6,832	166	6,666	21	1	3	1:40
th2	Thyroid2	7,034	368	6,666	21	2	3	1:18
po1	Poker1	454,958	39,706	415,252	11	3	1	1:11
po2	Poker2	367,967	17,473	350,494	11	4	2	1:21
gcd	GCD	100,000	24,652	75,348	20	2	1	1:3
scd	SCD	100,000	25,178	74,822	20	2	1	1:3
rcd	RCD	100,000	24,280	75,720	20	2	1	1:3

3.4.1 Datasets

As shown in the first eight rows of Tab. 3.3, we conducted experiments on eight real-world datasets. The Elec, Forest, Airlines, Poker1, and Pocker2 datasets are publicly available at MOA datasets[1]. The Mushroom, Thyroid1, and Thyroid2 datasets are publicly available at the UCI Machine Learning Repository[2].

- Elec (ele) was collected from the Australian New South Wales electricity market to predict the rise and fall of electricity prices, where prices are affected by market supply and demand, and are set every five minutes.

[1]http://moa.cs.waikato.ac.nz/datasets/
[2]http://archive.ics.uci.edu/ml/datasets.html

- Airlines (air) was used to predict whether or not a flight would be delayed.
- Mushroom (mus) includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms, where each species is identified as either edible or poisonous.
- Thyroid datasets (th1, th2) were used to identify whether or not a patient has thyroid disease. To form naturally imbalanced data, we selected class 1 and class 3 to form th1 and class 2 and class 3 to form th2.
- Poker datasets (po1, po2) consist of five playing cards drawn from a standard deck of 52 cards. Each card is described using its suit or rank.

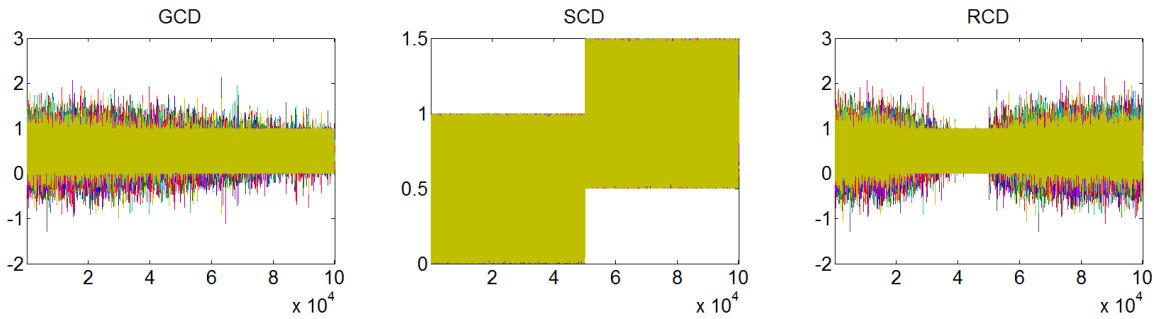
Moreover, experiments are conducted on three types of synthetic datasets generated by algorithms provided in MOA[3] which is shown in Fig. 3.5. During the instance generation, we randomly removed some instances from one class to form imbalanced datasets.

- Gradual concept drift (gcd), where a gradual concept drift begins at the 30,000th instance to the 100,000th instance.
- Sudden concept drift (scd) takes place suddenly at the 50,000th instance, and the dataset maintains the new concept until the 100,000th instance.
- Reoccurring concept drift (rcd) occurs at the 30,000th instance, and begins to shift back to the original concept at around the 50,000th instance until the 100,000th instance.

3.4.2 Evaluation criteria

We employ accuracy (AC), precision, Recall, F1 and G-mean (G-MEAN) to evaluate our method in this chapter. As discussed in Subsection 3.3.3, the accuracy may be skewed to the majority. To properly measure how many positive instances our algorithm labelled

[3]http://moa.cms.waikato.ac.nz/



(a) (b) (c)

Figure 3.5: Distribution of synthetic datasets given in Tab. 3.3.

are truly positive and how many negative instances our algorithm labelled are truly negative, we us the following two criteria:

$$Recall_{min} = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3.6}$$

$$Recall_{maj} = \frac{TrueNeagtve}{TrueNegative + FalseNegative} \tag{3.7}$$

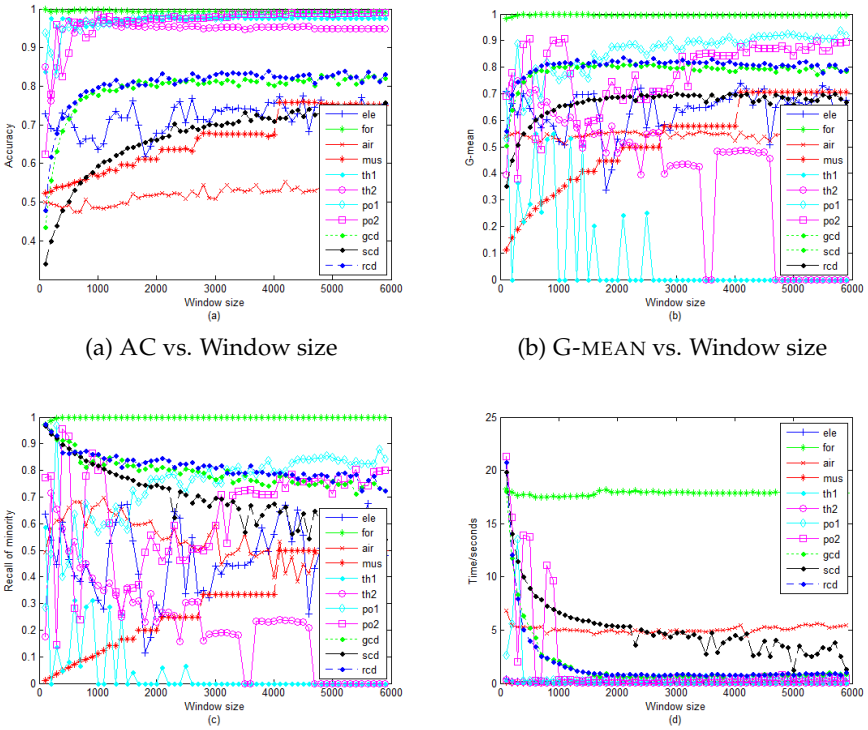
Since we consider that the positive instances are easily dominated by negative instances under imbalanced circumstance, we put more attention on $Recall_{min}$. However, we do not want to ignore the majority when we sample more minority instances. We use G-MEAN that considers the recalls of both minority $(Recall_{min})$ and majority $(Recall_{maj})$ together, and therefore it will only be large when they are large, which is a better choice under imbalanced conditions.

$$G - mean = \sqrt{Recall_{min} \times Recall_{maj}} \tag{3.8}$$

Then for the measurement of entire data stream, we adopt a previously adopted strategy [100], that assumes the number of total instances is n and n/M_b batches will be input sequentially and uses the average performance over all batches in the data stream as follows.

$$F = \frac{1}{\lceil n/M_b \rceil} \sum_{i=1}^{\lceil n/M_b \rceil} f_i$$

$$f, F \in \{Accuracy, Precision_{min}, Precision_{maj},$$



(a) AC vs. Window size

(b) G-MEAN vs. Window size

(c) Minority Recall vs. Window size

(d) Time vs. Window size

Figure 3.6: Sliding window size for different datasets.

$$Recall_{min}, Recall_{maj}, F1_{\min}, F1_{maj}, G-mean\} \tag{3.9}$$

Here, f is the indicator of each sliding window, and F is the average indicator of the data stream. We compared all the indicators in our experiments, but, owing to length restrictions, we report here only the results regarding AC, G-MEAN, $Recall_{min}$, and processing time. Many real-world applications are expected to accomplish necessary processing within a finite period, where a batch must at least be processed before the next batch arrives. Therefore, data streaming algorithms require a tradeoff between efficiency and effectiveness.

3.4.3 Sliding window size setup

As discussed in Subsection 3.3.2, there are not widely accepted standards available for selecting an optimal sliding window size with regard to different types of datasets. A larger window size provides a smaller number of windows with respect to a specific

dataset length, which reduces the frequency of classifier training, whereas, contrarily, a smaller window size introduces a greater number of windows with respect to a specific dataset length, increasing the frequency of classifier training. Moreover, a smaller window size also provides less training time for each classifier. Therefore, in present study, experiments were first conducted to determine the optimal sliding window size for different datasets.

Fig. 3.6 shows how the sliding window size affects the classification results, and substantial differences in the various indicators are observed for different datasets. For instance, in Fig. 3.6a, the for, th1, th2, po1 and po2 datasets exhibit quite high AC when the window size is 1000, and the AC remains relatively stable with increasing window size in datasets for, th1, po1 po2, but decreases slowly in the th2 dataset.

However, as shown in Fig. 3.6b, the maximum G-MEAN value for the for dataset occurs at a window size of 500, whereas maximum values are obtained at 400 and 600 for the th1 and th2 datasets, respectively. By comparing the $Recall_{min}$ values shown in Fig. 3.6c, we find that, along with the window size increases, the $Recall_{min}$ values for th1 and th2 datasets, which lack positive instances, decline sharply, leading to decreasing G-MEAN values. This indicates that the density of minority instances be-comes increasingly sparse with increasing window size. Furthermore, we note from Fig. 3.6d that an increasing window size initially decreases the training and classifying time; however, for a window size greater than 1000, the time cost exhibits a general trend of slow growth for all datasets considered except scd. For a fixed data stream length, the processing time is the product of two parts: the training and classifying time of each window and the number of windows. The initial reduction in the processing time is due to the decreasing number of windows, whereas the longer processing time required for each window results in the general rise in later stages with increasing window size.

Based on considerations of the effectiveness and processing efficiency, we establish the optimal window size for each dataset. The values are listed in Tab. 3.4, and the following experiments were conducted according to this standard.

It is necessary to point out that, in actual applications, it is unrealistic to calculate the optimal sliding window size in advance. As an alternative, the optimal sliding window size can be determined based on a small set of available instances. Here, we focus on

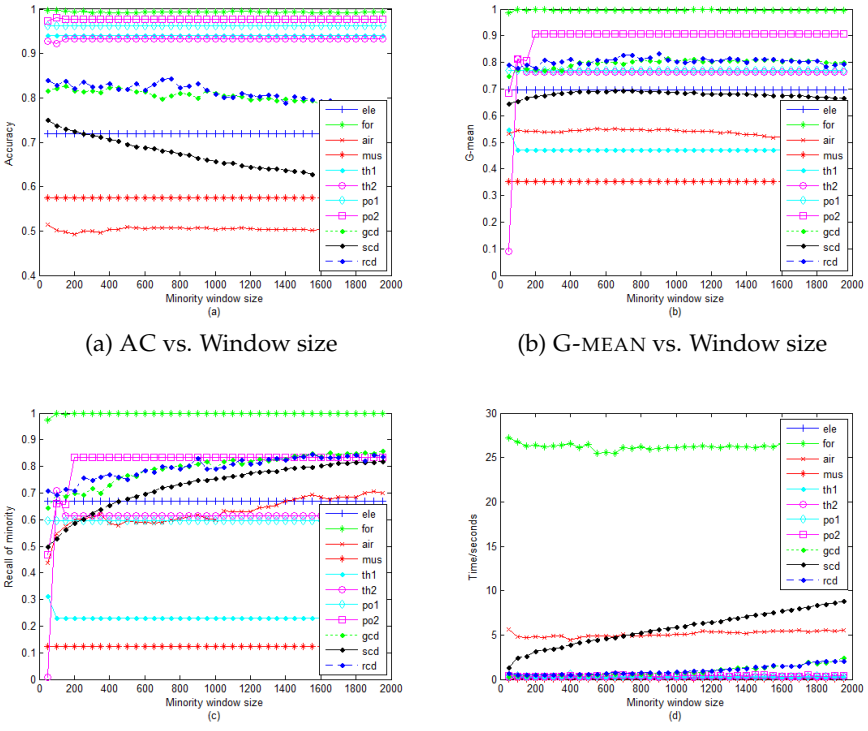
Table 3.4: Optimal sliding window sizes for different datasets

	ele	for	air	mus	th1	th2	po1	po2	gcd	scd	rcd
Window size	1300	1300	800	900	100	200	600	1000	1400	1400	1300

examining the effect of window size on different datasets, and a self-adaptive optimal sliding window is reserved for future study.

3.4.4 Minority window size setup

We examined the influence of the minority window size on the classification performance, and present the results in Fig. 3.7. The results indicate that most datasets are insensitive to the minority window size. However, the mus, gcd, scd, and rcd datasets exhibit a decreasing AC with increasing minority window size, as shown in Fig. 3.7a. As shown in Fig. 3.7b, the G-MEAN values of most datasets, except air, gcd, scd, and rcd, remain stable around a window size of 100. Since the minority win-dow is designed to im-



(a) AC vs. Window size

(b) G-MEAN vs. Window size

(c) Minority Recall vs. Window size

(d) Time vs. Window size

Figure 3.7: Minority window size for different datasets.

Table 3.5: Optimal minority window sizes for different datasets

	ele	for	air	mus	th1	th2	po1	po2	gcd	scd	rcd
Window size	500	700	700	100	300	300	100	200	800	800	900

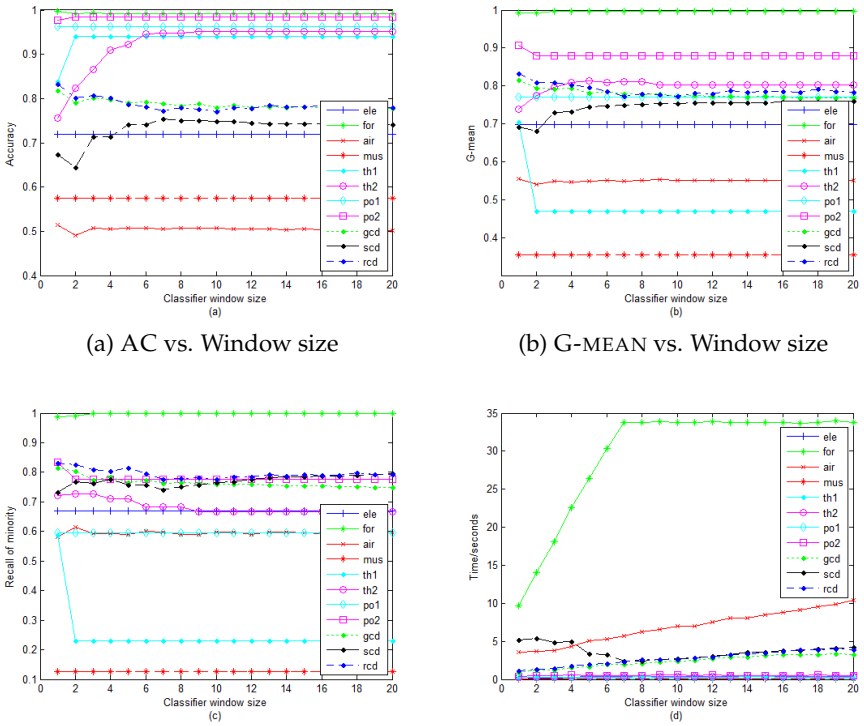
prove the probability of identifying positive class instances under imbalanced conditions, the results in Fig. 3.7c demonstrate that an increasing window size increases the values for most datasets. However, the for, ele, th1, and mus datasets remain nearly unchanged regardless of the window size because minority instances within these four datasets are distributed more evenly than in the other datasets. In addition, the ele dataset has a low imbalance rate (1:1.35), and, thus, the window size has little influence on.

Fig. 3.7d indicates that the processing time typically increases as the minority window size increases, although the processing times of th1 and th2 remain nearly unchanged because the total numbers of minority instances within these two datasets are only 166 and 368. Based on considerations of effectiveness and efficiency, we established the optimal size of the minority window for each dataset, as listed in Tab. 3.5, and the following experiments were conducted according to this standard.

3.4.5 Classifier window size setup

Because we employ an ensemble classifier and weighted majority vote, experiments were conducted to determine the optimal classifier window size, and the results are show in Fig. 3.8.

As show in Fig. 3.8a, the accuracies of th2 and scd initially increase with increasing classifier window size, and then remain stable, whereas gcd and rcd are observed to decrease, particularly for small window sizes. The other datasets however appear to be insensitive to the classifier window size. The G-MEAN values shown in Fig. 3.8b exhibit very similar trends to those of the AC. However, the $Recall_{min}$ values shown in Fig. 3.8b decrease slowly with increasing classifier window size, indicating that a large number of sub-classifiers is not always a good choice. The processing time consumed with respect to the classifier window size is show in Fig. 3.8d. Based on considerations of effectiveness and efficiency, the optimal classifier window sizes for most of the datasets considered are less than 6.



(a) AC vs. Window size

(b) G-MEAN vs. Window size

(c) Minority Recall vs. Window size

(d) Time vs. Window size

Figure 3.8: Classifier window size for different datasets.

3.4.6 Comparison with existing methods

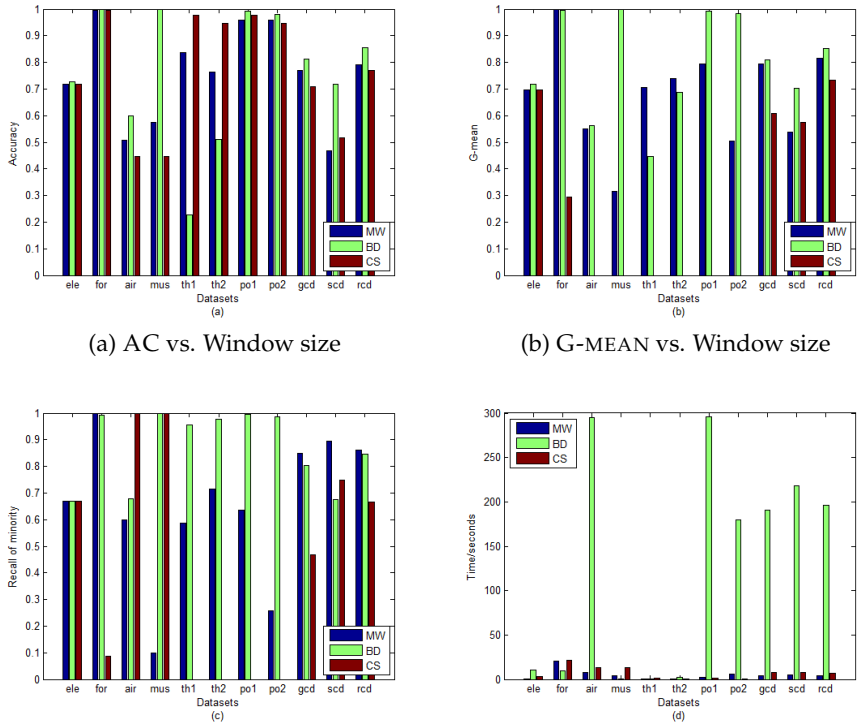
The performance of the proposed MWEL (MW) method is compared with two state-of-the-art approaches, denoted as the BD [100] and CS [190] approaches.

- BD propagates positvie instances and misclassified instances in the negative class to make the boundary instances as distinguishable as possible, thus increasing precision while minimally affecting recall.

- CS treated conditions with or without concept drift differently to apply different classifier updating polices after balances the dataset with sampling method and trians new sub-classifier on the training set. Specifically, when the concept drift is detected, data instances represented new concepts are injected into old training sets corresponding to existing sub-classifiers with a certain probability p to re-train sub-classifiers for the ensemble classifier. While no concept drift found, CS selectively update the sub-classifiers according to their AUC value. Follow the author, we set

$p = 0.75$.

We choose them for the reason that the authors claimed that their approaches performed better than those to which they were compared, but these two methods have not been compared directly with each other under equivalent evaluation metrics. The results of the comparison are shown in Fig. 3.9.



(a) AC vs. Window size

(b) G-MEAN vs. Window size

(c) Minority Recall vs. Window size

(d) Time vs. Window size

Figure 3.9: Distribution of synthetic datasets given in Tab. 3.3.

Fig. 3.9a shows that the AC of MW is comparable those of BD and CS for most of the datasets considered. Because MW focuses greater attention on minority instances, the precision of the majority instances suffer, which affects the AC. Fig. 3.9b shows that the G-MEAN values of MW are generally close to those of BD; and MW outperforms CS for all datasets. When considering G-MEAN and $Recall_{min}$ in Fig. 3.9c together, we note that, for th1 and th2, BD exhibits obvious advantages in relative to MW, while the advantages are reversed with respect to G-MEAN. BD obtains better $Recall_{min}$ results because the approach accumulates all positive instances to build successor sub-classifiers. However, an excessive number of minority instances may overwhelm the majority, and,

Table 3.6: Wilcoxon signed rank test statistics for comparing methods

	$\mathbf{p-value}$			
	AC	G-MEAN	$Recall_{min}$	Time
MW vs. BD	0.27832031	0.14746094	0.14746094	0.03222656
MW vs. CS	0.46484375	0.00488281	0.76464844	0.00976563
BD vs. CS	0.14746094	0.00097656	0.36523437	0.32031250

thus, the G-MEAN results suffer. In addition, a large number of minority instances also incurs greater processing time, as shown in Fig. 3.9d. MW is much more efficient than BD, except for datasets for and mus, which are nearly balanced, and is consistently more efficient that CS except for datasets po1 and po2. In actuality, the processing time for BD on the air dataset was greater than twenty hours, but we set it to 280 s as the upper bound for display purposes.

In addition to the visual comparison given in Fig. 3.9, we also applied the Wilcoxon signed rank test to compare the statistical differences among MW, BD and CS, and the corresponding p-values are listed in Tab. 3.6. Although BD achieves larger G-MEAN and $Recall_{min}$ values than MW for some datasets, no statistically significant difference is observed among the three methods in terms of AC with a significance level $\aleph = 0.05$. However, for G-MEAN and , MW is better than CS. As for processing time, MW performs better than BD and CS at the given significance level.

3.5 Summary

In this chapter, we proposed a multi-window based ensemble learning (MWEL) framework to predict the class labels of newly arriving instances for classification of imbalanced streaming data. We utilize multiple windows to preserve the current data batch, selected positive instances, and the set of latest sub-classifiers as well as the corresponding sets of instances used to train each sub-classifier. Moreover, before predicting the label of incoming instances, we update the weight of each sub-classifier by calculating the similarity between the current window and previous windows used to train each sub-classifier. A weighed majority voting strategy is then used to predict the class label.

A new sub-classifier is trained only when the current ensemble classifier exhibits low precision for one or both minority and majority classes. Under conditions of substantial imbalance, we oversampled minority instances and under-sampled ma-jority instances. Extensive experiments on both real-world datasets and synthetic datasets demonstrated that our method can process imbalanced stream data efficiently and effectively, and, in certain respects, outperforms existing methods, particularly with respect to processing time.

Chapter 4

Semi-supervised Topic Detection for Text stream in Online Social Context

As an important research direction of text mining, topic detection and tracking (TDT) under modern media circumstances has been dramatically innovated with the ever-changing online social network and inconspicuous connections among participants in the Internet communities. Apart from the word features of analysing materials, such as news articles and personal or professional comments, the auxiliary information attracts increasing attention from the research community. Meanwhile, numerous interrelations hiding in the corpus and corresponding network participants also promote topics' evolving, not only apparent solid connections, for example two documents that have the same tags and two participants who are close colleague, but also weak connections which are often unspectacular and with little causal relations. Therefore, answering the question how to exploit and use this hidden information in the social network will extend the landscape of research on TDT. In this paper, we employ the followers' preference extracted from Twitter as the social context that accompanied the corresponding news articles and explore the interior links among them to develop a non-negative factorization methods with semi-supervised information derived from the original data. Furthermore, experiments are conducted on real and semi-synthetic data sets to test the performance of topic detection and feature selection for further documents clustering with k-NN algorithm. The results demonstrate that the proposed method outperforms baselines and state-of-the-art methods.

4.1 Introduction

TOPIC detection and tracking (TDT) is no doubt a well-studied research field under the circumstance of information overload, as well as the rise of new media, for example social media and we media. The various platforms of the latter, like the online social networking services, have provided the means that people can find things out with their own version of the truth for themselves and share their views instantly, which

dramatically changed the way how society is informed. However, as the classic 80-20 rule was widely observed in many fields, not just business and management, traditional medium still reigns supreme, spreading the so-called most powerful ideas and only the perspectives they want people know to the gullible public. In spite of this, the new media actually innovate the information transmission on width and depth, as well as the speed. Therefore, the research of TDT gradually attempts to combine text analysis and social link together to robustly design topic models. Many benefits come with this progress. First of all, it could alleviate some of the multiple problems caused by the lexical variation, lexical sophistication and lexical errors people used to describe a particular event or idea to some extent. And secondly, it would be a more realistic version of the analysis for text content and social context, which mutually restricts and supports each other. In addition to these, it is not in contradiction with the research of community detection and social links mining in social media, but enhance each other and open a new avenue for both of them.

Most of the traditional content-based topic discovery methods, including matrix decomposition based methods [25,91,112] and probabilistic approaches [20,72,147], mainly focus on textual content mining for latent topics, rather than other incidental information, such as the geography location, time and user related records. Nevertheless, the real implication of words is highly dependent on the context, in which they occur, not only because of the complexity of vocabulary, but also because of the various expressing forms. Different time and circumstances may give spectators different perspectives even for the same article. For example, we will have absolutely different thoughts to Donald Trump's profiles in the past year and several years ago. An article of Trumps profiles was merely a matter of celebrity resume several years ago. Maybe it was a little piece in a stack of commercial documents in five years ago, which might involve corporate development, investment, financing and so on. While seeing his profile in the last year, the most likely topic was about the US election under America's political problems and it has been more of a national political issue even concern to other countries around the world after his election victory. That is to say, his profile plays a role in affecting the readers' political stands, rather than a normal description of a celebrity nowadays. We could observe similar changes in many other occasions, most of which are always ignored by

many classical topic discovery algorithms.

Normally, the internal coherence of user's preference in a particular period helps us to find groups of people that share same common interests and topics. Based on this "issue of common concern" view, Kalyanam etc. [81] assumes that the latent topics can also be expressed by users' distribution apart from textual contents. Hence, by combining these two parts, their method works out more precise topics. But in the meantime, the algorithm complexity increases dramatically as the amount of users becomes very huge. The distribution of users plays a defining role, which is always changing with the arising, evolving and defusing of topics over time. Moreover, the total amount of users is immense, comparing with the quantity of textual features caused by the enormous volume of participants the online social network brings. In a twist, the previous hypothesis can be understood as the perspective of "users with common shares". In other words, the distribution of posts and shares can also represent users' preferences. In this paper, we will use two matrices to denote the relational information of the textual content and the social context, then factorize each of them with a generalized function, bonding factors from different matrices together. To associate above two parts, we propose a novel non-negative matrix factorization (NMF) approach based on collective matrix factorization approach [149]. To our knowledge, existing NMF algorithms in the domain of topic discovery are unsupervised learning which lack features to exploit implicit associations inside of the data matrix. To retrieve and leverage these relations, we consider the constraint propagation, which has applied to spectral [107] and validate on image clustering [171] recently.

For the research of topic detection, it is infeasible to utilise either fully labelled training set or a part of labelled data as the works in image processing and face recognition [102, 107,171] did, not merely due to the expensive cost [16], but because the topic labels, which is known as "hard" or inherent constraints, is the target that we seek in its own right. This implies that all the data points we have are unconstrained with respect to each other in the initialization phase of textual content and social context. In this paper, to overcome this unfavourable precondition, we propose a constraint propagation scheme to explore the pairwise constraints within data points which are perceived as "weak" or potential constraints that may be influenced by certain circumstances. We construct two weight

matrices respectively on the initial data matrices' local structures in the first step and propagate the restriction between two data points over the whole matrix in both vertical and horizontal directions until convergence. New weight matrices will be developed later with the pairwise constraints for each original data matrices, and will be applied as regularization terms to preserve and consolidate the geometrical structure in the low-dimensional representation space in the following collective NMF step. Furthermore, we propose a locally weighted matrix factorization (LWNMF) method on both textual content and social context matrices to obtain reliable approximation. This weight scheme is used to precisely measure the geographical distance between the original data points and the approximate value, which can be used in the next iteration as an updated weight matrix to locally minimize the cost function.

The main contributions reported in this Chapter are summarised as follows:

- When discover topics in online text streams, the auxiliary information, the corresponding social context in particular, is considered as a influence factor because of the challenges the online social media brings. We propose a collective NMF method for multi-domain to comply with this trend.

- We leverage the constraint propagation scheme to adequately exploit the hidden supervised information in both text content matrix and social context matrix. The supervised information will helps preserve the geometric structure in low-dimensional representation space during the optimisation of the collective NMF objective function.

- Differing from the state-of-the-art collective NMF method on topic detection task, LETCS [81], which treats the joint matrices as an extra feature to the same domain and ignore the fact that users in the matrix reflecting the social context are practically impossible to complete at the beginning, our method is capable of processing cases with changing and evolving groups of users. The user preference will be used as the auxiliary information which is independent from the corpus to some extent. In other words, our method can be also adopted to scenarios without social context.

- We implement the proposed Collective NMF method with the Constraint Propagation (NMFCP) on real datasets and synthetic datasets over different scenarios.

Experimental results show that our method can improve the topic detection and documents clustering performance greatly and also validate the effectiveness of the constraint propagation scheme.

- We afterwards propose a locally weighted scheme to precisely measure the geographical distance between the original data points and the approximate data points after each iteration. The weight will be used to better approximate certain parts of the data matrix in later iterations. Applying this weight scheme on N-MFCP is to seek an improvement of the algorithm stability, which is verified by experiments focusing on topic detection task.

The rest of this paper is organised as follows. Section 4.2 briefly reviews the previous work on topic detection and tracking. We give a preliminary definition of our model in section 3. Section 4 explains our constraints propagation scheme in detail. Section 5 proposes our multi-domains NMF with constraints propagation methods with the updating rules and computational complexity analysis. Section 6 shows the experimental results and section 7 concludes the paper.

4.2 Previous Work

Topic detection and tracking has been a fundamental problem in a wide range of applications, such as news event analysis[108], information discovery [79, 177], social interest discovery [43,99], social emotion learning [197] and expert system [137]. Though methods of TDT in above applications for different domains have been paid particular attentions to, nearly all of them are developed on the basis of latent topic models which represented by pLSA (probabilistic latent semantics analysis [73]) and LDA (latent dirichlet allocation [20]). Latent topic model, generally speaking, aims to model representations to indicate the latent variables, topics in particular, in underlying structure of discrete data. In this paper, we primarily concerned with two roughly categories of topic modelling approaches. One is probabilistic model based approaches [20, 147] and another one NMF-based approaches [81, 157].

Comparing to the basic latent semantic analysis (LSA) [91], pLSA [72] model each topics as multinomial distributions over words, which extend the SVD based LSA with

the statistical distribution. LDA [20] developed pLSA with a Dirichlet distribution on the distribution of topics for each article and words distribution to this topic is multinomial distribution. To exploit the link structure among documents, PHITS [37] extend the pLSA by defining a generative process for citations as hyperlinks. Erosheva et al. propose a similar idea Link-LDA [55]. Nallapati et al. [118] developed pairwise Link-LDA and Link-PLSA-LDA by emphasizing directional citations and asymmetric citing links. For dynamic features in data streams, Blei et al. further extended LDA to Dynamic Topic Model (DTM) [19] which is a representation of discrete dynamic topic model (dDTM). In the counterpart of dDTM, known as continuous time dynamic topic model (cDTM), Wang et al. [169] employed Brownian motion to model continuous evolution topics in sequential time-series data. With the prevalent trend of social media, for example posts and comments on Twitter, some works [115, 165] are proposed to overcome challenges including text shortness, low meaningful description and high velocity stream. Twevent [96] is a feature-pivot methods, in which tweet segments are used to detect and cluster events.

Non-negative matrix factorization (NMF) [94] has been a mainstream method of part-based representation for research communities of information retrieval, pattern recognition and computer vision. Previous works [44, 62] have demonstrated the close connection between NMF and pLSA. Xu [183] proposed a NMF-based document clustering method resorting to LSA [91] idea, in which it does not require the decomposed matrices are orthogonal and it guarantees that feature values of all documents are non-negative in all latent semantic directions and each document can be presented as additive combination of the base latent topics. Cao et al. proposed Online-NMF in [25] to detect and track the moving of latent factors in data streams, considering multiple topics coocurrence. A quite similar optimization strategy to NMF was used as dictionary learning in [82], which identifies the incoming observation documents as a novel one or not with ℓ_1-penalty to measure reconstruction error and new dictionary will be learnt with novel documents. Other NMF-based dynamic approaches [139, 168] were proposed to capture the evolving set with temporal regularization terms. Recently, Suh et al. [157] impose ensemble model on NMF-based method to discover more precise local topics under noisy circumstance.

4.3 Preliminary and Problem Formulation

The study of Non-negative Matrix Factorization can be traced back to Lee and Seungs work [94] in which the non-negative constraint was added and parts-based learning of objects was figured out for computers as human brains do. Manifolds of NMF has been developed after that. In this section, we briefly introduce the fundamental contributions that related to our problem made by previous works and formulate our problem followed by.

4.3.1 Standard Non-negative Matrix Factorization (NMF)

Given an input matrix $\mathbf{X} \in \mathbb{R}_{+}^{m \times n}$ with n data points of m dimensional features, the standard NMF decomposes the original $\mathbf{X}$ to two low-rank non-negative matrices $\mathbf{H} \in \mathbb{R}^{m \times k}$ and $\mathbf{V} \in \mathbb{R}^{n \times k}$, whose linear product approximates as close to the original matrix as possible and normally the new rank $K \ll \min(M, N)$. The formulation is as follow:

$$\mathbf{X} \approx \mathbf{H}\mathbf{V}^T$$
$$\underset{\mathbf{H},\mathbf{V} \geq 0}{\operatorname{argmin}} \mathcal{D}(\mathbf{X}, \mathbf{H}\mathbf{V}^T) + \mathcal{R}(\mathbf{H}, \mathbf{V}) \tag{4.1}$$

Where, loss function $\mathcal{D}(\mathbf{X}, \mathbf{H}\mathbf{V}^T)$ quantifies the cost of the approximation and $\mathcal{R}(\mathbf{H}, \mathbf{V})$ is the regularization penalty. From the definition, we can always find two entities and a relation between them. The NMF process, in essence, is a linear regression, which determines the relation description between the two entities by given that the relation exists.

4.3.2 Collective Matrix Factorization (CMF)

By considering the complicated interactions in real application, [149] proposed the collective NMF aims to trade on the correlations between matrices that contain more than one relation and associates the factors involving in different relations together with a generalised-linear link function. Take a two-relation schema as an example: two data matrices, $\mathbf{X} \in \mathbb{R}^{m \times n}$ representing the users-movies relation and $\mathbf{U} \in \mathbb{R}^{n \times l}$ representing the movies-genres relation, we use the shared factor $\mathbf{V}$ in both constructions and have

the decomposition and objective function as follows:

$$\mathbf{X} \approx \mathbf{H}\mathbf{V}^T \text{ and } \mathbf{U} \approx \mathbf{V}\mathbf{G}^T \tag{4.2}$$

$$\underset{\mathbf{H},\mathbf{V},\mathbf{G}\geq 0}{\operatorname{argmin}} \mu\mathcal{D}(\mathbf{X}, \mathbf{H}\mathbf{V}^T) + (1-\mu)\mathcal{D}(\mathbf{U}, \mathbf{V}\mathbf{G}^T) + \mathcal{R}(\mathbf{H}, \mathbf{V}, \mathbf{G}) \tag{4.3}$$

Where $\mu \in [0, 1]$ is a trade-off parameter to weight the relative importance between two relations. If necessary, a third or more relations, movies-actors, can be involved. Then the objective function of CMF can be extended to:

$$\underset{\mathbf{H},\mathbf{V},\mathbf{G},\mathbf{W},\cdots\in\{\mathbf{C}\}}{\operatorname{argmin}} \Big(\mu_1\mathcal{D}(\mathbf{X}, \mathbf{H}\mathbf{V}^T) + \mu_2\mathcal{D}(\mathbf{U}, \mathbf{V}\mathbf{G}^T) + \mu_3\mathcal{D}(\mathbf{Z}, \mathbf{G}\mathbf{W}^T) + \cdots\Big) + \mathcal{R}(\mathbf{H}, \mathbf{V}, \mathbf{G}, \cdots) \tag{4.4}$$

$\mathbf{H}, \mathbf{V}, \mathbf{G}, \mathbf{W}, \cdots \in \{\mathbf{C}\}$ represents constraints that latent factors subject to and trade-off parameter μ_i of $\sum_i \mu_i = 1$ regulates the relative weight among different relations.

Collective Matrix Factorization can be further categorised into two types, multi-view CMF [48] and multi-relation CMF [149]. Let us take CMF in topic detection as an example. The two types of methods combines relevant external information beyond textual information itself in principle; however, they have different assumptions on the newly involved matrix. LETCS [81] represents multi-view CMF. It treats the new factor, users, in the second relation matrix as features of documents in the user matrix; while our method explores users' preference by the combination of documents from the second relation matrix. At the same time, multi-relation CMF utilise the users' preference to constraint the relations of documents in the first relation matrix.

4.3.3 Problem Definition

Though the new articles generates incrementally in real scenario, at a time step, we collect a batch of new articles released in a time interval from the news stream as a corpus represented by a term-document matrix $\mathbf{X} = [X_1, X_2, \cdots, X_n]$, and each document X_i is a tf-idf vector over the vocabulary of m terms. Under this view, the m terms are textual features. As soon as an article releases, reactions of readers to it, such as leaving comments on it and reposting it, will be continually generated as a flow in the social network plat-

forms. We regard these reactions as a user preference reflecting the latent social context and similarly fix the time interval in which reactions are collected as the second relation between articles and users to form a collective matrix factorization problem as follow:

$$\begin{aligned} \text{minimise} \quad & \mu\mathcal{D}(\mathbf{X}, \mathbf{H}\mathbf{V}^T) + (1-\mu)\mathcal{D}(\mathbf{U}, \mathbf{V}\mathbf{G}^T) + \mathcal{R}(\mathbf{H}, \mathbf{V}, \mathbf{G}) \\ \text{subject to} \quad & \mathbf{H}, \mathbf{V}, \mathbf{G} \geq 0 \end{aligned} \tag{4.5}$$

Where, non-negative matrix $\mathbf{U} = [U_1, U_2, \cdots, U_l] \in \mathbb{R}^{n \times l}$, where l users are activated in this time interval and n articles are involved as features. By definition, each column in matrix $\mathbf{U}$ is also a data point, but the preference feature is n dimension.

4.4 Constraint Propagation

As the NMF is an unsupervised method which optimise the convex objective function to obtain good result, the supervised information of the data set are not being used generally. [171] mentioned that class labels and pairwise constraints are two commonly accepted sources of supervised information for a data set. The former, our target, is what fixed in the internal nature of each data point, while the later just gives us a weak pairwise connection of two data points, whether the two points can be linked or not. Intuitively, the later can be derived from the former, for example, points with the same label share a must-link connection and otherwise a cannot-link appears between them, vice versa; however, we cannot do such inverse deduction on the pairwise constraints that embodies its weakness. On the other hand, the pairwise constraints can be obtained more widely and enhanced by some propagation mechanisms in graph theory [107, 171]. In this section, we explain how the constraints will propagate for data points in the original matrices $\mathbf{X}$ and $\mathbf{U}$ from horizontal and vertical view. An illustration can be found in Fig. 4.1.

At the beginning, we construct the initial pairwise constraint matrices $\mathbf{Z}_\mathbf{X} = \{Z_{Xij}\}_{n \times n}$ and $\mathbf{Z}_\mathbf{U} = \{Z_{Uij}\}_{l \times l}$ with Eq. 4.6. Because the constraint propagation procedures of matrices $\mathbf{X}$ and $\mathbf{U}$ will follow the same schema, we demonstrate one matrix $\mathbf{Z}$ instead of both of them here for concision. Then, in the following, unique definitions will not be made for matrices $\mathbf{X}$ and $\mathbf{U}$ separately. For original n-dimensional data point-feature matrix, denotes $\mathbb{C}_i$ as the feature set involved by data point and $\mathbf{Z} = \{z_{ij}\}_{n \times n}$can be initialized

as:

$$z_{ij} = \begin{cases} +1, & r > \alpha \\ -1, & r < \beta \\ 0, & otherwise. \end{cases} \tag{4.6}$$

Where $r = \frac{|C_i \cap C_j|}{|C_i \cup C_i|}$. α and β are the adjustable parameters for deciding whether two data points can be linked by the same label, in our case, same topic. Jaccard similarity coefficient is used here to measure the similarity between the pair of data points.

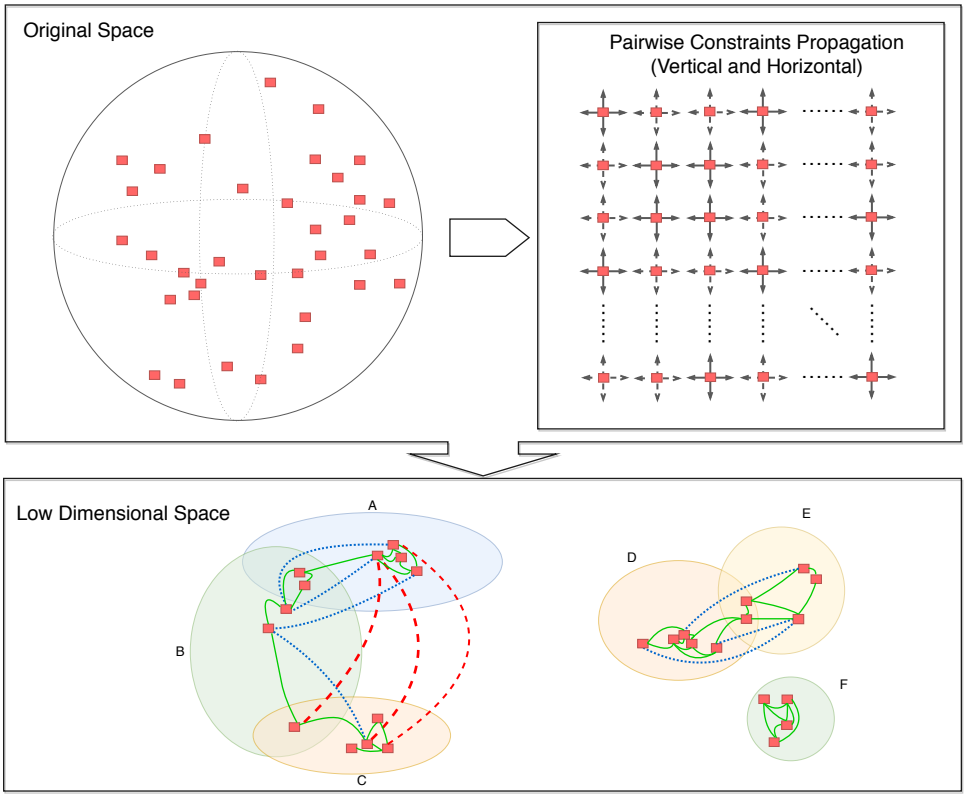


Figure 4.1: After pairwise constraints propagated in vertical and horizontal directions, more connections been found and enhanced.

We have directly obtained the pairwise constraints with little information loss by far. It can be found that some pairs of data points are not constrained (i.e. $z_{ij} = 0$), that is, the corresponding data points x_i and x_j are initially unlabelled. Therefore, the goal is to transductive infer the labels of the unlabelled points [195]. Here we denote the propagated pairwise constraints matrix as $\mathcal{F} = \{f_{ij}\}_{n \times n} : |f_{ij}| \leq 1$. More concretely, $\mathbf{Z}$ is the initial status of $\mathcal{F}$.

The weight matrix $\mathbf{W} = \{w_{ij}\}_{n\times n}$ is used to show the proximity of two data points x_i and x_j. For document processing tasks in IR community, the dot-product weighting is pretty common [24]. Here, we use normalized dot-product, which is also known as cosine similarity of the two vectors to initially define $\mathbf{W}$ as

$$w_{ij} = \begin{cases} \frac{x_i \cdot x_j}{|x_i||x_j|}, & r \geq \alpha \\ 0, & otherwise. \end{cases} \tag{4.7}$$

Construct a symmetric matrix $\mathbf{L} = \mathbf{S}^{-1/2}\mathbf{W}\mathbf{S}^{-1/2}$ with a diagonal matrix $\mathbf{S}$ in which diagonal elements $s_{ii} = \sum_j w_{ij}$ and off-diagonal elements $s_{ij} = 0$. The following iterations are executed from vertical and horizontal perspectives, showing how each data points takes over the information from its neighbours and keep its initial information.

1. For the vertical constraint propagation, iterate

$$\mathcal{F}_v(t) = \delta\mathbf{L}\mathcal{F}_v(t-1) + (1-\delta)\mathbf{Z} \tag{4.8}$$

   until converge, where the parameter $\delta \in (0,1)$ specifies the ratio of information from itself and its neighbours.

2. For the horizontal constraint propagation, iterate $\mathcal{F}_h(t) = \delta\mathcal{F}_h(t-1)\mathbf{L} + (1-\delta)\mathcal{F}_v^*$, where $\mathcal{F}_v^* = (1-\delta)(\mathbf{I}-\delta\mathbf{L})^{-1}\mathbf{Z}$ is the limit of $\{\mathcal{F}_v(t)\}$.

   Proof. By previous definition, $\mathcal{F}(0) = \mathcal{F}_v(0) = \mathbf{Z}$. By the Equation (3), we have

$$\mathcal{F}_v(1) = (\delta\mathbf{L})\mathcal{F}_v(0) + (1-\delta)\mathbf{Z}$$

$$\mathcal{F}_v(t) = (\delta\mathbf{L})^t\mathbf{Z} + (1-\delta)\sum_{i=0}^{t-1}(\delta\mathbf{L})^i\mathbf{Z}. \tag{4.9}$$

   Since $0 < \delta < 1$ and the eigenvalues of $\mathbf{L}$ in $[-1,+1]$, according to the principle of infinite geometric series, we have

$$\lim_{t\to\infty}(\delta\mathbf{L})^t = 0 \text{ and} \lim_{t\to\infty}\sum_{i=0}^{t-1}(\delta\mathbf{L})^i = (\mathbf{I}-\delta\mathbf{L})^{-1}$$

Therefore, $\mathcal{F}_v(t)$ converges to

$$\mathcal{F}_v^* = (1-\delta)(\mathbf{I}-\delta\mathbf{L})^{-1}\mathbf{Z} \tag{4.10}$$

□

3. Denote $\mathcal{F}^* = \mathcal{F}_h^*$ is the final representation of the propagated pairwise constraints, where $\mathcal{F}_h^* = (1-\delta)\mathcal{F}_v^*(\mathbf{I}-\delta\mathbf{L})$ is the limit of $\{\mathcal{F}_h(t)\}$.

Proof. By step 2) we have $\mathcal{F}_h^T(t) = \delta\mathbf{L}^T\mathcal{F}_h^T(t-1) + (1-\delta)\mathcal{F}_v^{*T} = \delta\mathbf{L}\mathcal{F}_h^T(t-1) + (1-\delta)\mathcal{F}_v^{*T}$. That is, the horizontal propagation converges as the vertical propagation did.

$$\mathcal{F}_h^{*T} = (1-\delta)(\mathbf{I}-\delta\mathbf{L})^{-1}\mathcal{F}_v^{*T} \tag{4.11}$$

Hence, the pairwise constraint matrix can be final represented as

$$\mathcal{F}^* = \mathcal{F}_h^* = (1-\delta)\mathcal{F}_v^*(\mathbf{I}-\delta\mathbf{L})^{-1}$$

$$\mathcal{F}^* = (1-\delta)^2(\mathbf{I}-\delta\mathbf{L})^{-1}\mathbf{Z}(\mathbf{I}-\delta\mathbf{L})^{-1} \tag{4.12}$$

□

In the next step, the pairwise constraint matrix $\mathcal{F}^*$ is used to regulate the original weight matrix $\mathbf{W}$. To make it clear, we use a new weight matrix $\tilde{\mathbf{W}} = \{w_{ij}\}_{n\times n}$

$$\tilde{w}_{ij} = \begin{cases} 1-(1-f_{ij}^*)(1-w_{ij}), & f_{ij}^* \geq 0 \\ (1+f_{ij}^*)w_{ij}, & f_{ij}^* < 0. \end{cases} \tag{4.13}$$

The procedures of constraint propagation for text context and user preference matrices are summarised in Algorithm 4.1.

In the following steps, $\tilde{\mathbf{W}}$ will be used for constrained non-negative matrix factorization. From the algorithm we designed above, $\tilde{\mathbf{W}}$ shows the following nice properties:

1. $\tilde{\mathbf{W}}$ is symmetric and non-negative.

Algorithm 4.1: Constraint Propagation for Textual Content Matrix and Social Context Matrix

Input: Article matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, user preference matrix $\mathbf{U} \in \mathbb{R}^{n \times l}$, parameter δ
Output: Weight matrices $\tilde{\mathbf{W}}_{\mathbf{X}} \in \mathbb{R}^{n \times n}$, $\tilde{\mathbf{W}}_{\mathbf{U}} \in \mathbb{R}^{l \times l}$

1 Construct the initial pairwise constraints matrices $\mathbf{Z}_{\mathbf{X}}$ and $\mathbf{Z}_{\mathbf{U}}$ by Eq. 4.6
2 Initialize propagated pairwise constraints matrices $\mathcal{F}_{\mathbf{X}}$ and $\mathcal{F}_{\mathbf{U}}$ with $\mathbf{Z}_{\mathbf{X}}$ and $\mathbf{Z}_{\mathbf{U}}$ correspondingly
3 Define weight matrices $\mathbf{W}_{\mathbf{X}}$ and $\mathbf{W}_{\mathbf{U}}$ by Eq. 4.7
4 Construct symmetric matrices $\mathbf{L}_{\mathbf{X}} = {\mathbf{S_1}}^{-1/2}\mathbf{W}{\mathbf{S_1}}^{-1/2}$ and $\mathbf{L}_{\mathbf{U}} = {\mathbf{S_2}}^{-1/2}\mathbf{W}{\mathbf{S_2}}^{-1/2}$, where diagonal matrix $\mathbf{S_1} = \left\{\sum_j w_{ij}\right\}_{n \times n}$ and $\mathbf{S_2} = \left\{\sum_j w_{ij}\right\}_{l \times l}$
5 Update $\mathcal{F}_{\mathbf{X}}$ and $\mathcal{F}_{\mathbf{U}}$ from vertical and horizontal perspectives by the limit $\mathcal{F}^*$ in Eq. 4.12
6 Compute New weight matrices $\tilde{\mathbf{W}}_{\mathbf{X}}$ and $\tilde{\mathbf{W}}_{\mathbf{U}}$ by Eq. 4.13

Proof. The symmetric feature is inherited from the symmetric of both matrix $\mathcal{F}^*$ and $\mathbf{W}$. By the definition, $w_{ij} \in [0, 1]$ and $\left|{f_{ij}}^*\right| \leq 1$, then we have

$$\tilde{w}_{ij} = \begin{cases} 1 - (1 - f_{ij}^*)(1 - w_{ij}) \geq 1 - (1 - w_{ij}) \geq 0, & f_{ij}^* \geq 0 \\ (1 + f_{ij}^*)w_{ij} > 0, & f_{ij}^* < 0 \end{cases}.$$

□

2. $\tilde{\mathbf{W}}$ is adjusted by $\mathcal{F}^*$ with no distinction between $f_{ij}^* \geq 0$ and $f_{ij}^* < 0$.

Proof. $\tilde{w}_{ij}$ is differentiable at $w_{ij} = 0$ and for all f_{ij}^*, ${d\tilde{w}_{ij}}/{dw_{ij}} = 1 - \left|f_{ij}^*\right|$. □

3. $\tilde{\mathbf{W}}$ shows that the pairwise constraint between two data points has been reinforced after propagation.

Proof. The Equation (8) is a monotonically increasing function of f_{ij}^*. Therefore, the new weight matrix $\tilde{\mathbf{W}}$ increases $\tilde{w}_{ij} \geq w_{ij}$ with $f_{ij}^* \geq 0$ and decreases $\tilde{w}_{ij} < w_{ij}$ with $f_{ij}^* < 0$. □

As mentioned before, more potential pairwise information gained from the constraint propagation procedure is incorporated into new weight matrices $\tilde{\mathbf{W}}_{\mathbf{X}} \in \mathbb{R}^{n \times n}$ and

$\tilde{\mathbf{W}}_{\mathbf{U}} \in \mathbb{R}^{l \times l}$. The properties above guarantee that data points sharing the same topics have relatively larger associated scores and vice versa. Next, we will perform non-negative matrix factorization with weight matrices.

4.5 Semi-supervised Non-Negative Matrix Factorization with Constraint Propagation

According to the traditional non-negative matrix factorization, the original non-negative data matrix will be decomposed to two non-negative matrices, whose linear product approximates as accurate to the original matrix as possible. In our case, we decompose the articles matrix $\mathbf{X} = [x_{ri}] \in \mathbb{R}^{m \times n}$ and the user preference matrix $\mathbf{U} = \left[u_{ip}\right] \in \mathbb{R}^{n \times l}$ in terms of latent topics. We firstly set the number of topics as k, which is usually smaller than n, l and m. Then, we have:

$$\mathbf{X} \approx \mathbf{H}\mathbf{V}^T \qquad s.t. \quad \mathbf{H}, \mathbf{V} \geq \mathbf{0} \tag{4.14}$$

$$\mathbf{U} \approx \mathbf{V}\mathbf{G}^T \qquad s.t. \quad \mathbf{V}, \mathbf{G} \geq \mathbf{0} \tag{4.15}$$

Where, $\mathbf{H} \in \mathbb{R}^{m \times k}$ is a topic matrix. Each column $H_{\cdot k'}$ represents a latent topic expressed as a combination of several terms. Each row $V_i = [v_{i1}, \cdots, v_{ik}]^T$ of the matrix $\mathbf{V} \in \mathbb{R}^{n \times k}$ can be regarded as a low dimensional representation of $\mathbf{X_i}$ under the new basis $\mathbf{H}$, that is how each article is arranged in terms of the latent topics discovered in $\mathbf{H}$. The decomposition of matrix $\mathbf{U}$ is sharing the same $\mathbf{V}$ to fulfil our assumption that users will only be concerned with the news what interest them. Therefore, Equation Eq. 4.15 illustrates how users are grouped in terms of the articles having the same latent topics. Similar to the role of matrix $\mathbf{V}$ in Equation Eq. 4.14, each row $G_p = \left[g_{p1}, \cdots, g_{pk}\right]^T$ of the matrix $\mathbf{G} \in \mathbb{R}^{l \times k}$ can be regarded as the low dimensional representation of $\mathbf{U}$ with respect to the new basis $\mathbf{V}$ and each column in $\mathbf{G}$ represents a community consisting of users who have the similar interest in terms of a topic at this moment.

Here, we choose Euclidean distance to measure the dissimilarity of data points under the lower dimensional representation $\mathbf{V}$ and $\mathbf{G}$. To combine the constraint propagation

information obtained a in Section 4.4, we have the following enhanced distance terms:

$$\phi(\mathbf{V}) = \frac{1}{2}\sum_{i,j=1}^{n} \left\|V_i - V_j\right\|^2 \cdot \tilde{W}_{X_{ij}} \tag{4.16}$$

$$\phi(\mathbf{G}) = \frac{1}{2}\sum_{p,q=1}^{l} \left\|G_p - G_q\right\|^2 \cdot \tilde{W}_{U_{pq}} \tag{4.17}$$

The properties of weight matrices $\tilde{\mathbf{W}}_{\mathbf{X}}$ and $\tilde{\mathbf{W}}_{\mathbf{U}}$ ensure that in low dimensional representation, minimising $\phi(\mathbf{V})$ and $\phi(\mathbf{G})$ can draw similar data points closer and distance data points belong to different topics in geometrical space which consistent with the intrinsic relationships of data points in original data matrix. Since the distances show the geometrical affinity between data points, they are also called geometrical regularization terms. In the following, we incorporate the two terms into our collective matrix factorization problem.

4.5.1 The Objective Function

To maximum the approximate decomposition, our optimization problem is minimise the error between original data matrices and approximation matrices as follow:

$$f(\mathbf{V},\mathbf{H},\mathbf{G}) = \underset{\mathbf{V},\mathbf{H},\mathbf{G}}{\operatorname{argmin}}\, \mu(\left\|\mathbf{X} - \mathbf{H}\mathbf{V}^T\right\|_F^2 + \lambda_1\phi(\mathbf{V})) + (1-\mu)(\left\|\mathbf{U} - \mathbf{V}\mathbf{G}^T\right\|_F^2 + \lambda_2\phi(\mathbf{G})) + \mathcal{R}$$
$$\text{s.t.} \qquad \mathbf{V},\mathbf{H},\mathbf{G} \geq 0 \tag{4.18}$$

Here, we use l1-norm based regularization $\mathcal{R} = \gamma_1\|\mathbf{V}\|_1 + \gamma_2\|\mathbf{U}\|_1 + \gamma_3\|\mathbf{H}\|_1$ to promote the sparsity. The regularization parameters λ_1 and λ_2 controls the contribution proportions of the supervised information parts in our objective function. Parameter $\mu \in [0,1]$ controls the balance between text content part and social context part. If $\mu = 0$, the text content would be ignored and only users be grouped together with topics.

4.5.2 Updating Rules

The objective function $f(\mathbf{V},\mathbf{H},\mathbf{G})$ is not convex in all the variables together because of the non-negative constraint. Therefore, we turn to find its local minima instead of the

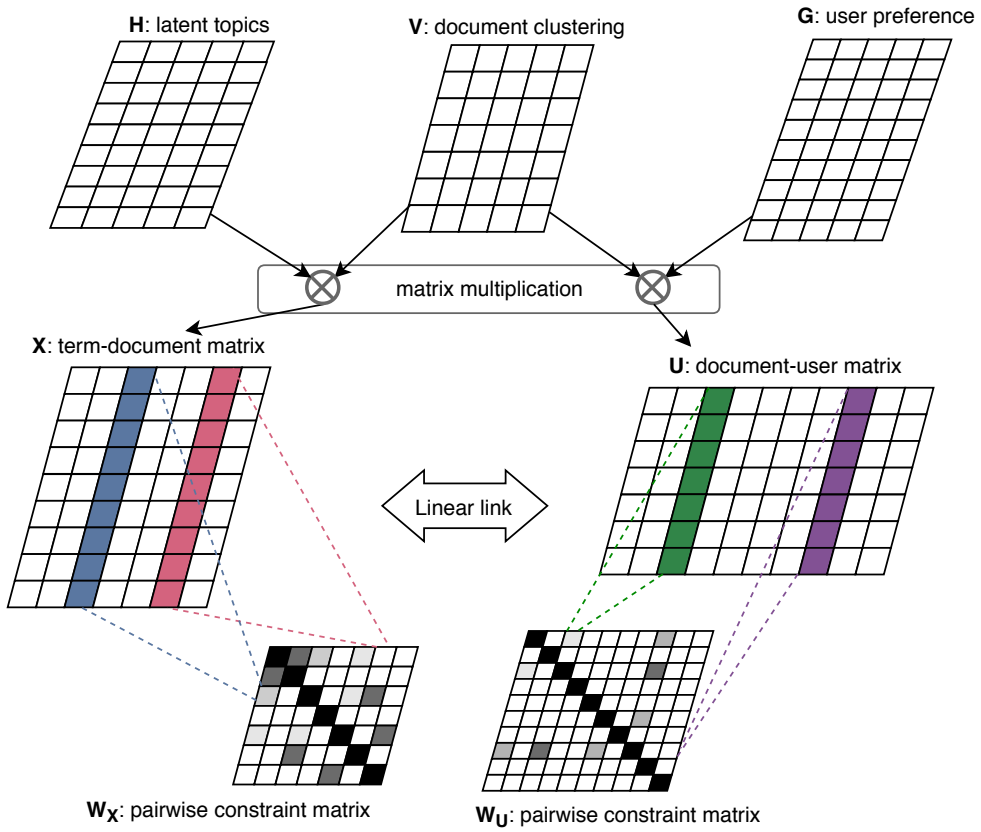


Figure 4.2: An Illustration of Algorithm NMFCP.

global minima with the classical multiplicative updating rules [94]. We first rewrite it as follow:

$$
\begin{aligned}
f(\mathbf{V},\mathbf{H},\mathbf{G}) =& \mu(\left\|\mathbf{X}-\mathbf{H}\mathbf{V}^T\right\|_F^2 + \frac{1}{2}\lambda_1 \sum_{i,j=1}^{n} \left\|V_i - V_j\right\|^2 \cdot \tilde{W}_{X_{ij}}) \\
&+ (1-\mu)(\left\|\mathbf{U}-\mathbf{V}\mathbf{G}^T\right\|_F^2 + \frac{1}{2}\lambda_2 \sum_{p,q=1}^{l} \left\|G_p - G_q\right\|^2 \cdot \tilde{W}_{U_{pq}}) + \mathcal{R} \\
=& \mu(\left\|\mathbf{X}-\mathbf{H}\mathbf{V}^T\right\|_F^2 + \lambda_1 \sum_{i=1}^{n} V_i^T V_i \tilde{D}_{V_{ii}} - \lambda_1 \sum_{i,j=1}^{n} V_i^T V_j \tilde{W}_{X_{ij}}) \\
&+ (1-\mu)(\left\|\mathbf{U}-\mathbf{V}\mathbf{G}^T\right\|_F^2 + \lambda_2 \sum_{p=1}^{l} G_p^T G_p \tilde{D}_{G_{pp}} - \lambda_2 \sum_{p,q=1}^{l} G_p^T G_q \tilde{W}_{U_{pq}}) + \mathcal{R} \\
=& \mu\left(\mathrm{tr}(\mathbf{X}-\mathbf{H}\mathbf{V}^T)(\mathbf{X}-\mathbf{H}\mathbf{V}^T)^T) + \lambda_1 \,\mathrm{tr}(\mathbf{V}^T\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V}) - \lambda_1 \,\mathrm{tr}(\mathbf{V}^T\tilde{\mathbf{W}}_{\mathbf{X}\mathbf{V}})\right) \\
&+ (1-\mu)\left(\mathrm{tr}((\mathbf{U}-\mathbf{V}\mathbf{G}^T)(\mathbf{U}-\mathbf{V}\mathbf{G}^T)^T) + \lambda_2 \,\mathrm{tr}(\mathbf{G}^T\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G}) - \lambda_2 \,\mathrm{tr}(\mathbf{G}^T\tilde{\mathbf{W}}_{\mathbf{U}}\mathbf{G})\right) \\
&+ \mathcal{R}
\end{aligned}
\tag{4.19}
$$

Here, $\text{tr}(\cdot)$ is the trace of a matrix and $\tilde{\mathbf{D}}_\mathbf{V}$ and $\tilde{\mathbf{D}}_\mathbf{G}$ is diagonal matrices whose diagonal elements $\tilde{D}_{V_{ii}} = \sum_{j=1}^{n} \tilde{W}_{X_{ij}}$ and $\tilde{D}_{G_{pp}} = \sum_{q=1}^{l} \tilde{W}_{U_{pq}}$. Denote $\tilde{\mathbf{L}}_\mathbf{X} = \tilde{\mathbf{D}}_\mathbf{V} - \tilde{\mathbf{W}}_\mathbf{X}$ and $\tilde{\mathbf{L}}_\mathbf{U} = \tilde{\mathbf{D}}_\mathbf{G} - \tilde{\mathbf{W}}_\mathbf{U}$, which are symmetric matrices. Then, the objective function can be shown as:

$$\begin{aligned} f(\mathbf{V},\mathbf{H},\mathbf{G}) =& \mu(\text{tr}\left((\mathbf{X}\mathbf{X}^T) - 2\,\text{tr}(\mathbf{X}\mathbf{V}\mathbf{H}^T) + \text{tr}(\mathbf{H}\mathbf{V}^T\mathbf{V}\mathbf{H}^T) + \lambda_1\,\text{tr}(\mathbf{V}^T\tilde{\mathbf{L}}_\mathbf{X}\mathbf{V})\right) \\ &+ (1-\mu)\left(\text{tr}(\mathbf{U}\mathbf{U}^T) - 2\,\text{tr}(\mathbf{U}\mathbf{G}\mathbf{V}^T) + \text{tr}(\mathbf{V}\mathbf{G}^T\mathbf{G}\mathbf{V}^T) + \lambda_2\,\text{tr}(\mathbf{G}^T\tilde{\mathbf{L}}_\mathbf{U}\mathbf{G})\right) \\ &+ \mathcal{R} \end{aligned} \tag{4.20}$$

This variation applied properties $\text{tr}(A+B) = \text{tr}(A) + \text{tr}(B)$, $\text{tr}(A) = \text{tr}(A^T)$ and $\text{tr}(BA^T) = \text{tr}(AB^T)$. With the Karush Kuhn Tucker condition, we have the primary feasibility $\mathbf{V} \geq 0, \mathbf{H} \geq 0$ and $\mathbf{G} \geq 0$. We define the Lagrangian as:

$$\mathcal{L}(\mathbf{V},\mathbf{H},\mathbf{G},\Psi,\Phi,\Omega) = f(\mathbf{V},\mathbf{H},\mathbf{G}) + \text{tr}(\Psi\mathbf{V}^T)\,\text{tr}(\Phi\mathbf{H}^T) + \text{tr}(\Omega\mathbf{G}^T) \tag{4.21}$$

Let Ψ, Φ and Ω be Lagrange multiplier matrices and their elements $\psi_{ik'}, \phi_{rk'}$ and $\omega_{pk'}$ are the Lagrange multipliers for constraints $v_{ik'} \geq 0, h_{rk'} \geq 0$ and $g_{pk'} \geq 0$ respectively. The partial derivatives of the objective function in Equation Eq. 4.21 with respect to each variable are:

$$\begin{aligned} \partial\mathcal{L}/\partial\mathbf{V} &= \mu(-2\mathbf{X}^T\mathbf{H} + 2\mathbf{V}\mathbf{H}^T\mathbf{H} + 2\lambda_1\tilde{\mathbf{L}}_\mathbf{X}\mathbf{V}) \\ &\quad - (1-\mu)(2\mathbf{U}\mathbf{G} + 2\mathbf{V}\mathbf{G}^T\mathbf{G}) + \Psi + \gamma_1 ee^T \qquad (4.22) \\ \partial f/\partial\mathbf{H} &= -2\mathbf{X}\mathbf{V} + 2\mathbf{H}\mathbf{V}^T\mathbf{V} + \Phi + \gamma_2 ee^T \qquad (4.23) \\ \partial f/\partial\mathbf{G} &= -2\mathbf{U}^T\mathbf{V} + 2\mathbf{G}\mathbf{V}^T\mathbf{V} + 2\lambda_2\tilde{\mathbf{L}}_\mathbf{U}\mathbf{G} + \Omega + \gamma_3 ee^T \qquad (4.24) \end{aligned}$$

Where vector $e = [1, 1..., 1]^T$. Using the complementary slackness: $\Psi\mathbf{V} = 0, \Phi\mathbf{H} = 0$ and $\Omega\mathbf{G} = 0$, the update equations are derived as follows:

$$\mathbf{V} \leftarrow \mathbf{V} \odot \frac{\mu\mathbf{X}^T\mathbf{H} + (1-\mu)\mathbf{U}\mathbf{G} + \mu\cdot\lambda_1\tilde{\mathbf{W}}_\mathbf{X}\mathbf{V} - \gamma_1 ee^T}{\mathbf{V}(\mu\mathbf{H}^T\mathbf{H} + (1-\mu)\mathbf{G}^T\mathbf{G}) + \mu\cdot\lambda_1\tilde{\mathbf{D}}_\mathbf{V}\mathbf{V}} \tag{4.25}$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{X}\mathbf{V} - \gamma_2 ee^T}{\mathbf{H}\mathbf{V}^T\mathbf{V}} \tag{4.26}$$

$$\mathbf{G} \leftarrow \mathbf{G} \odot \frac{\mathbf{U}^T\mathbf{V} + \lambda_2\tilde{\mathbf{W}}_\mathbf{U}\mathbf{G} - \gamma_3 ee^T}{\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\tilde{\mathbf{D}}_\mathbf{G}\mathbf{G}} \tag{4.27}$$

Obviously, when $\lambda_1, \lambda_2 \to 0$, the above updating rules reduce to the updating rules in origin NMF [94]. The algorithm is summarized in Algorithm 4.2.

Algorithm 4.2: Semi-supervised NMF with Constrain Propagation

Input: Article matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, user preference matrix $\mathbf{U} \in \mathbf{R}^{n \times l}$,weight matrices $\tilde{\mathbf{W}}_\mathbf{X} \in \mathbb{R}^{n \times n}$, $\tilde{\mathbf{W}}_\mathbf{U} \in \mathbb{R}^{l \times l}$, $1 \leq k \leq \min(n, l, m)$, regularization parameters $\lambda_1, \lambda_2, \gamma_1, \gamma_2and\gamma_3$, trade-off parameter μ and convergence controller ϵ

Output: $\mathbf{H} \in \mathbb{R}^{m \times k}$, $\mathbf{V} \in \mathbb{R}^{n \times k}$, $\mathbf{G} \in \mathbf{R}^{l \times k}$

```
Initialize H, V, G by random matrices
Construct weight matrices W̃_X, W̃_U by using Algorithm 4.1
while Eq. 4.18 > ε do
    Fix H and G, update V by Eq. 4.25
    Fix V and G, update H by Eq. 4.26
    Fix H and V, update G by Eq. 4.27
end
```

4.5.3 Convergence Study

Regarding the updating rules above, we have the following theorem:

Theorem 4.1. *The objective function in Eq. 4.18 is non-increasing under updating rules in Eq. 4.25 to Eq. 4.27.*

A convergence proof of standard NMF example can be found in [94]. Thus, we follow the similar procedure starting with the definition of an auxiliary function as following:

Definition 4.1. *$J(x, x')$ is an auxiliary function for $F(x)$ if it satisfies: $J(x, x') \geq F(x)$ and $J(x, x) = F(x)$.*

For an auxiliary function $J(x, x')$ of $F(x)$: we have $F(x)$ is non-increasing under following condition:

$$x^{(t+1)} = \underset{x}{\operatorname{argmin}}\, J(x, x^{(t)}), \tag{4.28}$$

derived from

$$F(x^{(t+1)}) = J(x^{(t+1)}, x^{(t+1)}) \leq J(x^{(t+1)}, x^{(t)}) \leq J(x^{(t)}, x^{(t)}) = F(x^{(t)}).$$

Proof of Theorem 4.1

Under the Definition 4.1, the key step to prove Theorem 4.1 is to find a proper auxiliary function with respect to $\mathbf{V}$, $\mathbf{H}$ and $\mathbf{G}$. We rewrite the objective function in Eq. 5.1 as follow:

$$O = \mu \left(\left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T \right\|_F^2 + \lambda_1 \operatorname{tr}(\mathbf{V}^T \tilde{\mathbf{L}}_{\mathbf{X}} \mathbf{V}) \right) + (1-\mu) \left(\left\| \mathbf{U} - \mathbf{V}\mathbf{G}^T \right\|_F^2 + \lambda_2 \operatorname{tr}(\mathbf{G}^T \tilde{\mathbf{L}}_{\mathbf{U}} \mathbf{G}) \right) + \mathcal{R} \tag{4.29}$$

We use $F_{\mathbf{V}}$, $F_{\mathbf{H}}$ and $F_{\mathbf{G}}$ to denote the part of O which is only relevant to $\mathbf{V}$, $\mathbf{H}$ and $\mathbf{G}$ respectively.

Updating V While fixing $\mathbf{H}$ and $\mathbf{G}$, we can separately minimize O with respect to each element v_{ab} of $\mathbf{V}$. Then, we rewrite the objective function relevant to V as:

$$F_{\mathbf{V}} = \mu \left(\left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T \right\|_F^2 + \lambda_1 \operatorname{tr}(\mathbf{V}^T \tilde{\mathbf{L}}_{\mathbf{X}} \mathbf{V}) \right) + (1-\mu) \left\| \mathbf{U} - \mathbf{V}\mathbf{G}^T \right\|_F^2 + \gamma_1 \left\| \mathbf{V} \right\|_1 \tag{4.30}$$

Lemma 4.1. *Function*

$$\begin{aligned} J(v, v_{ab}^{(t)}) =& F_{\mathbf{V}}(v_{ab}^{(t)}) + F'_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) \\ &+ \frac{\left(\mathbf{V}[\mu \mathbf{H}^T\mathbf{H} + (1-\mu)\mathbf{G}^T\mathbf{G}] + \mu \cdot \lambda_1 \tilde{\mathbf{D}}_{\mathbf{V}} \mathbf{V}\right)_{ab}}{v_{ab}^{(t)}} (v - v_{ab}^{(t)})^2 \end{aligned} \tag{4.31}$$

is an auxiliary function for $F_{\mathbf{V}}$.

Proof. $J(v, v) = F_{\mathbf{V}}(v)$ is trivial. Then we only need to prove $J(v, v_{ab}^{(t)}) \geq F_{\mathbf{V}}(v)$. To do this, we have the Taylor series expansion of $F_{\mathbf{V}}(v)$:

$$F_{\mathbf{V}}(v) = F_{\mathbf{V}}(v_{ab}^{(t)}) + F'_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) + \frac{1}{2} F''_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)})^2. \tag{4.32}$$

For each element v_{ab} in $\mathbf{V}$, it is easy to derive that

$$\begin{aligned} & F'_{\mathbf{V}}(v_{ab}^{(t)}) = \frac{\partial F_{\mathbf{V}}}{\partial v_{ab}} \\ =& \mu[(-2\mathbf{X}^T\mathbf{H} + 2\mathbf{V}\mathbf{H}^T\mathbf{H})_{ab} + (2\lambda_1 \mathbf{L}_{\mathbf{X}}\mathbf{V})_{ab}] + (1-\mu)[(-2\mathbf{U}\mathbf{G})_{ab} + (2\mathbf{V}\mathbf{G}^T\mathbf{G})_{ab}] + \gamma_1 \end{aligned}$$

and

$$F''_{\mathbf{V}}(v_{ab}^{(t)}) = \mu[(2\mathbf{H}^T\mathbf{H})_{bb} + 2\lambda_1\tilde{\mathbf{L}}_{\mathbf{X}aa}] + (1-\mu)(2\mathbf{G}^T\mathbf{G})_{bb}.$$

Thus, by Eq. 4.31 and Eq. 4.32, $J(v, v_{ab}^{(t)}) \geq (F_{\mathbf{V}})_{ab}(v)$ is equivalent to:

$$\frac{\left(\mathbf{V}[\mu\mathbf{H}^T\mathbf{H} + (1-\mu)\mathbf{G}^T\mathbf{G}] + \mu\cdot\lambda_1\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V}\right)_{ab}}{v_{ab}^{(t)}} \geq \frac{1}{2}F''_{\mathbf{V}}(v_{ab}^{(t)})$$

Which is

$$\begin{aligned}&\left(\mathbf{V}[\mu\mathbf{H}^T\mathbf{H} + (1-\mu)\mathbf{G}^T\mathbf{G}] + \mu\cdot\lambda_1\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V}\right)_{ab}\\ &\geq \left(\mu[(\mathbf{H}^T\mathbf{H})_{bb} + \lambda_1\tilde{\mathbf{L}}_{\mathbf{X}aa}] + (1-\mu)(\mathbf{G}^T\mathbf{G})_{bb}\right)\cdot v_{ab}^{(t)}\end{aligned} \tag{4.33}$$

We have:

$$\mu(\mathbf{V}\mathbf{H}^T\mathbf{H})_{ab} = \mu\sum_{i=1}^{k} v_{ai}^{(t)}(\mathbf{H}^T\mathbf{H})_{ib} \geq \mu v_{ab}^{(t)}(\mathbf{H}^T\mathbf{H})_{bb},$$

and

$$(1-\mu)(\mathbf{V}\mathbf{G}^T\mathbf{G})_{ab} \geq (1-\mu)\sum_{p=1}^{l} v_{ap}^{(t)}(\mathbf{G}^T\mathbf{G})_{pb} = (1-\mu)v_{ab}^{(t)}(\mathbf{G}^T\mathbf{G})_{bb},$$

and

$$(\lambda_1\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V})_{ab} = \lambda_1\sum_{j=1}^{n}\tilde{\mathbf{D}}_{\mathbf{V}aj}v_{jb}^{(t)} \geq \lambda_1\tilde{\mathbf{D}}_{\mathbf{V}aa}v_{ab}^{(t)} \geq \mu\lambda_1(\mathbf{D}_{\mathbf{V}} - \tilde{\mathbf{W}}_{\mathbf{X}})_{aa}v_{ab}^{(t)} = \mu\lambda_1\tilde{\mathbf{L}}_{\mathbf{X}aa}v_{ab}^{(t)}.$$

Thus, Eq. 4.33 holds and $J(v, v_{ab}^{(t)}) \geq F_{\mathbf{V}}(v)$. □

Then, We can derive the following updating rule for v_{ab} by replacing auxiliary function in Eq. 4.28 with Eq. 4.31:

$$\begin{aligned}v_{ab}^{(t+1)} &= v_{ab}^{(t)} - v_{ab}^{(t)}\frac{F'_{\mathbf{V}}(v_{ab}^{(t)})}{\left(\mathbf{V}[\mu\mathbf{H}^T\mathbf{H} + (1-\mu)\mathbf{G}^T\mathbf{G}] + \mu\cdot\lambda_1\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V}\right)_{ab}}\\ &= v_{ab}^{(t)}\frac{\left(\mu\mathbf{X}^T\mathbf{H} + (1-\mu)\mathbf{U}\mathbf{G} + \mu\cdot\lambda_1\tilde{\mathbf{W}}_{\mathbf{X}}\mathbf{V} - \gamma_1 e^T e\right)_{ab}}{\left(\mathbf{V}(\mu\mathbf{H}^T\mathbf{H} + (1-\mu)\mathbf{G}^T\mathbf{G}) + \mu\cdot\lambda_1\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V}\right)_{ab}}\end{aligned}$$

Since Eq. 4.31 is an auxiliary function, $F_{\mathbf{V}}$ is non-increasing under this updating rule.

Updating H We focus on updating $\mathbf{H}$ while fixing $\mathbf{V}$ and $\mathbf{G}$. We rewrite the objective function as:

$$F_{\mathbf{H}} = \mu \left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T \right\|_F^2 + \gamma_2 \left\| \mathbf{H} \right\|_1 \tag{4.34}$$

Lemma 4.2. *Function*

$$J(h, h_{ab}^{(t)}) = F_{\mathbf{H}}(h_{ab}^{(t)}) + F'_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)}) + \frac{(\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab}}{h_{ab}^{(t)}}(h - h_{ab}^{(t)})^2 \tag{4.35}$$

is an auxiliary function for $F_{\mathbf{H}}$.

Proof. $J(h, h) = F_{\mathbf{H}}(h)$ is obvious. Then we only need to prove $J(h, h_{ab}^{(t)}) \geq F_{\mathbf{H}}(h)$. Similar to the proof of Lemma 4.1, we first write the Taylor series expansion of $F_{\mathbf{H}}(h)$ as:

$$F_{\mathbf{H}}(h) = F_{\mathbf{H}}(h_{ab}^{(t)}) + F'_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)}) + \frac{1}{2}F''_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)})^2 \tag{4.36}$$

where $F'_{\mathbf{H}}(h_{ab}^{(t)}) = (-2\mathbf{X}\mathbf{V} + 2\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab} + \gamma_2$, and $F''_{\mathbf{H}}(h_{ab}^{(t)}) = 2(\mathbf{V}^T\mathbf{V})_{ab}$. By comparing Eq. 4.35 with Eq. 4.36 we find that $J(h, h_{ab}^{(t)}) \geq F_{\mathbf{H}}(h)$ is equivalent to

$$\frac{(\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab}}{h_{ab}^{(t)}} \geq (\mathbf{V}^T\mathbf{V})_{ab} \tag{4.37}$$

We have $(\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab} = \sum_{i=1}^{k} h_{ai}^{(t)}(\mathbf{V}^T\mathbf{V})_{ib} \geq h_{ab}^{(t)}(\mathbf{V}^T\mathbf{V})_{ab}$. Thus, the matrix inequality in Eq. 4.37 holds and $J(h, h_{ab}^{(t)}) \geq (F_{\mathbf{H}})_{ab}(h)$. □

We can now demonstrate the convergence of Eq. 4.18 under the following updating rule by replacing Eq. 4.28 with $J(h, h_{ab}^{(t)})$ in Eq. 4.35:

$$h_{ab}^{(t+1)} = h_{ab}^{(t)} - h_{ab}^{(t)} \frac{F'_{\mathbf{H}}(h_{ab}^{(t)})}{(\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab}} = h_{ab}^{(t)} \frac{(\mathbf{X}\mathbf{V} - \gamma_2 ee^T)_{ab}}{(\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab}}$$

Since Eq. 4.35 is an auxiliary function, $F_{\mathbf{H}}$ is non-increasing under this updating rule.

Updating G We focus on updating $\mathbf{G}$ while fixing $\mathbf{V}$ and $\mathbf{H}$. We rewrite the objective function relevant to $\mathbf{G}$ as

$$F_{\mathbf{G}} = \left\|\mathbf{U} - \mathbf{V}\mathbf{G}^T\right\|_F^2 + \lambda_2 \operatorname{tr}(\mathbf{G}^T\tilde{\mathbf{L}}_{\mathbf{U}}\mathbf{G}) + \gamma_3 \|\mathbf{G}\|_1 \tag{4.38}$$

Lemma 4.3. *Function*

$$J(g, g_{ab}^{(t)}) = F_{\mathbf{G}}(g_{ab}^{(t)}) + F'_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)}) + \frac{(\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G})_{ab}}{g_{ab}^{(t)}}(g - g_{ab}^{(t)})^2 \tag{4.39}$$

is an auxiliary function for $F_{\mathbf{G}}$.

Proof. $J(g, g) = F_{\mathbf{G}}(g)$ is obvious. Then we only need to prove $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$. We compare the Taylor series expansion of $F_{\mathbf{G}}(g)$ as:

$$F_{\mathbf{G}}(g) = F_{\mathbf{G}}(g_{ab}^{(t)}) + F'_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)}) + \frac{1}{2}F''_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)})^2 \tag{4.40}$$

where $F'_{\mathbf{G}}(g_{ab}^{(t)}) = (-2\mathbf{U}^T\mathbf{V} + 2\lambda_2\tilde{\mathbf{L}}_{\mathbf{U}}\mathbf{G})_{ab} + 2(\mathbf{G}\mathbf{V}^T\mathbf{V})_{ab} + \gamma_3$ and $F''_{\mathbf{G}}(g_{ab}^{(t)}) = 2\lambda_2\tilde{\mathbf{L}}_{\mathbf{U}aa} + 2(\mathbf{V}^T\mathbf{V})_{bb}$. By comparing Eq. 4.39 with Eq. 4.40, we find that $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$ is equivalent to

$$\frac{(\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G})_{ab}}{g_{ab}^{(t)}} \geq \frac{1}{2}(F_{\mathbf{G}})''_{ab} = \lambda_2\tilde{\mathbf{L}}_{\mathbf{U}aa} + (\mathbf{V}^T\mathbf{V})_{bb} \tag{4.41}$$

Since, we have

$$(\mathbf{G}\mathbf{V}^T\mathbf{V})_{ab} = \sum_{i=1}^{k} g_{ai}^{(t)}(\mathbf{V}^T\mathbf{V})_{ib} \geq g_{ab}^{(t)}(\mathbf{V}^T\mathbf{V})_{bb}$$

and

$$(\lambda_2\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G})_{ab} = \lambda_2\sum_{p=1}^{l}\tilde{\mathbf{D}}_{\mathbf{G}ap}g_{pb}^{(t)} \geq \lambda_2\tilde{\mathbf{D}}_{\mathbf{G}aa}g_{ab}^{(t)} \geq \lambda_2(\tilde{\mathbf{D}}_{\mathbf{G}} - \tilde{\mathbf{W}}_{\mathbf{U}})_{aa}g_{ab}^{(t)} = \lambda_2\tilde{\mathbf{L}}_{\mathbf{U}aa}g_{ab}^{(t)}$$

Thus, Eq. 4.41 holds and $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$. □

We can now replace Eq. 4.28 by $J(g, g_{ab}^{(t)})$ in Eq. 4.39 to derive the update rule:

$$g_{ab}^{(t+1)} = g_{ab}^{(t)} - g_{ab}^{(t)} \frac{F'_{\mathbf{G}}(g_{ab}^{(t)})}{(\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G})_{ab}} = g_{ab}^{(t)} \frac{(\mathbf{U}^T\mathbf{V} + \lambda_2\tilde{\mathbf{W}}_{\mathbf{U}}\mathbf{G} - \gamma_3 ee^T)_{ab}}{(\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G})_{ab}}$$

since Eq. 4.39 is an auxiliary function, $F_{\mathbf{G}}$ is non-increasing under this updating rule.

4.5.4 Computational Complexity Analysis

Next, we analyse the computational complexity of our algorithm NMFCP comparing with the origin NMF. Intuitively, we assume the multiplicative update iteration stops at time t, the overall cost of the origin NMF is $O(tMNK)$, where the input data matrix is an M dimensional matrix with N data points and K is set as the number of latent factors. In our case, the multiplicative updates cost for non-negative factorization is $O(tMnK + tnlK)$. Before this, the NMFCP needs $O(n^2M + l^2n)$ to construct the weight matrices for $\mathbf{X}$ and $\mathbf{U}$, respectively. Therefore, the maximum overall cost for NMFCP is $O(MnK + tnlK + n^2M + l^2n)$.

4.6 A Locally Weighted Algorithm

In this section, based on above proposed NMFCP, we propose a locally weighted NMF algorithm (LWNMF) which measures the geography distance between the original data points and the approximate data points on each iteration of optimisation and tunes up the optimising focus for next iteration to better emphasise certain parts of the data matrix. Notice that the local weight proposed in this section is not the same one that described in Section 4.4 reflecting the weak constraint between original data points. We use symbol $\mathbf{Q}$ to denote this local weight matrix. It will be updated in the beginning of the new iteration according to the approximation result of the last iteration. In the following, we will give details about the weight setting function, the incorporated objective function and the computation of LWNMF.

4.6.1 Weight Setting

The geographical distance between the original matrix and approximated matrix is measured by a diagonal weight matrix $\mathbf{Q}$. Intuitively, this distance monotonically decreases along with the iteration of approximation. We use the Gaussian kernel function $w_{ij} = \exp\left(-\frac{1}{2}\left(\frac{d_{ij}}{\sigma}\right)^2\right)$ to update $\mathbf{Q}$ as follow:

$$q_{ii}^{(t+1)} = \exp\left(-\frac{\left\|X_i - X_i^{(t)}\right\|_F^2}{2\sigma^2}\right), \quad \sigma > 0 \tag{4.42}$$

Where $q_{ii}^{(t+1)}$ is the local weight of data point X_i in the iteration $t+1$. $\left\|X_i - X_i^{(t)}\right\|_F^2$ is the geographical distance between the original data point X_i and the approximation point in the iteration t. $exp(\mathbf{.})$ is the exponential function and σ is a free parameter determining the width of Gaussian kernel, which is the scale of the weight q_{ii} under our definition. If X_i and $X_i^{(t)}$ is very close ($\left\|X_i - X_i^{(t)}\right\|_F^2 \approx 0$), q_{ii} will approaches 1; while $X_i^{(t)}$ is far from X_i ($\left\|X_i - X_i^{(t)}\right\|_F^2 \gg 0$), q_{ii} goes to 0. The selection of parameter σ is very crucial for performance and a challenge problem that engaged much attention from research community [24, 178].

4.6.2 The Weighted Objective Function

We know that a bigger value of q_{ii} indicates a better approximation of a data point in one iteration and more weight should be given for the next iteration. With the above definition of locally weight matrix $\mathbf{Q}$, we express the approximation of data points as $X_i \approx \sum_{k=1}^{K} h_k v_{ik} q_{ii}$. Through minimizing the following objective function, we will find the above approximation:

$$\begin{aligned} f(\mathbf{V},\mathbf{H},\mathbf{G}) = & \underset{\mathbf{H},\mathbf{V},\mathbf{G}}{\operatorname{argmin}} \mu\left(\left\|\mathbf{X} - \mathbf{H}\mathbf{V}^T\mathbf{Q}_\mathbf{X}\right\|_F^2 + \lambda_1\phi(\mathbf{V})\right) \\ & + (1-\mu)\left(\left\|\mathbf{U} - \mathbf{V}\mathbf{G}^T\mathbf{Q}_\mathbf{U}\right\|_F^2 + \lambda_2\phi(\mathbf{G})\right) + \mathcal{R} \\ \text{s.t.} \quad & \mathbf{V},\mathbf{H},\mathbf{G} \geq 0 \end{aligned} \tag{4.43}$$

subject to $\mathbf{V} \geq 0$, $\mathbf{H} \geq 0$ and $\mathbf{G} \geq 0$, where $\phi(\mathbf{V}) = \frac{1}{2} \sum_{i,j=1}^{n} \left\| V_i - V_j \right\|^2 \cdot \tilde{W}_{X_{ij}}$ and $\phi(\mathbf{G}) = \frac{1}{2} \sum_{i,j=1}^{l} \left\| G_p - G_q \right\|^2 \cdot \tilde{W}_{U_{pq}}$ are two geometrical regularization term mentioned in Section 4.5. λ_1 and λ_2 are the regularization parameters respectively. Weight matrix $\mathbf{Q_X}$ and $\mathbf{Q_U}$ are initially constructed with identity matrix $\mathbf{I} = diag(1, 1, ..., 1)$ and updated in the beginning of each iteration until Eq. 4.43 approaches the local minima.

4.6.3 Computation and Convergence

The objective function in Eq. 4.43 is also not convex in all the variables together as explained in Subsection 4.5.2. Next, we find its local minima instead of the global minima by iteratively updating one variable while fixing others. We first define its Lagrangian function with Lagrange multipliers Ψ, Φ and Ω as follow:

$$\begin{aligned}\mathcal{L}(\mathbf{V}, \mathbf{H}, \mathbf{G}, \Psi, \Phi, \Omega) =& \mu \left(\left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T\mathbf{Q_X} \right\|_F^2 + \lambda_1 \operatorname{tr}(\mathbf{V}^T \tilde{\mathbf{L}}_\mathbf{X} \mathbf{V}) \right) \\ &+ (1-\mu) \left(\left\| \mathbf{U} - \mathbf{V}\mathbf{G}^T\mathbf{Q_U} \right\|_F^2 + \lambda_2 \operatorname{tr}(\mathbf{G}^T \tilde{\mathbf{L}}_\mathbf{U} \mathbf{G}) \right) \\ &+ \mathcal{R} + \operatorname{tr}(\Psi \mathbf{V}^T) + \operatorname{tr}(\Phi \mathbf{H}^T) + \operatorname{tr}(\Omega \mathbf{G}^T) \end{aligned} \tag{4.44}$$

Computing the partial derivatives of Eq. 4.44 with respect to each variable and using the KKT conditions, we will obtain the minima with setting partial derivatives to zero, $(\partial \mathcal{L} / \partial \mathbf{H})\, h_{rk'} = 0$, $(\partial \mathcal{L} / \partial \mathbf{V})\, v_{ik'} = 0$ and $(\partial \mathcal{L} / \partial \mathbf{G})\, g_{pk'} = 0$. Since the locally weight matrices $\mathbf{Q_X}$ and $\mathbf{Q_U}$ are symmetrical diagonal matrices, denote $\tilde{\mathbf{Q}}_\mathbf{X} = \mathbf{Q_X}\mathbf{Q_X}^T = diag(q_{ii}^2)$ and $\tilde{\mathbf{Q}}_\mathbf{U} = \mathbf{Q_U}\mathbf{Q_U}^T = diag(q_{pp}^2)$. The updating equations are derived as follows:

$$\mathbf{V} \leftarrow \mathbf{V} \odot \frac{\mu(\mathbf{Q_X}\mathbf{X}^T\mathbf{H} + \lambda_1 \tilde{\mathbf{W}}_\mathbf{X}\mathbf{V}) + (1-\mu)\mathbf{U}\mathbf{Q_U}\mathbf{G} - \gamma_1 e^T e}{\mu(\tilde{\mathbf{Q}}_\mathbf{X}\mathbf{V}\mathbf{H}^T\mathbf{H} + \lambda_1 \tilde{\mathbf{D}}_\mathbf{V}\mathbf{V}) + (1-\mu)\mathbf{V}\mathbf{G}^T\tilde{\mathbf{Q}}_\mathbf{U}\mathbf{G}} \tag{4.45}$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{X}\mathbf{Q_X}\mathbf{V} - \gamma_2 e^T e}{\mathbf{H}\mathbf{V}^T\tilde{\mathbf{Q}}_\mathbf{X}\mathbf{V}} \tag{4.46}$$

$$\mathbf{G} \leftarrow \mathbf{G} \odot \frac{\mathbf{Q_U}\mathbf{U}^T\mathbf{V} + \lambda_2 \tilde{\mathbf{W}}_\mathbf{U}\mathbf{G} - \gamma_3 e^T e}{\tilde{\mathbf{Q}}_\mathbf{U}\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2 \tilde{\mathbf{D}}_\mathbf{G}\mathbf{G}} \tag{4.47}$$

Regarding the updating rules above, we have the following theorem:

Theorem 4.2. *The objective function in Equation Eq. 4.43 is non-increasing under updating rules in Eq. 4.47 to Eq. 4.47.*

We give theoretical proof to Theorem 4.2 separately with respect to each variables in the rest of this section.

Proof of Theorem 4.2

Under the Definition 4.1, we should first properly define auxiliary function for $\mathbf{V}$, $\mathbf{H}$ and $\mathbf{G}$ with reference to the objective function in Eq. 4.43. We use $F_{\mathbf{V}}$, $F_{\mathbf{H}}$ and $F_{\mathbf{G}}$ to denote the part of O which is only relevant to $\mathbf{V}$, $\mathbf{H}$ and $\mathbf{G}$ respectively.

Updating V While fixing $\mathbf{H}$ and $\mathbf{G}$, we rewrite $F_{\mathbf{V}}$ as follow:

$$F_{\mathbf{V}} = \left(\mu\left[\left\|\mathbf{X} - \mathbf{H}\mathbf{V}^T\mathbf{Q}_{\mathbf{X}}\right\|_F^2 + \lambda_1 \operatorname{tr}(\mathbf{V}^T\tilde{\mathbf{L}}_{\mathbf{X}}\mathbf{V})\right]) + (1-\mu)\left\|\mathbf{U} - \mathbf{V}\mathbf{G}^T\mathbf{Q}_{\mathbf{U}}\right\|_F^2 + \gamma_1\|\mathbf{V}\|_1\right)_{ab} \tag{4.48}$$

Considering any element v_{ab} in $\mathbf{V}$, we can easily derive that:

$$\begin{aligned} F'_{\mathbf{V}}(v_{ab}^{(t)}) =& \frac{\partial F_{\mathbf{V}}}{\partial v_{ab}} \\ =& \mu[\mathbf{Q}_{\mathbf{X}aa}(-2\mathbf{X}^T\mathbf{H})_{ab} + \tilde{\mathbf{Q}}_{\mathbf{X}aa}^T(2\mathbf{V}\mathbf{H}^T\mathbf{H})_{ab} + (2\lambda_1\mathbf{L}_{\mathbf{X}}\mathbf{V})_{ab}] \\ &+ (1-\mu)[(-2\mathbf{U}\mathbf{Q}_{\mathbf{U}}\mathbf{G})_{ab} + (2\mathbf{V}\mathbf{G}^T\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G})_{ab}] + \gamma_1 \end{aligned} \tag{4.49}$$

and

$$F''_{\mathbf{V}}(v_{ab}^{(t)}) = \mu[2\tilde{\mathbf{Q}}_{\mathbf{X}aa}^T(\mathbf{H}^T\mathbf{H})_{bb} + 2\lambda_1\tilde{\mathbf{L}}_{\mathbf{X}aa}] + (1-\mu)(2\mathbf{G}^T\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G})_{bb}. \tag{4.50}$$

Lemma 4.4. *Function*

$$\begin{aligned} J(v, v_{ab}^{(t)}) =& F_{\mathbf{V}}(v_{ab}^{(t)}) + F'_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) \\ &+ \frac{\left(\mu(\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V}\mathbf{H}^T\mathbf{H} + \lambda_1\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V}) + (1-\mu)\mathbf{V}\mathbf{G}^T\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G}\right)_{ab}}{v_{ab}^{(t)}}(v - v_{ab}^{(t)})^2 \end{aligned} \tag{4.51}$$

is an auxiliary function for $F_{\mathbf{V}}$.

Proof. $J(v,v) = F_{\mathbf{V}}(v)$ is obvious. Then we only need to prove $J(v, v_{ab}^{(t)}) \geq F_{\mathbf{V}}(v)$. To do this, we have the Taylor series expansion of $F_{\mathbf{V}}(v)$:

$$F_{\mathbf{V}}(v) = F_{\mathbf{V}}(v_{ab}^{(t)}) + F'_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) + \frac{1}{2}F''_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)})^2. \tag{4.52}$$

Thus, by Eq. 4.51, Eq. 4.52 and Eq. 4.50, $J(v, v_{ab}^{(t)}) \geq (F_{\mathbf{V}})_{ab}(v)$ is equivalent to:

$$\frac{\left(\mu(\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V}\mathbf{H}^T\mathbf{H} + \lambda_1\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V}) + (1-\mu)\mathbf{V}\mathbf{G}^T\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G}\right)_{ab}}{v_{ab}^{(t)}} \tag{4.53}$$

$$\geq \mu[\tilde{\mathbf{Q}}_{\mathbf{X}aa}^T(\mathbf{H}^T\mathbf{H})_{bb} + \lambda_1\tilde{\mathbf{L}}_{\mathbf{X}aa}] + (1-\mu)(\mathbf{G}^T\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G})_{bb} \tag{4.54}$$

We have:

$$\mu(\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V}\mathbf{H}^T\mathbf{H})_{ab} = \mu\sum_{i=1}^{k} v_{ai}^{(t)}\tilde{\mathbf{Q}}_{\mathbf{X}aa}(\mathbf{H}^T\mathbf{H})_{ib} \geq \mu v_{ab}^{(t)}\tilde{\mathbf{Q}}_{\mathbf{X}aa}(\mathbf{H}^T\mathbf{H})_{bb},$$

and

$$(1-\mu)(\mathbf{V}\mathbf{G}^T\tilde{\mathbf{Q}}_{\mathbf{U}}\,\mathbf{G})_{ab} \geq (1-\mu)\sum_{p=1}^{l} v_{ap}^{(t)}(\mathbf{G}^T\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G})_{pb} = (1-\mu)v_{ab}^{(t)}(\mathbf{G}^T\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G})_{bb},$$

and

$$(\lambda_1\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V})_{ab} = \lambda_1\sum_{j=1}^{n}\tilde{\mathbf{D}}_{\mathbf{V}aj}v_{jb}^{(t)} \geq \lambda_1\tilde{\mathbf{D}}_{\mathbf{V}aa}v_{ab}^{(t)} \geq \mu\lambda_1(\mathbf{D}_{\mathbf{V}} - \tilde{\mathbf{W}}_{\mathbf{X}})_{aa}v_{ab}^{(t)} = \mu\lambda_1\tilde{\mathbf{L}}_{\mathbf{X}aa}v_{ab}^{(t)}.$$

Thus, Eq. 4.54 holds and $J(v, v_{ab}^{(t)}) \geq F_{\mathbf{V}}(v)$. □

Then, We can derive the following updating rule for v_{ab} by replacing auxiliary function in Eq. 4.28 with Eq. 4.51:

$$\begin{aligned} v_{ab}^{(t+1)} &= v_{ab}^{(t)} - v_{ab}^{(t)}\frac{F'_{\mathbf{V}}(v_{ab}^{(t)})}{\left(\mu(\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V}\mathbf{H}^T\mathbf{H} + \lambda_1\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V}) + (1-\mu)\mathbf{V}\mathbf{G}^T\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G}\right)_{ab}} \\ &= v_{ab}^{(t)}\frac{\left(\mu\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{X}^T\mathbf{H} + (1-\mu)\mathbf{U}\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G} + \mu\lambda_1\tilde{\mathbf{W}}_{\mathbf{X}}\mathbf{V} - \gamma_1 e^T e\right)_{ab}}{\left(\mu(\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V}\mathbf{H}^T\mathbf{H} + \lambda_1\tilde{\mathbf{D}}_{\mathbf{V}}\mathbf{V}) + (1-\mu)\mathbf{V}\mathbf{G}^T\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G}\right)_{ab}} \end{aligned}$$

Since Eq. 4.51 is an auxiliary function, $F_{\mathbf{V}}$ is non-increasing under this updating rule.

Updating H We focus on updating $\mathbf{H}$ while fixing $\mathbf{V}$ and $\mathbf{G}$. We rewrite the objective function as:

$$F_{\mathbf{H}} = \mu \left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T\mathbf{Q}_{\mathbf{X}} \right\|_F^2 + \gamma_2 \left\| \mathbf{H} \right\|_1 \tag{4.55}$$

Considering any element h_{ab} in $\mathbf{H}$, it is easy to derive

$$F'_{\mathbf{H}}(h_{ab}^{(t)}) = (-2\mathbf{X}\mathbf{Q}_{\mathbf{X}}\mathbf{V} + 2\mathbf{H}\mathbf{V}^T\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V})_{ab} + \gamma_2 \tag{4.56}$$

and

$$F''_{\mathbf{H}}(h_{ab}^{(t)}) = 2(\mathbf{V}^T\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V})_{ab} \tag{4.57}$$

Lemma 4.5. *Function*

$$J(h, h_{ab}^{(t)}) = F_{\mathbf{H}}(h_{ab}^{(t)}) + F'_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)}) + \frac{(\mathbf{H}\mathbf{V}^T\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V})_{ab}}{h_{ab}^{(t)}}(h - h_{ab}^{(t)})^2 \tag{4.58}$$

is an auxiliary function for $F_{\mathbf{H}}$.

Proof. $J(h, h) = F_{\mathbf{H}}(h)$ is obvious. Then we only need to prove $J(h, h_{ab}^{(t)}) \geq F_{\mathbf{H}}(h)$. Similar to the proof of Lemma 4.4, we first write the Taylor series expansion of $F_{\mathbf{H}}(h)$ as:

$$F_{\mathbf{H}}(h) = F_{\mathbf{H}}(h_{ab}^{(t)}) + F'_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)}) + \frac{1}{2}F''_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)})^2 \tag{4.59}$$

By comparing Eq. 4.35 with Eq. 4.36 and Eq. 4.57, we find that $J(h, h_{ab}^{(t)}) \geq F_{\mathbf{H}}(h)$ is equivalent to

$$\frac{(\mathbf{H}\mathbf{V}^T\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V})_{ab}}{h_{ab}^{(t)}} \geq (\mathbf{V}^T\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V})_{ab} \tag{4.60}$$

Since we have $(\mathbf{H}\mathbf{V}^T\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V})_{ab} = \sum_{i=1}^{k} h_{ai}^{(t)}(\mathbf{V}^T\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V})_{ib} \geq h_{ab}^{(t)}(\mathbf{V}^T\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V})_{ab}$. Thus, the matrix inequality in Eq. 4.37 holds and $J(h, h_{ab}^{(t)}) \geq (F_{\mathbf{H}})_{ab}(h)$. □

We can now demonstrate the convergence of Eq. 4.43 under the following updating

rule by replacing Eq. 4.28 with $J(h, h_{ab}^{(t)})$ in Eq. 4.58:

$$h_{ab}^{(t+1)} = h_{ab}^{(t)} - h_{ab}^{(t)} \frac{F'_{\mathbf{H}}(h_{ab}^{(t)})}{(\mathbf{H}\mathbf{V}^T\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V})_{ab}} = h_{ab}^{(t)} \frac{(\mathbf{X}\mathbf{V}\mathbf{Q}_{\mathbf{X}} - \gamma_2 ee^T)_{ab}}{(\mathbf{H}\mathbf{V}^T\tilde{\mathbf{Q}}_{\mathbf{X}}\mathbf{V})_{ab}}$$

Since Eq. 4.35 is an auxiliary function, $F_{\mathbf{H}}$ is non-increasing under this updating rule.

Updating G We focus on updating $\mathbf{G}$ while fixing $\mathbf{V}$ and $\mathbf{H}$. We rewrite the objective function relevant to $\mathbf{G}$ as

$$F_{\mathbf{G}} = \left\| \mathbf{U} - \mathbf{V}\mathbf{G}^T\mathbf{Q}_{\mathbf{U}} \right\|_F^2 + \lambda_2 \operatorname{tr}(\mathbf{G}^T\tilde{\mathbf{L}}_{\mathbf{U}}\mathbf{G}) + \gamma_3 \|\mathbf{G}\|_1 \tag{4.61}$$

Lemma 4.6. *Function*

$$J(g, g_{ab}^{(t)}) = F_{\mathbf{G}}(g_{ab}^{(t)}) + F'_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)}) + \frac{(\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G})_{ab}}{g_{ab}^{(t)}}(g - g_{ab}^{(t)})^2 \tag{4.62}$$

is an auxiliary function for $F_{\mathbf{G}}$.

Proof. $J(g, g) = F_{\mathbf{G}}(g)$ is obvious. Then we only need to prove $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$. We compare the Taylor series expansion of $F_{\mathbf{G}}(g)$ as:

$$F_{\mathbf{G}}(g) = F_{\mathbf{G}}(g_{ab}^{(t)}) + F'_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)}) + \frac{1}{2}F''_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)})^2 \tag{4.63}$$

where $F'_{\mathbf{G}}(g_{ab}^{(t)}) = (-2\mathbf{Q}_{\mathbf{U}aa}\mathbf{U}^T\mathbf{V} + 2\lambda_2\tilde{\mathbf{L}}_{\mathbf{U}}\mathbf{G})_{ab} + 2\mathbf{Q}_{\mathbf{U}aa}(\mathbf{G}\mathbf{V}^T\mathbf{V})_{ab} + \gamma_3$ and $F''_{\mathbf{G}}(g_{ab}^{(t)}) = 2\lambda_2\tilde{\mathbf{L}}_{\mathbf{U}aa} + 2\tilde{\mathbf{Q}}_{\mathbf{U}aa}(\mathbf{V}^T\mathbf{V})_{bb}$. By comparing Eq. 4.39 with Eq. 4.40, we find that $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$ is equivalent to

$$\frac{(\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G})_{ab}}{g_{ab}^{(t)}} \geq \frac{1}{2}(F_{\mathbf{G}})''_{ab} = \lambda_2\tilde{\mathbf{L}}_{\mathbf{U}aa} + \tilde{\mathbf{Q}}_{\mathbf{U}aa}(\mathbf{V}^T\mathbf{V})_{bb} \tag{4.64}$$

Since, we have

$$(\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G}\mathbf{V}^T\mathbf{V})_{ab} = \mathbf{Q}_{\mathbf{U}aa}\sum_{i=1}^{k} g_{ai}^{(t)}(\mathbf{V}^T\mathbf{V})_{ib} \geq g_{ab}^{(t)}\tilde{\mathbf{Q}}_{\mathbf{U}aa}(\mathbf{V}^T\mathbf{V})_{bb}$$

and

$$(\lambda_2\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G})_{ab} = \lambda_2 \sum_{p=1}^{l} \tilde{\mathbf{D}}_{\mathbf{G}ap} g_{pb}^{(t)} \geq \lambda_2 \tilde{\mathbf{D}}_{\mathbf{G}aa} g_{ab}^{(t)} \geq \lambda_2 (\tilde{\mathbf{D}}_{\mathbf{G}} - \tilde{\mathbf{W}}_{\mathbf{U}})_{aa} g_{ab}^{(t)} = \lambda_2 \tilde{\mathbf{L}}_{\mathbf{U}aa} g_{ab}^{(t)}$$

Thus, Eq. 4.64 holds and $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$. □

We can now replace Eq. 4.28 by $J(g, g_{ab}^{(t)})$ in Eq. 4.62 to derive the update rule:

$$g_{ab}^{(t+1)} = g_{ab}^{(t)} - g_{ab}^{(t)} \frac{F'_{\mathbf{G}}(g_{ab}^{(t)})}{(\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G})_{ab}} = g_{ab}^{(t)} \frac{(\mathbf{Q}_{\mathbf{U}}\mathbf{U}^T\mathbf{V} + \lambda_2\tilde{\mathbf{W}}_{\mathbf{U}}\mathbf{G} - \gamma_3 ee^T)_{ab}}{(\tilde{\mathbf{Q}}_{\mathbf{U}}\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\tilde{\mathbf{D}}_{\mathbf{G}}\mathbf{G})_{ab}}$$

since Eq. 4.62 is an auxiliary function, $F_{\mathbf{G}}$ is non-increasing under this updating rule.

4.7 Experiment and Evaluation

We conduct the experiments focusing on two tasks: detecting the on-going topics and clustering the documents to corresponding topic. The topics are represented by each column of H, while the document clustering results are obtained by an extra k-means cluster algorithm on matrix V.

Four evaluation metrics are employed from these two perspectives, which are Normalized Discounted Cumulative Gain, Mean Average Precision, Accuracy and Normalized Mutual Information. We compare our proposed method with five other NMF-related algorithms, i.e., NMF [94], GNMF [24], JPP [168], LETCS [81] and CMF [149] on two types data sets.

4.7.1 Compared Algorithms

- NMF [94]: Standard NMF method implemented with multiplicative updating rules and F-norm formulation.
- GNMF [24]: Geometric information is utilized as a p-nearest neighbour graph extracted from the original data set. We search 5 nearest neighbours of each data sample to build the Laplacian graph and set the regularisation parameters λ to 10^3 by the author.

- JPP [168]: A time-based model jointly decomposes the past and the present textual matrix with a transition matrix to simulate evolution status between two consecutive time steps. We its prarmeter used to balances the present and the past information to 10^7 by the author.
- CMF [149]: A method associates the factors involving in different relations together with a generalised-linear link function.
- LETCS [81]: A collective matrix factorization method models topic evolution extended from JPP by adding social context matrix and its transition matrix. The parameteres used to balance the present and the past information are also set to 10^7.

4.7.2 Datasets

To evaluate the effectiveness of introducing the social content information as collective NMF and applying constraint propagated weight on unsupervised NMF, we select two types of data sets. The first one is provided by [81] consisting of all the articles published by 80 international news sources in a period of 14 days in April, 2013 and a list of all tweets which link to each articles within 12 hours after the corresponding article's publication. The hashtags (#) quoted by tweets were treated as ground truth topics of the documents which were associated with those tweets and the links between tweets and articles were used to construct the social context matrix. In our experiments, we selected two categories hashtags 5 topics which are defined as: *1) Content-stable hashtags* are those that did not evolve too much in terms of text content, but keep attracting varied attention during the period of collecting; *2) Community-stable hashtags* are relatively stable for their community, but the real events they referring to vary a lot. For example, *#WolrdCup* and *#GRAMMYs*. We will abbreviate them to TS and CS in the following experiments. The second is a serious of semi-synthetic data sets generated from NIST Topic Detection and Tracking (TDT2)[1] text corpus. The full TDT2 corpus is composed of data collected from 6 news sources during 180 days of 1998. Here, we use a short version provided by Cai[2], keeping documents labelled with only one topic. We chose the given topic number

[1] http://www.itl.nist.gov/iad/mig/tests/tdt/1998/
[2]http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

5,7,10,15,20 and 25 and for each of them, generated 20 random non-repeat datasets to tests with algorithms NMF, GNMF and our NMFCP. Since the continuous streaming feature does not exist on generated synthetic datasets, JPP and LETCS are not suitable and without social context domain, CMF reduced to NMF. Tab. 4.1 shows the results of TDT2 where values in bold are significant improvements using the paired Wilcoxon signed rank test with $p \leq 0.05$.

4.7.3 Evaluation Metrics

To evaluate the algorithms' capability of detecting the on-going topics, we use Normalized Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP) as metrics. We use the top 10 and ranking words as the relevant words to express the ground truth topics as well as the topics obtained by the algorithms, mapping the latter to the former with the cosine similarity and setting the ground truth relevance values as binary values. For topics clustering performance, we use Accuracy and Mutual Information with a k-means clustering algorithm applying on the returned documents-words matrix to compare with ground truth topic clusters. The ground truth topic of ith article $\mathbf{V}_{true}(i)$ is extracted from hashtags appearing in the associated tweets and the ground truth topics-words distribution $\mathbf{H}_{true}$ is calculated by $\mathbf{H}_{true} = \mathbf{X}\mathbf{V}_{true}$.

*Normalized discounted cumulative gain (*NDCG*)*: Similar to the assumption of highly relevant documents, we have the following assumption:

Assumption 4.1. *Highly relevant words are more useful when ranking higher in the topics-words distribution.*

Assumption 4.2. *The relevance scores of relevant words are set as equivalent value 1.*

The NDCG is defined as:

$$NDCG(\mathbf{H}, \mathbf{H}_{true}) = (\sum_{i=1}^{k} \frac{DCG_k}{IDCG_k})/k$$

Here, k is the number of latent topics. $DCG_k = \sum_{j=1}^{r} \frac{rel_j}{\log 2(j+1)}$ is the discounted cumulative gain of the kth topic. rel_j is the relevant score of jth relevant word and r is the number of relevant words for the kth topic detected by algorithms. $IDCG_k = \sum_{j=1}^{R} \frac{rel_j}{\log 2(j+1)}$is the

ground truth DCG (Ideal DCG). R is the number of relevant words in ground truth. By the top 10 ranking word representing method, we have $r \leq R = 10$. A higher NDCG score indicates a closer approximation to the ground truth.

*Mean average precision (*MAP*)* calculate the mean of average precision (AP) for all the topics which reflect the algorithms global performance.

$$AP_i = (\sum_{j=1}^{R} \frac{j}{rank_j})/R$$

, where $R = 10$ is the number of relevant words for the topic in ground truth and $rank_j$ is the sequence number of relevant word j in retrieved words list. $\frac{j}{rank_j} = 0$ if relevant word j is not retrieved by algorithms in topic i. The formula of mean average precision is shown as follows:

$$MAP = (\sum_{i=1}^{k} AP_i)/k.$$

The metrics Accuracy and Normalized Mutual Information is used to evaluate the clustering performance of documents. We compare the documents-words distribution matrix $\mathbf{V}$ obtained by algorithms to the real distribution, ground truth $\mathbf{V}_{true}$.

Before testing the clustering performance, a k-means clustering algorithm is applied to each data point in $\mathbf{V}$ (that is, the article $d_i \in n$) to get the exact cluster label l_i of d_i. The ground truth label is l_gnd given by $\mathbf{V}_{true}$. To some extent, the results would depend on the clustering results.

*Accuracy (*AC*)* simply measures the percentage of correct labels for all the articles. It defines as:

$$AC = \frac{\sum_{i=1}^{n} \delta(l_i, l\_gnd_i)}{n}$$

$\delta(l_i, l\_gnd_i)$ is the delta function that equals to 1 if $l_i = l\_gnd_i$ and equals to 0 otherwise. n is the total number of data points.

*Normalized Mutual information (*NMI*)*: Mutual information of two variables X and Y describes the mutual dependence between the two variables, for example in our case, the cluster labels l of data points in $\mathbf{V}$ and the ground truth cluster labels l_gnd of $\mathbf{V}_{true}$. More specifically, it is the average reduction of the uncertainty about the label of a data point in $\mathbf{V}$ when knowing its ground truth label in $\mathbf{V}_{true}$. Normally, the uncertainty of a set of clusters $C = \{c_1, c_2, ..., c_k\}$corresponding to X is represented by entropy as $H(X) =$

$-\sum_{c_i \in C} P(c_i)\log P(c_i) = -\sum_{c_i \in C} \frac{|c_i|}{n}\log\frac{|c_i|}{n}$. Analogously, a set of cluster $C' = \{c'_1, c'_2, ..., c'_\ell\}$ of another variable Y has the similar entropy $H(Y)$ to quantity its uncertainty. Sometimes they are also written as $H(C)$ and $H(C')$. Intuitively, the value of the mutual information of these two variables $MI(X,Y) \in [0, \min(H(X), H(Y))]$ with the following formula:

$$\begin{aligned} MI(X,Y) &= \sum_k \sum_\ell P(c_i, c'_j)\log_2 \frac{P(i,j)}{P(c_i)P(c'_j)} \\ &= \sum_k \sum_\ell \frac{\left|c_i \bigcap c'_j\right|}{n}\log_2 \frac{n\left|c_i \bigcap c'_j\right|}{|c_i|\left|c'_j\right|} \end{aligned}$$

Where $P(c_i) = \frac{|c_i|}{n}$ or $P(c'_j) = \frac{|c'_j|}{n}$ denote the probabilities that a data point belongs to cluster c_i in C or c'_j in C'. $P(i,j) = \frac{|c_i \bigcap c'_j|}{n}$ denotes the joint probability distribution that a data point belongs to cluster c_i in C and to c'_j in C'. As $MI(X,Y)$ is bonded by their entropies, rather than a constant value, it is necessary to use normalized mutual information for easy comparison between the results of different variable pairs.

Table 4.1: Detection and Clustering Performance

#Topic (k)		5	7	10	15	20	25
NDCG	NMF	0.746	0.716	0.636	**0.571**	0.54	0.472
	GNMF	0.777	0.731	**0.676**	0.584	0.518	**0.493**
	NMFCP	**0.806**	**0.769**	**0.679**	**0.592**	**0.556**	**0.495**
MAP	NMF	0.765	0.725	0.633	**0.571**	0.506	0.423
	GNMF	0.766	0.65	0.581	0.509	0.437	412
	NMFCP	**0.816**	**0.773**	**0.673**	0.561	**0.512**	**0.429**
AC	NMF	0.864	0.876	0.819	0.724	0.665	0.659
	GNMF	0.933	0.739	0.79	0.762	0.737	**0.729**
	NMFCP	**0.956**	**0.945**	**0.906**	**0.821**	**0.768**	**0.728**
NMI	NMF	0.772	0.805	0.806	0.745	0.718	0.720
	GNMF	0.800	0.553	0.71	0.728	0.729	**0.763**
	NMFCP	**0.837**	**0.851**	**0.841**	**0.785**	**0.767**	0.756

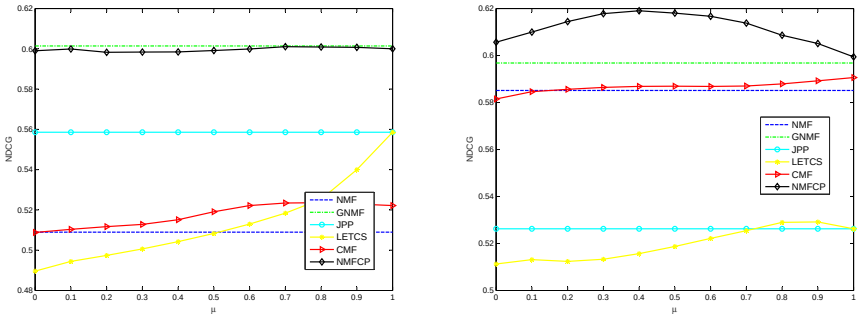
4.7.4 Detection Results of NMFCP

The detection results are examined by the matrices *Normalized discounted cumulative gain* and *Mean average precision*. We set $\delta = 0.2$ and regularization parameters $\lambda_1 = \lambda_2 =$

10^3 for our NMFCP algorithm. Similar, we set regularisation parameter $\lambda = 10^3$ and neighbour $p = 5$ for geometric structure of GNMF. The left two lists in Tab. 4.1 shows the detection results on TDT2 data set. Our proposed method almost outperforms other two algorithms on all NDCG and MAP value. The comparisons with NMF and GNMF indicate that the constraint propagation on the potential links of data points is effect, which will be discussed in the later section. What is interesting is that, the origin NMF algorithm performs well for some case, even better than GNMF. This is consistent with some of the experiments results in [102, 171] with image datasets. The figures below also show that NMFCP algorithm can get better results than others in most cases varying parameters, especially when facing the complicated social context(CS dataset). However, the detection results on TS dataset (e.g. Figs. 4.3a and 4.4a) are always unsatisfactory comparing to GNMF, which is probably due to the interference caused by social context part.

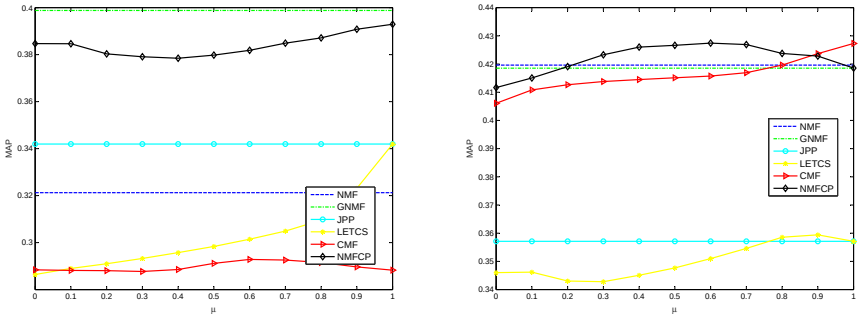
4.7.5 Clustering Results of NMFCP

The detection results are examined by the matrices AC and NMI. From Tab. 4.1, we can see our NMFCP performs almost as well, dramatically better than NMF and GNMF when number of topics below 15. When topic number increases, the accuracy and normalized mutual information decrease on all algorithms as a common trend. On CS dataset, LETCS obtained outstanding clustering results (e.g., Figs. 4.5b, 4.6b, 4.9b, 4.10b, 4.13b and 4.13b) because it treats the whole set of users as a feature set of the documents, which is equivalent of clustering documents with users' distribution. However, as we mentioned ahead, this is unpractical for real application, data streams in our daily social network in particular. Besides LETCS, algorithms involving pairwise links between data points (i.e. GNMF and NMFCP) outperforms others in almost all cases. The performance varies with parameters will be discussed in the following section. The interesting thing is that LETCS detection performance is relatively low on CS dataset (e.g., Figs. 4.3, 4.4, 4.7a and 4.8a), on the contrary, the corresponding clustering results are surprisingly good (e.g.,Figs. 4.5b, 4.6b, 4.9b and 4.10b).



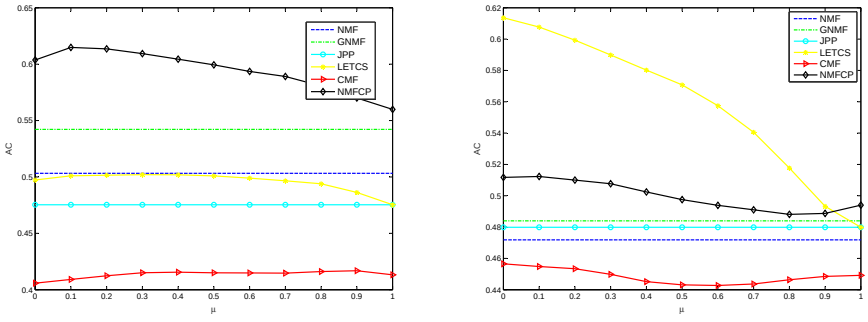
(a) TS (b) CS

Figure 4.3: The NDCG performance versus parameter μ



(a) TS (b) CS

Figure 4.4: The MAP performance versus parameter μ

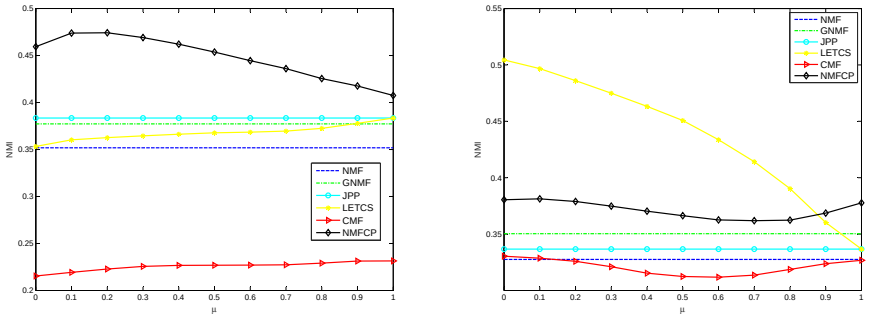


(a) TS (b) CS

Figure 4.5: The AC performance versus parameter μ

4.7.6 Parameters Discussion for NMFCP

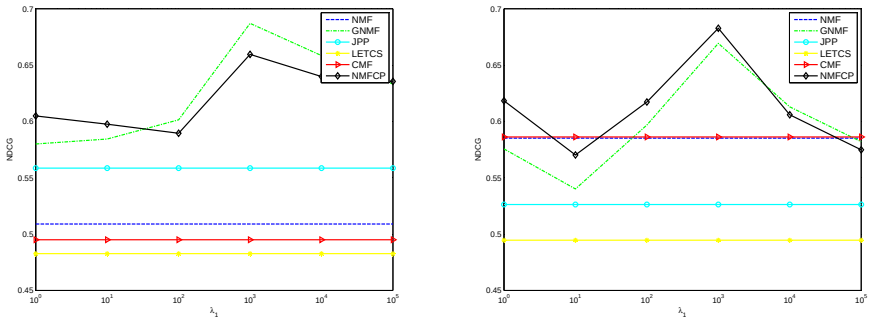
Our NMFCP algorithm has three basic parameters. The trade-off parameter μ determines the relatively importance of collective domains. Therefore, it only works in col-



(a) TS (b) CS

Figure 4.6: The NMI performance versus parameter μ

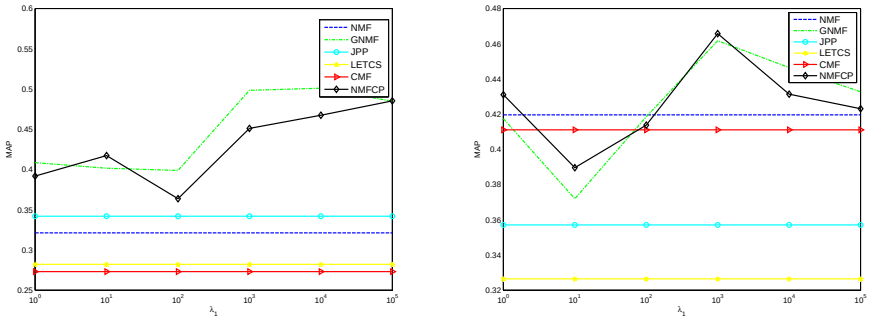
lective methods LETCS, CMF and NMFCP. The regularisation parameter λ_1 and λ_2 controls to what extent the supervised information works, corresponding to matrix $\mathbf{X}$ and matrix $\mathbf{U}$ in shown in Eq. 4.18. Figs. 4.3 to 4.14 show the average performance of algorithms during the period of 14 days with varying μ, λ_1 and λ_2, respectively.



(a) TS (b) CS

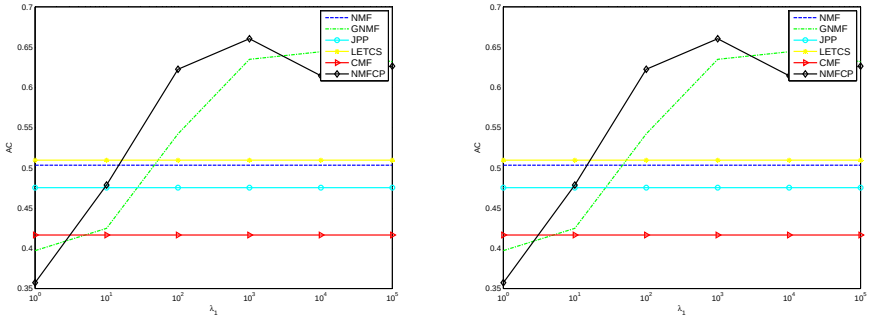
Figure 4.7: The NDCG performance versus parameter λ_1

As we can see in Figs. 4.3 to 4.6, μ influences the results of LETCS, CMF and NMFCP, even in the content-stable dataset(TS), the social context information helps promote the performance. LETCS is affected most obviously since it is highly related to the social information as features. What is unsatisfactory is that the changing trends of detection and clustering performance on TS and CS datasets are inconsistent. For example, the best detection performance on CS was achieved for (Figs. 4.3b and 4.4b), but from Figs. 4.5b and 4.6b we can find that the performance degrade in that range. Nevertheless, this inconsistent between detection and clustering results are not unique, the other two



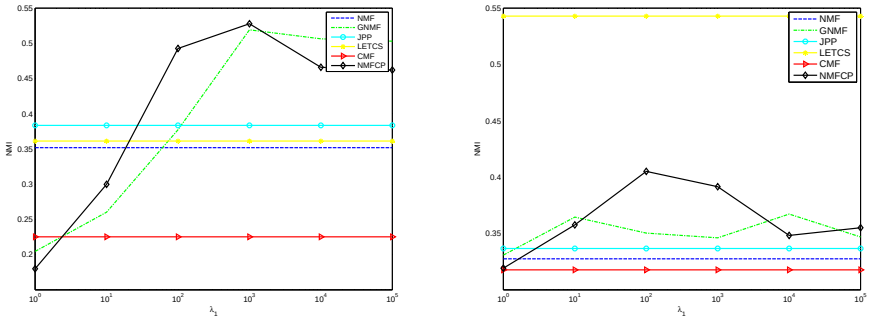
(a) TS (b) CS

Figure 4.8: The MAP performance versus parameter λ_1



(a) TS (b) CS

Figure 4.9: The AC performance versus parameter λ_1

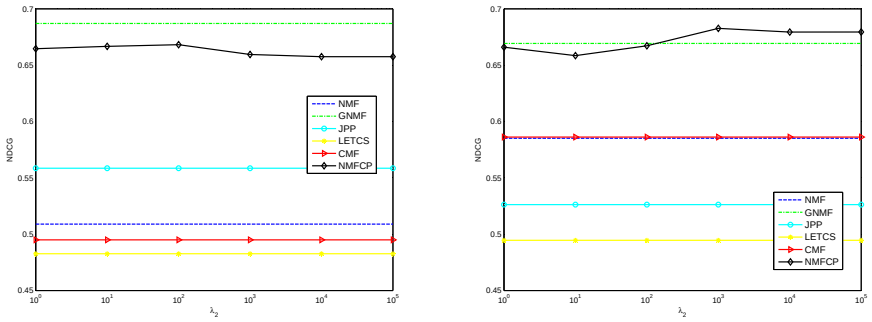


(a) TS (b) CS

Figure 4.10: The NMI performance versus parameter λ_1

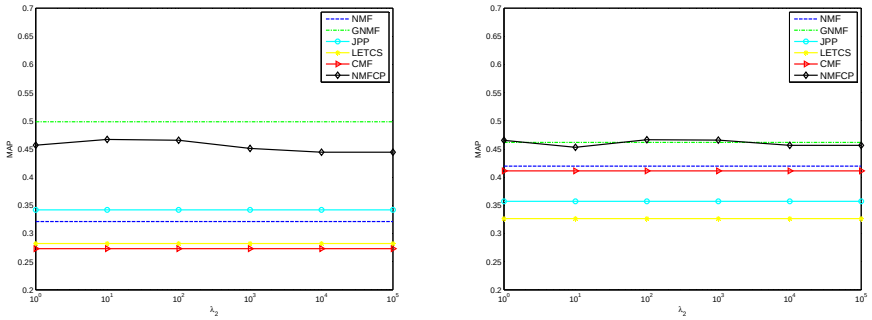
collective methods (LETCS and CMF) show the similar contradiction to some extent. It may be because that the two tasks are separate problems. We evaluated the performance of topic detection using matrix H, while obtained clusters of documents through an extra

k-means clustering algorithm on matrix V. However, this will remain to be one of our future work.



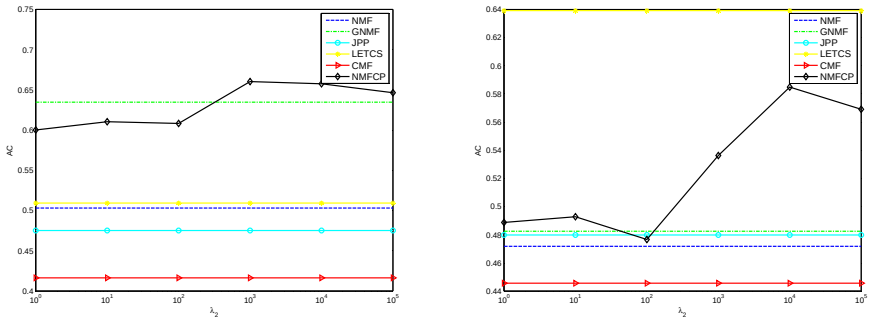
(a) TS (b) CS

Figure 4.11: The NDCG performance versus parameter λ_2



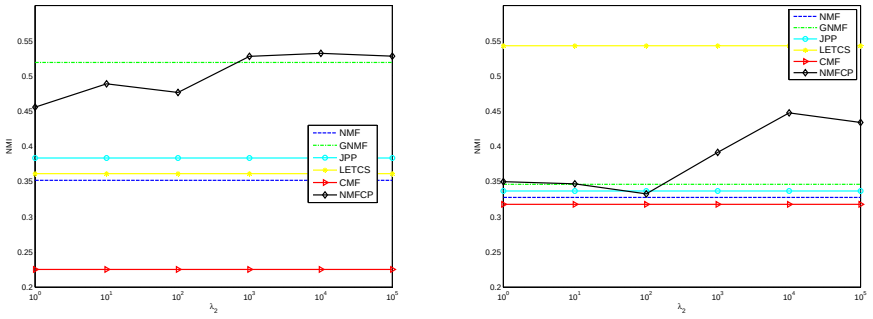
(a) TS (b) CS

Figure 4.12: The MAP performance versus parameter λ_2



(a) TS (b) CS

Figure 4.13: The AC performance versus parameter λ_2



(a) TS (b) CS

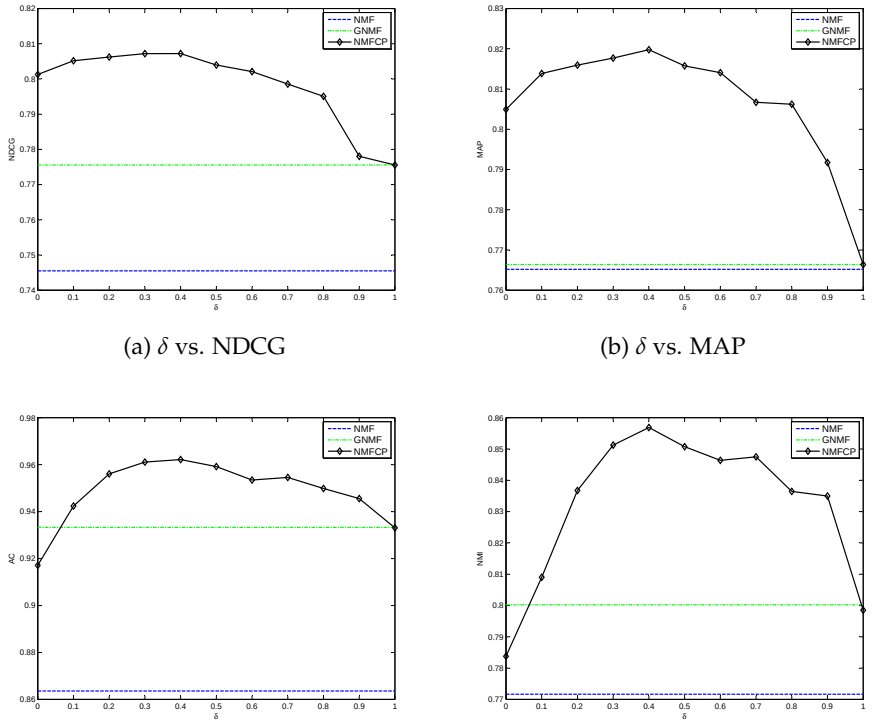
Figure 4.14: The NMI performance versus parameter λ_2

Figs. 4.7 to 4.10 show the performance of NMFCP versus parameterλ_1, which controls how much the supervised information of text content would contributes. It seems that both detection and clustering performances of NMFCP are relatively outstanding when λ_1 is in the range of $\left[10^2, 10^3\right]$ and tends to be decrease after reached the peak, except the clustering performance on CS dataset. But comparing to others, the results of $\lambda_1 = 10^3$ are still acceptable. So the value of λ_1 can be set to 10^3.

Figs. 4.11 to 4.14 show the performance of NMFCP versus parameterλ_2, which is unique to NMFCP, controlling how much the supervised information of social context would contributes. It can be seen that the detection performance of NMFCP is stable with respect to its clustering performance, especially on the TS dataset, that indicates λ_2 affects detection performance slightly. The clustering performance of NMFCP improved as λ_2 increases in the range of $\left[10^2, 10^3\right]$ and keep relatively stable after λ_2 reaches 10^3. Therefore, we can also set λ_2 to 10^3.

4.7.7 Constraint Propagation Effect

The effect of constraint propagation is evaluated on TDT2 dataset by varying the parameter δ. When $\delta = 1$, the soft constraints are not propagated among data points and NMFCP almost reduced to GNMF without social context information; while $\delta = 0$, the pairwise soft constraints among data points are fully propagated among data points vertically and horizontally. As shown in **??**, when $\delta = 1$, the performance of NMFCP get close to GNMF. We observe that the best detection and clustering performances were



(a) δ vs. NDCG

(b) δ vs. MAP

(c) δ vs. AC

(d) δ vs. NMI

Figure 4.15: Performance versus propagation parameter δ on TDT2

achieved for $\delta \in [0.2, 0.6]$ and drop rapidly when δ approaches 1. The results also indicate that constraint propagation is of great help.

4.7.8 Detection Performance of LWNMF

We first show the detection evaluation results on TDT2 datasets with Tab. 4.2. For each given topic number k, we generated 20 random non-repeat datasets to conduct evaluations with algorithms NMF, GNMF, NMFCP and LWNMF. For topic number $k = 7, 8, 9, 10$ and 15, the LWNMF significantly outperforms other comparing algorithms according to the paired Wilcoxon signed rank test with $p \leq 0.05$. As a matter of fact, for $k = 5$ and 20 topics, there are no significant differences in the detection performances between NMFCP and LWNMF algorithms.

We also evaluate the algorithm on real datasets. Except CS and TS mentioned in Subsection 4.7.2, we involve mixed-stable hashtags (MS) which are defined for those topics that are normally community-stable and content-stable. We can find in Tab. 4.16, NM-

Table 4.2: Detection Performance of LWNMF

#Topic	NDCG				MAP			
(k)	NMF	GNMF	NMFCP	LWNMF	NMF	GNMF	NMFCP	LWNMF
5	0.746	0.776	**0.811**	**0.812**	0.765	0.766	**0.826**	**0.830**
6	0.702	0.771	0.771	0.774	0.712	0.808	0.804	0.806
7	0.685	0.769	0.775	**0.780**	0.687	0.786	0.795	**0.800**
8	0.695	0.725	0.725	**0.733**	0.704	0.734	0.739	**0.749**
9	0.637	0.689	0.689	**0.698**	0.633	0.691	0.694	**0.707**
10	0.652	0.687	0.693	**0.701**	0.649	0.661	0.694	**0.705**
15	0.571	0.578	0.581	**0.593**	0.575	0.525	0.558	**0.577**
20	0.537	0.531	**0.556**	**0.558**	0.508	0.484	**0.532**	**0.532**

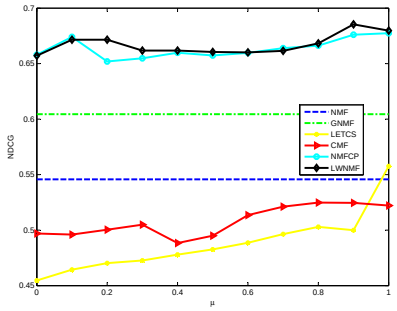
FCP and LWNMF outperform others in most cases, except on MS, where LETCS shows special good performance; however, this is a single case. As we can see, the performance of LWNMF is relatively stable comparing to NMFCP, especially in TS and MS data set, which indicates that the smoothness improved by local weight.

4.7.9 Parameter Analysis for LWNMF

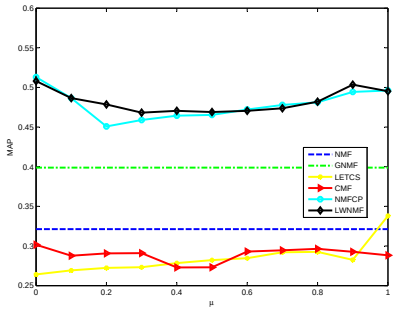
We focus on discussing two of our essential parameters: the trade-off parameter μ and the bandwidth parameter σ of the weight scale. Fig. 4.16 and Fig. 4.17 show the performance of our method varies with the parameters μ and σ, respectively.

When vary μ, we empirically set the regularization parameters $\lambda_1 = 10^3$ and $\lambda_2 = 10^0$, constraint propagation parameter $\delta = 0.2$. And remember, only collective matrix factorization based methods varies with respect to μ, while others remain unchanged. With the value of μ increasing in TS from 0, the curve dropped slightly maybe because of the interference from the social context. As more proportion moves back to text content where $\mu \geq 0,5$, the performance rose correspondingly. The experience results also imply that CS data set is difficult to detect since the performances of all algorithms are relatively low. But the social context helps as the curve continuously increasing when μ goes larger. We also notice that LETCS performs extraordinary well oin MS when $\mu \leq 0.9$ as shown in Fig. 4.16e and Fig. 4.16f which is not consistent with the performances of LETCS on other datasets. By contrast, the performances of LWNMF and NMFCP increase reasonably.

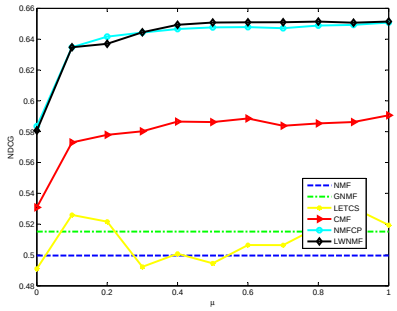
The parameter σ determines the width of the Gaussian kernel. In statistics, it is the standard deviation. In our case, we consider the Gaussian kernel as a weighting function



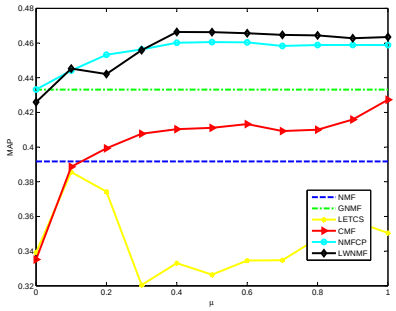
(a) μ vs. NDCG on TS



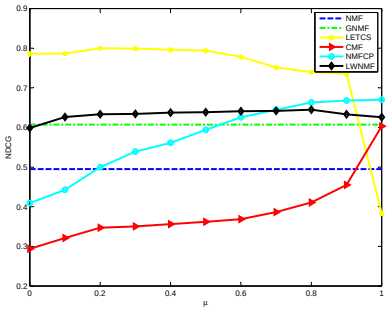
(b) μ vs. MAP on TS



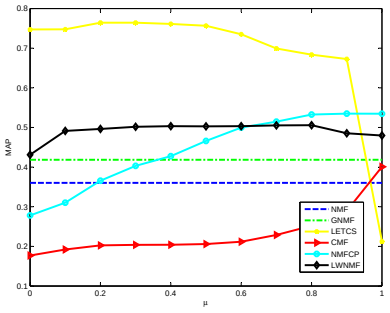
(c) μ vs. NDCG on CS



(d) μ vs. MAP on CS



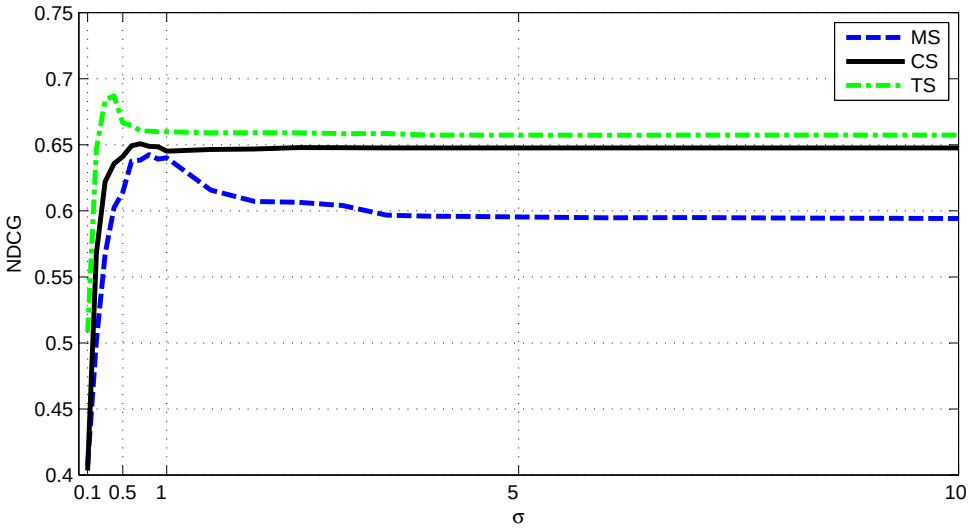
(e) μ vs. NDCG on MS



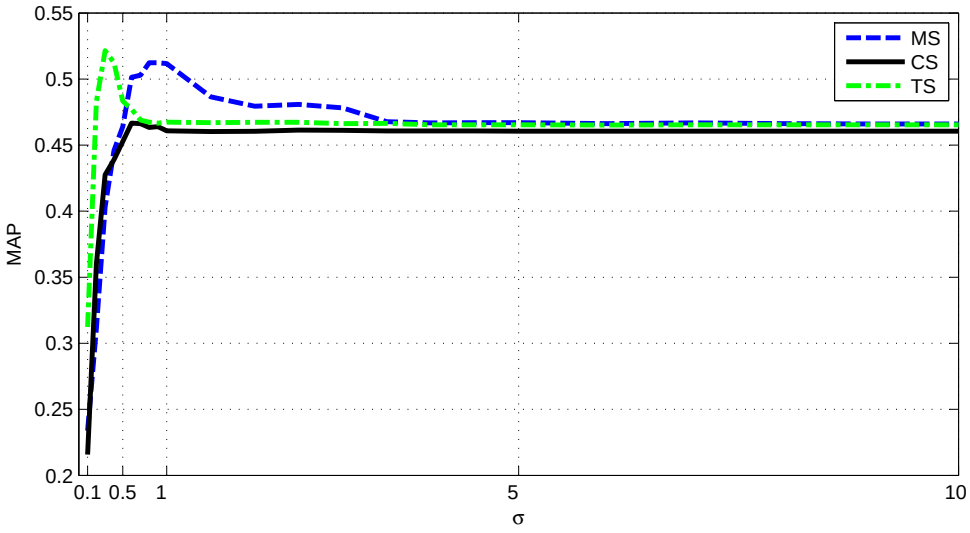
(f) μ vs. MAP on MS

Figure 4.16: Performances of LWNMF versus trade-off parameter μ

and refer to σ as a weight scale which subject to $\sigma \geq 0$. Therefore, the proper value of σ is very crucial to the performance. From Fig. 4.17, we can see a dramatically raise of performances before approaches 0.5 in each data set and curves remain relatively flat after a slight decrease in the range of $[0.5, 1]$. Apparently, the best performance of different dataset is achieved at different σ. The highest performance of TS appears in $0.3 \leq \sigma \leq 0.4$, while for MS, the summit appears in $0.6 \leq \sigma \leq 0.9$, and for CS, it shows in the range of $[0.6, 0.7]$.



(a) NDCG vs. σ



(b) MAP vs. σ

Figure 4.17: Performances of LWNMF versus bandwidth parameter σ

4.8 Summary

In this paper, we present a semi-supervised collective non-negative factorization method for topic detection and tracking, which leverage not only the basic text content, but also the concomitant social context as incidental information to face the ever-changing social network and varied vocabulary challenge. The model firstly applies constraint propagation technique to reveal the interrelations among the data points in each domain and the

propagated constraints are incorporated as regularisation term to help optimize the NMF objective function. The experimental results demonstrate that the propagation can improve the topic detection and document clustering performance effectively. Our methods have been conducted on datasets with social context and outperform others in most cases. Furthermore, we propose a locally weighted matrix factorization (LWNMF) method on both textual content and social context matrices to obtain reliable approximation. The improvement on stability of NMFCP is also validated by the experiments.

Chapter 5

Robust Hierarchical Ensemble Learning for Adaptive Text Mining

Topic detection finds the meaningfulness word representations over the corpus; while its associative task, document clustering assigns documents into the same number of clusters such that the documents in the same cluster share highly similar topics. To remove repercussions caused by outliers in the input documents corpus and overcome other nature properties of online text data, the semantic diversity and volatility of some words as well as the volatility of ongoing latent topics for example, we address the twin problem in this chapter by proposing a novel ensemble framework using Non-negative Matrix Factorization (NMF) with an orthonormal constraint for topic detection through hierarchical document clustering, integrating collective NMF that involves more than one relational matrix collected from online social network and semi-supervised NMF that consider and enhance weak constraints extracted from original data space. To better adapt to the data distribution, we use a dynamic cluster number k to build a flexible k-ary tree for hierarchy. To investigate the robustness of NMF, we use both $\ell_{2,1}$-norm and F-norm objective function with two optimization methods, augmented Lagrangian multiplier and Multiplicative updating rules, deducing RHE series methods (RHEs) on one hand, and exclude outliers obtained through the construction of hierarchy, on the other hand. Through extensive experiments on real datasets and semi-synthetic datasets, RHEs exhibit the robust and remarkable performance for both topic detection and document clustering. Moreover, we comparatively analyse the differences brought by $\ell_{2,1}$-norm and F-norm objective functions based on our experimental results.

5.1 Introduction

TOPIC detection finds the given number of word representations over the coming document corpus such that documents can be interpreted in great depth and meaningfulness, which has been investigated and discussed for many years in the text mining

area [130, 131, 157]. Its associative task, document clustering [24, 129], divides the documents into the given number of clusters such that the documents in the same cluster tell about highly similar topics.

Nonnegative matrix factorization (NMF) has been widely used in tasks of text mining because of its high interpretability and comprehensibility [10, 81, 139]. However, it has a vital challenge to avoid outlier issues for high dimensional data and noises in real applications. The outliers and noises can hardly be thoroughly removed through data pre-process, so that problems they caused have been batted around for ages and concerned in statistic machine learning methods [**?**, 77, 192].

Besides the noise and outlier problem mentioned above, with regard to text data, other essential features makes it discriminated between data from other fields. One case is called semantic diversity that the meaning of a polysemous words may vary a lot when the word appears in different ranges of linguistic contexts [71]. For instance, the word *war* can be mentioned in movies, in military operations, in political consultations or even in someones memories. It also happens sometimes accompanied by homonyms with multiple unrelated meanings. Another case is caused by the versatility of some words that are highly domain specific [157]. Those words are easy to be recognised as keywords for all the topics under this or related domains; however, the results of the detection hardly present further specific information since the keywords are of no distinctiveness in the inner of domains. For example, the word *algorithm* is a distinct term for articles of computing science rather than those of other fields, however, within the computing science domain, it is not enough to discriminate which specific topic, like cloud computing or health informatics, the article is working on. And worse than that is the rapid generation of text data facilitated by the online social network platforms which brings more complicated textual content and social context that results in the constantly changing latent topics.

To remove the repercussions caused by outliers and overcome nature properties of text data mentioned above, in this chapter, we take inspirations from two perspectives.

One perspective focuses on robustness of loss function. Formally, a corpus can be represented by a term-document matrix $\mathbf{X} = [X_1, \cdots, X_n]$, and each document X_i is a tf-idf vector over the vocabulary of m terms. NMF factorises $\mathbf{X}$ into two non-negative

matrices $\mathbf{H} \in \mathbb{R}_{+}^{m \times k}$, $\mathbf{V} \in \mathbb{R}_{+}^{n \times k}$ as $\mathbf{X} \approx \mathbf{H}\mathbf{V}^{T}$ where $k \ll \min(m, n)$ is the number of clusters and $\mathbf{H}$ reveals the k topics by sequences of terms. Typically, the number m can become pretty large, which brings researchers the high dimension problem and a significant chance of outliers; no matter what kind of linguistic representation model and topic model being applied. For this challenge, many works turn towards emphasising the robustness of applied models, for example making use of constraints between data points in the original data space [102], reducing the domination of errors in optimization objectives [47, 87] and developing collect matrix factorization with supplementary information obtained from social context. However, the uncertainty of additional outliers and noises also arises along with the combination of supplementary information, returning to the essential requirement of robustness. In this chapter, to boost the robustness of the optimisation process, we fisrt introduce $\ell_{2,1}$-norm to construct collective NMF objective function. Comparing with F-norm based objectives, we demonstrate the corresponding performance through our experiments over several real datasets.

The other perspective is based the topic hierarchy to address the semantic diversity and versatility. In real scenarios, it is very common to find that both the positive and negative opinions of a topic that appearing in a moment will develop into various perspectives and evolve into different related topics in the following consecutive days until the attention from the public quiets down. Thus, the upcoming topics may have somewhat correlations comparing to others which are completely irrelevant. Based on this observation, we propose a hierarchical topic detection scheme here to discover topics incrementally. This scheme is implemented with the hierarchical document clustering. A similar work can be reviewed in [90] which always divide a set of documents into two parts; however, our method will be more flexible and practical with regard to the real data. We first roughly partition the whole set of documents into several parts which can be more than two, followed by further partitioning. And, more remarkable, with the partitioning, dissidents will be removed from a cluster in an early stage, and the subsets will be cleaner. NMF is also known as an excellent soft clustering and data representation method. In our work, it is embedded in our hierarchy to output explicit clustering labels of documents and corresponding topics.

Another related matter is how to decide the number of topics in practical. Among

the existing topic models [131, 157], a common practice is to pre-define a constant k as the specific number of latent concepts, which means that the number of unknown topics is assumed without prior knowledge of data distributions. As a usual strategy in topic detection and document clustering, it is pragmatic to most applications to some extent, but not always suitable for discovering the quickly evolving and changing scenario of topics under the present age. To adapt the unknown topic distribution, we design a strategy of pruning to stop hierarchy when there is no more meaningful sub-category of existing leaf nodes. To judge the meaningfulness of candidatures, the leaf nodes, we design a two-step examining policy, checking how many valuable parts the candidatures can be divided into and how effective if a candidature is selected to be subdivided further. We will expand on this policy in Section 5.5, including the pruning strategy and outlier verification. In addition, we obtained the clusters of documents from the uniqueness of the orthonormal NMF [46].

Overall, the main contributions of this chapter are:

- We propose a Robust Hierarchical Ensemble (RHE) framework based on collective non-negative matrix factorization algorithm for topic detection associated with hierarchical document clustering in data with serious outliers and noises.

- During the hierarchical clustering, we use an adaptive k for clustering in each level, which gives us a flexible and practical interpretation of the topic detection procedure and reduces the dependence of the algorithm on a predefined number of topics. Therefore, the total number of topics in a time-step can be either set or not, that may only determine the termination condition of the hierarchy algorithm since it will stop at a moment when there is no more meaningful sub-topic containing in existing topics.

- For correctness of hierarchy, we scrutinise outliers carefully including the exclusion of suspects, and re-examination of dataset without these suspects followed by reversion of excluded set if necessary.

- Both $\ell_{2,1}$-norm and F-norm based objective functions are discussed, and three updating rules are provided for optimisation.

- Extensive experiments on both real datasets and synthetic datasets demonstrate the remarkable effectiveness of hierarchy structure. The results also confirm that our RHE methods significantly outperforms other baselines in robustness. Besides, a comparative analysis of the performance results of different objective functions that use $\ell_{2,1}$-norm and F-norm is conducted based on our results.

The rest of the paper is organised as follow. We first discuss some related works in Section 5.2 and review the existing NMF based methods as preliminaries and clarify the notations in Section 5.3. In Section 5.4, we propose our objective function and deduce updating rules for optimisation. In Section 5.5, we introduce our adaptive k-ary tree based hierarchical structure in detail, including the hierarchy construction, the candidature selection, the pruning strategy and the outlier verification. The experiments on real data sets are shown in Section 5.6 followed by conclusion in Section 5.7.

5.2 Related work

There is a large concentration of researchers focusing on the improvements in topic detection algorithms and topic models [131,157] confined to textual data itself. Generally, there are two directions. The first type of works eyes on datasets. Some of them [24, 143, 144] imposed an extra regularisation term of the local manifold information extracted from the original data space and succeed in leveraging the intrinsic geometry of the data distribution. We will review this approach in the next section as a significant preliminary. [102] followed this idea proposed an upgrade version, using so-called hard constraints, for particular datasets, of which partially labelled data is available. However, apparent auxiliary information is always limited to obtain, and sometimes it has no symmetry and transitivity, for example in multi-cluster or soft-cluster tasks. Therefore, some others [107, 171] exploited constraints more in-depth with a constraint propagation scheme in the entire data set. Another type of works improves the factorization optimisation function. No matter what kind of optimization solution is adopted, such as active-set method [90], Lagrange multiplier method [170, 192] and coordinate descent method [157], the conventional Frobenius norm (F-nrom) loss function used to measure the convergence is pointed out [76] that it would be significantly prone to ambient noises and outliers

under F-nrom framework because of the squared error; however, the alternative l_1-norm function fails to fulfill the rotation invariance [47, 119]. Be aware of this, some works [76, 87, 192] innovated the objective function by using a mixed norm, $l_{2,1}$-norm, as a more robust and adaptive option for practical data.

Meanwhile, other ideas emerged with the rise of online social network. Inspired by collective matrix factorization (CMF) for multi-view data [48] and multi-relation data [149], Kalyanam et al. proposed LETCS algorithm adopting textual content related social information that was collected from social service providers as a component of their multi-view matrices [81]. Our work [176] in last chapter put forward a CMF algorithm of multi-relational topic detection by exploiting the correlation between relations of terms-documents and documents-users, that has demonstrated that the supplementary information, representing by the relational matrix of documents-users where users are grouped by their interested documents, is helpful for discovering ongoing topics.

Before our work, the structure of categories and sub-categories with regard to topic definition were recognised and widely used as a novel view for organising and understanding text corpus. For example, in Reuter Corpus Volume I [95] adopted a topics hierarchy for Reuters documentation including 103 topic codes which distribute on different hierarchy, subordinating to the parent topic code of each. For clustering tasks out of text mining, hierarchical clustering developed upon the clustering objective functions of K-means, Gaussian mixture and MinMaxCut can be categorised as either bottom-up agglomerative approach or top-down divisive approach [45]. Fred and Jain proposed a clustering ensemble combining strategy by exploring the idea of evidence accumulation [60]. Their method applied certain combination techniques to combine different clustering partitions of a given data set in order to obtain a partition that is better than any individual partition. Gionis et al. formally stated the clustering aggregation problem in [63] where various algorithms making use of the connection between clustering aggregation and the problem of correlation clustering were proposed for the problem and large datasets. Both of them are bottom-up methods. As for hierarchy method integrating NMF clustering, to our best knowledge, research is still at the starting stage. There is only a few works can be found. Kuang and Park proposed a rank-2 NMF method with active-set-type algorithm to implement recursively clustering the text corpus into two parts [90].

Tu and Chen et al. proposed an adaptively hierarchical online method based on ONMF [110] algorithm for text stream which always relies on an initial clustering of documents determining the topic number [166].

5.3 Preliminary on Non-Negative Matrix Factorization

5.3.1 Notations

In our paper, matrices are written in boldface capital letters, such as $\mathbf{X}$ and $\mathbf{H}$. Vectors are the column vectors of a matrix, written in italic capital letters, such as X. Lowercase with subscript, such as X_m and x_{ij}, refers to the element of a vector or a matrix respectively. For the matrix $\mathbf{X} = \{x_{ij}\}$, X_j denotes the j-th column vector of $\mathbf{X}$. $\mathrm{tr}(\mathbf{X})$ is the trace of $\mathbf{X}$ and all the parameters are represented by Greek lowercase letters.

5.3.2 Clustering with Standard NMF

In Chapter 4, we have formalised the NMF problem for topic detection problem. It has also been widely used as a clustering approach [183, 191] owing to the multiple explanations of $\mathbf{H}$ and $\mathbf{V}$. Columns of $\mathbf{H}$ are the basis vectors of the new low-rank space [90] or cluster centroids [192] and $\mathbf{V}$ is the low-dimensional feature representation [191] or the weights of data points associated with cluster centroids [76]. With the constraint of orthogonality on columns of $\mathbf{V}$ applied, $\mathbf{V}$ is more likely to be the cluster indicator for clustering columns of $\mathbf{X}$. The orthogonal non-negative matrix factorization is defined as $\mathbf{X} \approx \mathbf{H}\mathbf{V}^T, s.t. \mathbf{V}^T\mathbf{V} = \mathbf{I}$. Through orthogonal NMF, each column X_j of $\mathbf{X}$ represents a document $d_j \in D$, the input corpus, then the columns of $\mathbf{H}$ can be directly interpreted to separate topics extracted from D, while the position of the maximum value in each row of $\mathbf{V}$ represents the document clustering label $c_j = \mathrm{argmax}_j v_{ij}$ of data point X_j. The objective function that quantifies the approximation of standard NMF using Euclidean distance is the square of Frobenius norm $\min_{\mathbf{H},\mathbf{V}\geq 0} \|\mathbf{X} - \mathbf{H}\mathbf{V}^T\|_F^2$. Previous works [24, 94] indicated that only local minima with regard to each variable could be found for this problem and [94] proposed an iterative "multiplicative update rules" which is efficient and also easy to implement compared to existing gradient descent methods of quick convergence. We

will recount our proposing method under this updating rule in Section 5.4.

5.3.3 Semi-supervised NMF

By adding some auxiliary information of the original data space, for example, hard constraint, like the label information of some of the data points [102], or the inherent weak connection between data points in the original space [24],supervised information was introduced into standard NMF spawning semi-supervised NMF. Objective functions are $\min_{\mathbf{H},\mathbf{Z}\geq 0} \|\mathbf{X} - \mathbf{H}\mathbf{Z}^T\mathbf{A}^T\|_F^2$ and $\min_{\mathbf{H},\mathbf{V}\geq 0} \|\mathbf{X} - \mathbf{H}\mathbf{V}^T\|_F^2 + \lambda \operatorname{tr}(\mathbf{V}^T\mathbf{L}\mathbf{V})$ correspondingly, where $\mathbf{A}$ is the label matrix and $\mathbf{Z}$ is the auxiliary matrix for approximation of the original matrix for the former and $\mathbf{L} = \mathbf{D} - \mathbf{W}$ denotes the Laplacian graph matrix which came from the assumption of spectral clustering that higher weight w_{ij} will be given to if two data points are close to each other, while low weight corresponds to those far from each other. Each diagonal element $d_{ii} = \sum_{j=1}^{n} w_{ij}$ of degree matrix $\mathbf{D}$ denotes the sum of weights of data points adjacent to the data point i and other off-diagonal elements $d_{ij} = 0$. However, since the connections between data points are too weak to be efficiently leveraged. Previous work [171] developed this assumption by enhancing the inter-connections of data points in image with similar objective functions. in last chapter, we proposed two manifold collective NMF algorithms working on text data delighted by this idea. For more details, please refer to Chapter 4.

5.3.4 Collective NMF

The idea of collective non-negative matrix factorization is also an important foundation of our works. Recently, it has been used in text data analysis with the prevalent of online social networks. We have discussed the theory in last chapter, Section 4.3.2 and have to emphasise the importance of outlier issue that may caused by the combination of social information. From this point of view, the motivation of finding a robust model for topic detection and document clustering for text data under online social context is necessary.

5.3.5 $\ell_{2,1}$-norm NMF: Seeking Robustness

The above algorithms were implemented with Frobenius norm. For robust purpose, $\ell_{2,1}$-norm based objective function was proposed to avoid the disadvantage of being prone to noises. Under the definition of the norm in ℓ_p-space, ℓ_2-norm of a vector $X = \{x_1, \cdots, x_m\}^T \in \mathbb{R}^n$ is defined as $\|\mathbf{X}\|_2 = \sqrt{\sum_{i=1}^m x_i^2}$. It was firstly extended to a matrix in [47] with the name $\ell_{2,1}$-norm of the matrix $\mathbf{X}$ and its form $\|\mathbf{X}\|_{2,1} = \sum_{j=1}^n \sqrt{\sum_{i=1}^m x_{ij}^2} = \sum_{j=1}^n \|X_j\|_2$ satisfies following conditions [87]:

1. Rotational invariance: $\|\mathbf{A}\|_{2,1} = \|\operatorname{Rotate}(\mathbf{A})\|_{2,1}$;
2. Positive scalability: for all scalar α, $|\alpha|\|\mathbf{A}\|_{2,1} = \|\alpha\mathbf{A}\|_{2,1}$;
3. Triangle inequality: $\|\mathbf{A} + \mathbf{B}\|_{2,1} \leq \|\mathbf{A}\|_{2,1} + \|\mathbf{B}\|_{2,1}$;
4. non-negative valued: $\|\mathbf{A}\|_{2,1} \geq 0$ and 5) definiteness: $\|\mathbf{A}\|_{2,1} = 0 \iff a_{ij} = 0$.

The objective function of $\ell_{2,1}$-norm NMF thereby changes into $\min_{\mathbf{H},\mathbf{V}\geq 0} \|\mathbf{X} - \mathbf{H}\mathbf{V}^T\|_{2,1}$ and theoretically, the impact caused by noises and outliers could be reduced because of the square root of Euclidean distance adopted on data approximation. Furthermore, a novel updating method was introduced to iterative process of $\ell_{2,1}$-norm NMF, Augmented Lagrange Multiplier (ALM) [15], which has faster convergence and allows a slackness of constraints on $\mathbf{X} \in \mathbb{R}_+^{m\times n}$ for particular tasks.

5.4 Robust NMF via $\ell_{2,1}$-norm

In this chapter, we propose the following Collective Non-negative Matrix Factorization objective function for our joint task: hierarchical document clustering assisted topic detection.

$$\begin{aligned} f = &\operatorname*{argmin}_{\mathbf{H},\mathbf{V},\mathbf{G}} \mu f_{text} + (1-\mu) f_{com} \\ \text{s.t.} \quad &\mathbf{V}^T\mathbf{V} = \mathbf{I} \text{ and } \mathbf{H}, \mathbf{V}, \mathbf{G} \geq 0 \end{aligned} \tag{5.1}$$

where $f_{text} = \|\mathbf{X} - \mathbf{H}\mathbf{V}^T\|_{2,1} + \lambda_1 \operatorname{tr}(\mathbf{V}^T\mathbf{L}_\mathbf{X}\mathbf{V})$ represents the approximation factorization on the information of textual content and $f_{com} = \|\mathbf{U} - \mathbf{V}\mathbf{G}^T\|_{2,1} + \lambda_2 \operatorname{tr}(\mathbf{G}^T\mathbf{L}_\mathbf{U}\mathbf{G})$ corresponds to the approximation factorization on the social community context. We use

$\mathbf{X} \in \mathbb{R}_{+}^{m \times n}$ and $\mathbf{U} \in \mathbb{R}_{+}^{n \times l}$ to emphasise the active l users and released n documents, represented by the linear combination of m words. These two parts constitute our above collective NMF optimization problem. The parameter $\mu \in [0, 1]$ balances the relative proportion of two parts. Thus, the surrounding community can be completely ignored by setting μ to 1 as existing majority topic detection works do. Differ from the Frobenius norm objective function, $\|\cdot\|_F$ is substituted to $\|\cdot\|_{2,1}$ in the objective in Eq. 5.1, neither of which is convex in all of the above variables. Alternatively, we turn our target into finding its local minimum instead of a global minimum updating each of the variables. Since the constraints of our new objective function are remained non-negative, transforming the universal used multiplicative updating rule to fit our proposed objective function is one intuitive option for updating the variables, utilising the Augmented Lagrange Multiplier (ALM) is also a possible way we considering. We will present the derivations of our updating rule for both methods in the next parts of this section, followed by the actual performances of each method tested on real data sets shown in the experiments section.

5.4.1 Optimization with Lagrangian Multiplier and Multiplicative Update Rules

We use notations $\omega_{kk'}, \varphi_{ik}, \psi_{jk}$ and ρ_{pk} as the Lagrangian multipliers for enforcing constraints $(\mathbf{V}_t^T \mathbf{V}_t - \mathbf{I})_{kk'} = 0$ and $v_{ik} \in \mathbf{V}, h_{jk} \in \mathbf{H}$ and $g_{pk} \in \mathbf{G} \geq 0$ respectively, and $\Omega = [\omega_{kk'}]$, $\Phi = [\varphi_{ik}]$, $\Psi = [\psi_{jk}]$ and $\mathrm{P} = [\rho_{pk}]$. Now we can define the Lagrangian function $\mathcal{L}$ as:

$$\mathcal{L} = f + tr(\Omega(\mathbf{V}^T\mathbf{V} - \mathbf{I})^T + \Phi\mathbf{V}^T + \Psi\mathbf{H}^T + \mathrm{P}\mathbf{G}^T) \tag{5.2}$$

Generally, for Lagrangian multipliers Ω, there are no specific constraints from KKT conditions. Also, the off-diagonal entities do not need to be zero. And, because of the rotational invariance property, Ω is a symmetric matrix of size $k \times k$. Similar to the widely-adopted Multiplicative update rules [94], we follow the standard Lagrangian multiplier theory to update each variable by fixing the others and equivalently address the following partial derivatives respect to each variable plus Lagrangian multipliers as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = \frac{\partial (f + tr(\Omega(\mathbf{V}^T\mathbf{V} - \mathbf{I})^T + \Phi\mathbf{V}^T))}{\partial \mathbf{V}} = 0 \tag{5.3}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{H}} = \frac{\partial (||\mathbf{X} - \mathbf{H}\mathbf{V}^T||_{2,1} + tr(\Psi\mathbf{H}^T))}{\partial \mathbf{H}} = 0 \tag{5.4}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{G}} = \frac{\partial (||\mathbf{U} - \mathbf{V}\mathbf{G}^T||_{2,1} + \lambda_2 tr(\mathbf{G}^T\mathbf{L}_\mathbf{U}\mathbf{G}) + tr(\mathrm{P}\mathbf{G}^T))}{\partial \mathbf{G}} = 0 \tag{5.5}$$

We use KKT conditions to solve this inequality constrained optimization problem. Besides the above stationarity conditions, We have the complementary slackness of $\varphi_{ik}v_{ik} = 0$, $\psi_{jk}h_{jk} = 0$ and $\rho_{pk}h_{pk} = 0$. Taking the stationary condition in Eq. 5.3 and the primal feasibility $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, we have $\Omega = \mathbf{V}_t^T \cdot (-\frac{\partial f}{\partial \mathbf{V}_t})$.

Before iteratively update variables until the objective function converges, we define two auxiliary diagonal matrices $\mathbf{Dx} \in \mathbb{R}^{n\times n}$ and $\mathbf{Du} \in \mathbb{R}^{l\times l}$ to simplify expressions of updating rules. Their diagonal elements are defined as $\mathbf{Dx}_{ii} = \left(\sqrt{\sum_{j=1}^{m}(\mathbf{X} - \mathbf{H}\mathbf{V}^T)_{ji}^2}\right)^{-1}$ and $\mathbf{Du}_{pp} = \left(\sqrt{\sum_{i=1}^{n}(\mathbf{U} - \mathbf{V}\mathbf{G}^T)_{ip}^2}\right)^{-1}$.

The above equations lead to the following update rules:

$$\mathbf{V} \leftarrow \mathbf{V} \odot \frac{\mathbf{A} + \mathbf{V}\mathbf{V}^T\mathbf{B}}{\mathbf{V}\mathbf{V}^T\mathbf{A} + \mathbf{B}} \tag{5.6}$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{X}\mathbf{D}\mathbf{x}\mathbf{V}}{\mathbf{H}\mathbf{V}^T\mathbf{D}\mathbf{x}\mathbf{V}} \tag{5.7}$$

$$\mathbf{G} \leftarrow \mathbf{G} \odot \frac{\mathbf{D}\mathbf{u}\mathbf{U}^T\mathbf{V} + \lambda_2\mathbf{W}_\mathbf{U}\mathbf{G}}{\mathbf{D}\mathbf{u}\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\mathbf{D}_\mathbf{G}\mathbf{G}} \tag{5.8}$$

where $\mathbf{A} = \mu(\mathbf{D}\mathbf{x}\mathbf{X}^T\mathbf{H} + \lambda_1\mathbf{W}_\mathbf{X}\mathbf{V}) + (1-\mu)\mathbf{U}\mathbf{D}_\mathbf{U}\mathbf{G}$ and $\mathbf{B} = \mathbf{D}_\mathbf{X}\mathbf{V}\mathbf{H}^T\mathbf{H} + \lambda_1\mathbf{D}_\mathbf{V}\mathbf{V}$. If F-norm based objective function is adopted, the updating rules could be derived similarly as follows:

$$\mathbf{V} \leftarrow \mathbf{V} \odot \frac{\mathbf{A}' + \mu\lambda_1\mathbf{V}\mathbf{V}^T\mathbf{D}_\mathbf{V}\mathbf{V}}{\mu\mathbf{V}\mathbf{V}^T\mathbf{A}' + \mu\lambda_1\mathbf{D}_\mathbf{V}\mathbf{V}} \tag{5.9}$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{X}\mathbf{V}}{\mathbf{H}\mathbf{V}^T\mathbf{V}} \tag{5.10}$$

$$\mathbf{G} \leftarrow \mathbf{G} \odot \frac{\mathbf{U}^T\mathbf{V} + \lambda_2\mathbf{W}_\mathbf{U}\mathbf{G}}{\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\mathbf{D}_\mathbf{G}\mathbf{G}} \tag{5.11}$$

where $\mathbf{A}' = \mu(\mathbf{X}^T\mathbf{H} + \lambda_1\mathbf{W}_\mathbf{X}\mathbf{V}) + (1-\mu)\mathbf{U}\mathbf{G}$.

5.4.2 Convergence of Lagrangian Multiplier Method

Regarding the updating rules derived in last section, we have the following two theorem:

Theorem 5.1. *With the updating rules in Eq. 5.6 to Eq. 5.8, the objective function in Eq. 5.1 is non-increasing.*

Theorem 5.2. *With the updating rules in Eq. 5.9 to Eq. 5.11, the F-norm based objective function in Eq. 5.1 where $\ell_{2,1}$-norm is replaced by F-norm is non-increasing.*

Next, we follow the details in Section 4.5.3 to prove Theorem 5.1 and Theorem 5.2 separately.

Proof of Theorem 5.1

Because of Definition 4.1, the key step to prove Theorem 5.1 is to find a proper auxiliary function with respect to $\mathbf{V}$, $\mathbf{H}$ and $\mathbf{G}$. We rewrite the objective function in Eq. 5.1 as follow:

$$O = \mu \left(\left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T \right\|_{2,1} + \lambda_1 \operatorname{tr}(\mathbf{V}^T \mathbf{L}_\mathbf{X} \mathbf{V}) \right) + (1-\mu) \left(\left\| \mathbf{U} - \mathbf{V}\mathbf{G}^T \right\|_{2,1} + \lambda_2 \operatorname{tr}(\mathbf{G}^T \mathbf{L}_\mathbf{U} \mathbf{G}) \right)$$

We use $F_\mathbf{V}$, $F_\mathbf{H}$ and $F_\mathbf{G}$ to denote the part of O which is only relevant to $\mathbf{V}$, $\mathbf{H}$ and $\mathbf{G}$ respectively.

Updating V We focus on updating $\mathbf{V}$ while fixing $\mathbf{H}$ and $\mathbf{G}$, then we rewrite the objective function relevant to V as:

$$F_\mathbf{V} = \mu \left(\left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T \right\|_{2,1} + \lambda_1 \operatorname{tr}(\mathbf{V}^T \mathbf{L}_\mathbf{X} \mathbf{V}) \right) + (1-\mu) \left\| \mathbf{U} - \mathbf{V}\mathbf{G}^T \right\|_{2,1} \tag{5.12}$$

Lemma 5.1. *Function*

$$J(v, v_{ab}^{(t)}) = F_\mathbf{V}(v_{ab}^{(t)}) + F'_\mathbf{V}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) + \frac{(\mathbf{V}\mathbf{V}^T\mathbf{A} + \mathbf{B})_{ab}}{v_{ab}^{(t)}}(v - v_{ab}^{(t)})^2 \tag{5.13}$$

is an auxiliary function for $F_\mathbf{V}$.

Proof. $J(v, v) = F_{\mathbf{V}}(v)$ is trivial. Then we only need to prove $J(v, v_{ab}^{(t)}) \geq F_{\mathbf{V}}(v)$. To do this, we have the Taylor series expansion of $F_{\mathbf{V}}(v)$:

$$F_{\mathbf{V}}(v) = F_{\mathbf{V}}(v_{ab}^{(t)}) + F'_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) + \frac{1}{2}F''_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)})^2. \tag{5.14}$$

For each element v_{ab} in $\mathbf{V}$, it is easy to derive that

$$\begin{aligned} F'_{\mathbf{V}}(v_{ab}^{(t)}) =& \frac{\partial F_{\mathbf{V}}}{\partial v_{ab}} \\ =& \mu[\mathbf{D}_{\mathbf{X}aa}(-2\mathbf{X}^T\mathbf{H} + 2\mathbf{V}\mathbf{H}^T\mathbf{H})_{ab} + (2\lambda_1\mathbf{L}_{\mathbf{X}}\mathbf{V})_{ab}] \\ &+ (1-\mu)[(-2\mathbf{U}\mathbf{D}_{\mathbf{U}}\mathbf{G})_{ab} + (2\mathbf{V}\mathbf{G}^T\mathbf{D}_{\mathbf{U}}\mathbf{G})_{ab}], \end{aligned}$$

and

$$F''_{\mathbf{V}}(v_{ab}^{(t)}) = \mu[\mathbf{D}_{\mathbf{X}aa} \cdot (2\mathbf{H}^T\mathbf{H})_{bb} + (2\lambda_1\mathbf{L}_{\mathbf{X}})_{aa}] + (1-\mu) \cdot (2\mathbf{G}^T\mathbf{D}_{\mathbf{U}}\mathbf{G})_{bb}.$$

Thus, by Eq. 5.13 and Eq. 5.14, $J(v, v_{ab}^{(t)}) \geq (F_{\mathbf{V}})_{ab}(v)$ is equivalent to:

$$\begin{aligned} &\frac{(\mathbf{V}\mathbf{V}^T\mathbf{A} + \mathbf{B})_{ab}}{v_{ab}^{(t)}} \geq \frac{1}{2}F''_{\mathbf{V}}(v_{ab}^{(t)}) \\ =& \mu[\mathbf{D}_{\mathbf{X}aa}(\mathbf{H}^T\mathbf{H})_{bb} + \lambda_1\mathbf{L}_{\mathbf{X}aa}] + (1-\mu)(\mathbf{G}^T\mathbf{D}_{\mathbf{U}}\mathbf{G})_{bb} \end{aligned} \tag{5.15}$$

We have:

$$\begin{aligned} &(\mathbf{V}\mathbf{V}^T\mathbf{A})_{ab} \geq (1-\mu)(\mathbf{V}\mathbf{V}^T\mathbf{U}\mathbf{D}_{\mathbf{U}}\mathbf{G})_{ab} = (1-\mu)(\mathbf{V}\mathbf{G}^T\mathbf{D}_{\mathbf{U}}\mathbf{G})_{ab} \\ &\geq (1-\mu)\sum_{p=1}^{l} v_{ap}^{(t)}(\mathbf{G}^T\mathbf{D}_{\mathbf{U}}\mathbf{G})_{pb} = (1-\mu)v_{ab}^{(t)}(\mathbf{G}^T\mathbf{D}_{\mathbf{U}}\mathbf{G})_{bb}, \end{aligned}$$

and $\mathbf{B} = (\mathbf{D}_{\mathbf{X}}\mathbf{V}\mathbf{H}^T\mathbf{H} + \lambda_1\mathbf{D}_{\mathbf{V}}\mathbf{V})_{ab}$, where

$$(\mathbf{D}_{\mathbf{X}}\mathbf{V}\mathbf{H}^T\mathbf{H})_{ab} \geq \mu(\mathbf{D}_{\mathbf{X}}\mathbf{V}\mathbf{H}^T\mathbf{H})_{ab} = \mu\mathbf{D}_{\mathbf{X}aa}\sum_{i=1}^{k} v_{ai}^{(t)}(\mathbf{H}^T\mathbf{H})_{ib} \geq \mu\mathbf{D}_{\mathbf{x}aa} \cdot v_{ab}^{(t)}(\mathbf{H}^T\mathbf{H})_{bb},$$

and

$$(\lambda_1 \mathbf{D_V V})_{ab} = \lambda_1 \sum_{j=1}^{n} \mathbf{D_V}_{aj} v_{jb}^{(t)} \geq \lambda_1 \mathbf{D_V}_{aa} v_{ab}^{(t)} \geq \mu\lambda_1 (\mathbf{D_V} - \mathbf{W_X})_{aa} v_{ab}^{(t)} = \mu\lambda_1 \mathbf{L_X}_{aa} v_{ab}^{(t)}.$$

Thus, Eq. 5.15 holds and $J(v, v_{ab}^{(t)}) \geq F_{\mathbf{V}}(v)$. □

We can derive the following updating rule for v_{ab} by replacing auxiliary function in Eq. 4.28 with Eq. 5.13:

$$v_{ab}^{(t+1)} = v_{ab}^{(t)} - v_{ab}^{(t)} \frac{F'_{\mathbf{V}}(v_{ab}^{(t)})}{(\mathbf{V}\mathbf{V}^T\mathbf{A} + \mathbf{B})_{ab}} = v_{ab}^{(t)} \frac{(\mathbf{A} + \mathbf{V}\mathbf{V}^T\mathbf{B})_{ab}}{(\mathbf{V}\mathbf{V}^T\mathbf{A} + \mathbf{B})_{ab}}$$

Since Eq. 5.13 is an auxiliary function, $F_{\mathbf{V}}$ is non-increasing under this updating rule.

Updating H We focus on updating $\mathbf{H}$ while fixing $\mathbf{V}$ and $\mathbf{G}$. We rewrite the objective function as:

$$F_{\mathbf{H}} = \mu \left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T \right\|_{2,1} \tag{5.16}$$

Lemma 5.2. *Function*

$$J(h, h_{ab}^{(t)}) = F_{\mathbf{H}}(h_{ab}^{(t)}) + F'_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)}) + \frac{(\mathbf{H}\mathbf{V}^T\mathbf{D_X V})_{ab}}{h_{ab}^{(t)}}(h - h_{ab}^{(t)})^2 \tag{5.17}$$

is an auxiliary function for $F_{\mathbf{H}}$.

Proof. $J(h, h) = F_{\mathbf{H}}(h)$ is obvious. Then we prove $J(h, h_{ab}^{(t)}) \geq F_{\mathbf{H}}(h)$. Similar to the proof of lemma 1, we first write the Taylor series expansion of $F_{\mathbf{H}}(h)$ as:

$$F_{\mathbf{H}}(h) = F_{\mathbf{H}}(h_{ab}^{(t)}) + F'_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)}) + \frac{1}{2} F''_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)})^2 \tag{5.18}$$

where $F'_{\mathbf{H}}(h_{ab}^{(t)}) = (-2\mathbf{X}\mathbf{D_X V} + 2\mathbf{H}\mathbf{V}^T\mathbf{D_X V})_{ab}$, and $F''_{\mathbf{H}}(h_{ab}^{(t)}) = 2(\mathbf{V}^T\mathbf{D_X V})_{ab}$. By comparing Eq. 5.17 with Eq. 5.18 we find that $J(h, h_{ab}^{(t)}) \geq F_{\mathbf{H}}(h)$ is equivalent to

$$\frac{(\mathbf{H}\mathbf{V}^T\mathbf{D_X V})_{ab}}{h_{ab}^{(t)}} \geq (\mathbf{V}^T\mathbf{D_X V})_{ab} \tag{5.19}$$

We have $(\mathbf{H}\mathbf{V}^T\mathbf{D_X V})_{ab} = \sum_{i=1}^{k} h_{ai}^{(t)} (\mathbf{V}^T\mathbf{D_X V})_{ib} \geq h_{ab}^{(t)} (\mathbf{V}^T\mathbf{D_X V})_{ab}$. Thus, the matrix in-

equality in Eq. 5.19 holds and $J(h, h_{ab}^{(t)}) \geq (F_{\mathbf{H}})_{ab}(h)$. □

We can now demonstrate the convergence of Eq. 5.1 under the following updating rule by replacing Eq. 4.28 with $J(h, h_{ab}^{(t)})$ in Eq. 5.17:

$$h_{ab}^{(t+1)} = h_{ab}^{(t)} - h_{ab}^{(t)} \frac{F'_{\mathbf{H}}(h_{ab}^{(t)})}{(\mathbf{H}\mathbf{V}^T\mathbf{D}_{\mathbf{X}}\mathbf{V})_{ab}} = h_{ab}^{(t)} \frac{(\mathbf{X}\mathbf{D}_{\mathbf{X}}\mathbf{V})_{ab}}{(\mathbf{H}\mathbf{V}^T\mathbf{D}_{\mathbf{X}}\mathbf{V})_{ab}}$$

Since Eq. 5.17 is an auxiliary function, $F_{\mathbf{H}}$ is non-increasing under this updating rule.

Updating G We focus on updating $\mathbf{G}$ while fixing $\mathbf{V}$ and $\mathbf{H}$. We rewrite the objective function relevant to $\mathbf{G}$ as

$$F_{\mathbf{G}} = \left\| \mathbf{U} - \mathbf{V}\mathbf{G}^T \right\|_{2,1} + \lambda_2 \operatorname{tr}(\mathbf{G}^T\mathbf{L}_{\mathbf{U}}\mathbf{G}) \tag{5.20}$$

Lemma 5.3. *Function*

$$J(g, g_{ab}^{(t)}) = F_{\mathbf{G}}(g_{ab}^{(t)}) + F'_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)}) + \frac{(\mathbf{D}_{\mathbf{U}}\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\mathbf{D}_{\mathbf{G}}\mathbf{G})_{ab}}{g_{ab}^{(t)}}(g - g_{ab}^{(t)})^2 \tag{5.21}$$

is an auxiliary function for $F_{\mathbf{G}}$.

Proof. $J(g, g) = F_{\mathbf{G}}(g)$ is obvious. Then we only need to prove $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$. We compare the Taylor series expansion of $F_{\mathbf{G}}(g)$ as:

$$F_{\mathbf{G}}(g) = F_{\mathbf{G}}(g_{ab}^{(t)}) + F'_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)}) + \frac{1}{2}F''_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)})^2 \tag{5.22}$$

where $F'_{\mathbf{G}}(g_{ab}^{(t)}) = (-2\mathbf{D}_{\mathbf{U}}\mathbf{U}^T\mathbf{V} + 2\lambda_2\mathbf{L}_{\mathbf{U}}\mathbf{G})_{ab} + 2\mathbf{D}_{\mathbf{U}aa}(\mathbf{G}\mathbf{V}^T\mathbf{V})_{ab}$ and $F''_{\mathbf{G}}(g_{ab}^{(t)}) = 2\lambda_2\mathbf{L}_{\mathbf{U}aa} + 2\mathbf{D}_{\mathbf{U}aa}(\mathbf{V}^T\mathbf{V})_{bb}$. By comparing Eq. 5.21 with Eq. 5.22, we find that $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$ is equivalent to

$$\frac{(\mathbf{D}_{\mathbf{U}}\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\mathbf{D}_{\mathbf{G}}\mathbf{G})_{ab}}{g_{ab}^{(t)}} \geq \frac{1}{2}(F_{\mathbf{G}})''_{ab} = \lambda_2\mathbf{L}_{\mathbf{U}aa} + \mathbf{D}_{\mathbf{U}aa}(\mathbf{V}^T\mathbf{V})_{bb} \tag{5.23}$$

We have $(\mathbf{D}_{\mathbf{U}}\mathbf{G}\mathbf{V}^T\mathbf{V})_{ab} = \mathbf{D}_{\mathbf{U}aa}\sum_{i=1}^{k} g_{ai}^{(t)}(\mathbf{V}^T\mathbf{V})_{ib} \geq \mathbf{D}_{\mathbf{U}aa}g_{ab}^{(t)}(\mathbf{V}^T\mathbf{V})_{bb}$ and $(\lambda_2\mathbf{D}_{\mathbf{G}}\mathbf{G})_{ab} = \lambda_2\sum_{p=1}^{l}\mathbf{D}_{\mathbf{G}ap}g_{pb}^{(t)} \geq \lambda_2\mathbf{D}_{\mathbf{G}aa}g_{ab}^{(t)} \geq \lambda_2(\mathbf{D}_{\mathbf{G}} - \mathbf{W}_{\mathbf{U}})_{aa}g_{ab}^{(t)} = \lambda_2(\mathbf{L}_{\mathbf{U}})_{aa}g_{ab}^{(t)}$. Thus, Eq. 5.23 holds and $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$. □

We can now replace Eq. 4.28 by $J(g, g_{ab}^{(t)})$ in Eq. 5.21 to derive the update rule:

$$g_{ab}^{(t+1)} = g_{ab}^{(t)} - g_{ab}^{(t)} \frac{F'_{\mathbf{G}}(g_{ab}^{(t)})}{(\mathbf{D_U G V}^T\mathbf{V} + \lambda_2 \mathbf{D_G G})_{ab}} = g_{ab}^{(t)} \frac{(\mathbf{D_U U}^T\mathbf{V} + \lambda_2 \mathbf{W_U G})_{ab}}{(\mathbf{D_U G V}^T\mathbf{V} + \lambda_2 \mathbf{D_G G})_{ab}}$$

since Eq. 5.21 is an auxiliary function, $F_{\mathbf{G}}$ is non-increasing under this updating rule.

Proof of Theorem 5.2

We rewrite the objective function in Eq. 5.1 replacing $\ell_{2,1}$-norm by F-norm as follow:

$$O' = \mu \left(\left\| \mathbf{X} - \mathbf{H V}^T \right\|_F^2 + \lambda_1 \operatorname{tr}(\mathbf{V}^T \mathbf{L_X V}) \right) + (1-\mu) \left(\left\| \mathbf{U} - \mathbf{V G}^T \right\|_F^2 + \lambda_2 \operatorname{tr}(\mathbf{G}^T \mathbf{L_U G}) \right) \tag{5.24}$$

and use $F_{\mathbf{V}}$, $F_{\mathbf{H}}$ and $F_{\mathbf{G}}$ to denote the part of O' which is only relevant to $\mathbf{V}$, $\mathbf{H}$ and $\mathbf{G}$ respectively. Same to above, because of Definition 4.1, the key step to prove Theorem 5.2 is to find a proper auxiliary function with respect to $\mathbf{V}$, $\mathbf{H}$ and $\mathbf{G}$.

Updating V We focus on updating $\mathbf{V}$ while fixing $\mathbf{H}$ and $\mathbf{G}$, then we rewrite the objective function in (5.24) relevant to V as:

$$F_{\mathbf{V}} = \mu \left(\left\| \mathbf{X} - \mathbf{H V}^T \right\|_F^2 + \lambda_1 \operatorname{tr}(\mathbf{V}^T \mathbf{L_X V}) \right) + (1-\mu) \left\| \mathbf{U} - \mathbf{V G}^T \right\|_F^2 \tag{5.25}$$

Considering any element v_{ab} in $\mathbf{V}$, it is easy to derive that

$$\begin{aligned} F'_{\mathbf{V}}(v_{ab}^{(t)}) = \frac{\partial F_{\mathbf{V}}}{\partial v_{ab}} =& \mu[(-2\mathbf{X}^T\mathbf{H} + 2\mathbf{V}\mathbf{H}^T\mathbf{H})_{ab} + (2\lambda_1 \mathbf{L_X V})_{ab}] \\ &+ (1-\mu)[(-2\mathbf{U G})_{ab} + (2\mathbf{V G}^T\mathbf{G})_{ab}], \end{aligned}$$

and

$$F''_{\mathbf{V}}(v_{ab}^{(t)}) = 2\mu(\mathbf{H}^T\mathbf{H})_{bb} + 2\mu(\lambda_1 \mathbf{L_X})_{aa} + 2(1-\mu)(\mathbf{G}^T\mathbf{G})_{bb}.$$

Since $F''_{\mathbf{V}}(v_{ab}^{(t)}) \geq 0$, the element-wise updating rule of Eq. 5.9 is sufficient to be proved as non-increasing. We continue the standard convergence proof with the auxiliary function

in the following.

Lemma 5.4. *Function*

$$J(v, v_{ab}^{(t)}) = F_{\mathbf{V}}(v_{ab}^{(t)}) + F'_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) + \frac{(\mathbf{V}\mathbf{V}^T\mathbf{A}' + \mu\lambda_1\mathbf{D}_{\mathbf{V}}\mathbf{V})_{ab}}{v_{ab}^{(t)}}(v - v_{ab}^{(t)})^2 \tag{5.26}$$

is an auxiliary function for $F_{\mathbf{V}}$.

Proof. $J(v, v) = F_{\mathbf{V}}(v)$ is obvious, then we only need to prove $J(v, v_{ab}^{(t)}) \geq F_{\mathbf{V}}(v)$. To do this, we have the Taylor series expansion of $F_{\mathbf{V}}(v)$:

$$F_{\mathbf{V}}(v) = F_{\mathbf{V}}(v_{ab}^{(t)}) + F'_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) + \frac{1}{2}F''_{\mathbf{V}}(v_{ab}^{(t)})(v - v_{ab}^{(t)})^2. \tag{5.27}$$

Thus, by Eq. 5.25 and Eq. 5.26, $J(v, v_{ab}^{(t)}) \geq (F_{\mathbf{V}})_{ab}(v)$ is equivalent to:

$$\frac{(\mathbf{V}\mathbf{V}^T\mathbf{A}' + \mu\lambda_1\mathbf{D}_{\mathbf{V}}\mathbf{V})_{ab}}{v_{ab}^{(t)}} \geq \frac{1}{2}F''_{\mathbf{V}}(v_{ab}^{(t)}) \tag{5.28}$$

Since we have:

$$(\mathbf{V}\mathbf{V}^T\mathbf{A}')_{ab} = \Big(\mathbf{V}\mathbf{V}^T((\mu(\mathbf{X}^T\mathbf{H} + \lambda_1\mathbf{W}_{\mathbf{X}}\mathbf{V})) + (1-\mu)(\mathbf{U}\mathbf{G}))\Big)_{ab}$$

Then, we have:

$$\mu(\mathbf{V}\mathbf{V}^T\mathbf{X}^T\mathbf{H})_{ab} = \mu(\mathbf{V}\mathbf{H}^T\mathbf{H})_{ab} = \mu\sum_{i=1}^{k} v_{ai}^{(t)}(\mathbf{H}^T\mathbf{H})_{ib} \geq \mu v_{ab}^{(t)}(\mathbf{H}^T\mathbf{H})_{bb},$$

and

$$(1-\mu)(\mathbf{V}\mathbf{V}^T\mathbf{U}\mathbf{G})_{ab} = (1-\mu)(\mathbf{V}\mathbf{G}^T\mathbf{G})_{ab} = (1-\mu)\sum_{i=1}^{k} v_{ai}^{(t)}(\mathbf{G}^T\mathbf{G})_{ib} \geq (1-\mu)v_{ab}^{(t)}(\mathbf{G}^T\mathbf{G})_{bb},$$

and

$$\mu\lambda_1(\mathbf{D}_{\mathbf{V}}\mathbf{V})_{ab} = \mu\lambda_1\sum_{j=1}^{n}\mathbf{D}_{\mathbf{V}aj}v_{jb}^{(t)} \geq \mu\lambda_1\mathbf{D}_{\mathbf{V}aa}v_{ab}^{(t)} \geq \mu\lambda_1(\mathbf{D}_{\mathbf{V}} - \mathbf{W}_{\mathbf{X}})_{aa}v_{ab}^{(t)} = \mu\lambda_1\mathbf{L}_{\mathbf{X}aa}v_{ab}^{(t)}.$$

Thus, Eq. 5.28 holds and $J(v, v_{ab}^{(t)}) \geq F_{\mathbf{V}}(v)$. □

We can derive the following updating rule for v_{ab} by replacing auxiliary function in Eq. 4.28 with Eq. 5.26:

$$v_{ab}^{(t+1)} = v_{ab}^{(t)} - v_{ab}^{(t)} \frac{F'_{\mathbf{V}}(v_{ab}^{(t)})}{(\mathbf{V}\mathbf{V}^T\mathbf{A} + \mu\lambda_1\mathbf{D}_{\mathbf{V}}\mathbf{V})_{ab}} = v_{ab}^{(t)} \frac{(\mathbf{A} + \mu\lambda_1\mathbf{V}\mathbf{V}^T\mathbf{D}_{\mathbf{V}}\mathbf{V})_{ab}}{(\mathbf{V}\mathbf{V}^T\mathbf{A} + \mu\lambda_1\mathbf{D}_{\mathbf{V}}\mathbf{V})_{ab}}$$

Since Eq. 5.26 is an auxiliary function, $F_{\mathbf{V}}$ is non-increasing under this updating rule.

Updating H We focus on updating $\mathbf{H}$ while fixing $\mathbf{V}$ and $\mathbf{G}$. We rewrite the objective function relevant to H as:

$$F_{\mathbf{H}} = \mu \left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T \right\|_F^2 \tag{5.29}$$

Considering any element h_{ab} in $\mathbf{H}$, we define the following lemma:

Lemma 5.5. *Fucntion*

$$J(h, h_{ab}^{(t)}) = F_{\mathbf{H}}(h_{ab}^{(t)}) + F'_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)}) + \frac{(\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab}}{h_{ab}^{(t)}}(h - h_{ab}^{(t)})^2 \tag{5.30}$$

is an auxiliary function for $F_{\mathbf{H}}$.

Proof. $J(h, h) = F_{\mathbf{H}}(h)$ is obvious. Then we prove $J(h, h_{ab}^{(t)}) \geq F_{\mathbf{H}}(h)$. Similar to the proof of lemma 1, we first write the Taylor series expansion of $F_{\mathbf{H}}(h)$ as:

$$F_{\mathbf{H}}(h) = F_{\mathbf{H}}(h_{ab}^{(t)}) + F'_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)}) + \frac{1}{2}F''_{\mathbf{H}}(h_{ab}^{(t)})(h - h_{ab}^{(t)})^2 \tag{5.31}$$

where $F'_{\mathbf{H}}(h_{ab}^{(t)}) = (-2\mathbf{X}\mathbf{V} + 2\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab}$, and $F''_{\mathbf{H}}(h_{ab}^{(t)}) = 2(\mathbf{V}^T\mathbf{V})_{ab}$. By comparing Eq. 5.30 with Eq. 5.31, we find that $J(h, h_{ab}^{(t)}) \geq F_{\mathbf{H}}(h)$ is equivalent to

$$\frac{(\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab}}{h_{ab}^{(t)}} \geq (\mathbf{V}^T\mathbf{V})_{ab} \tag{5.32}$$

We have $(\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab} = \sum_{i=1}^{k} h_{ai}^{(t)}(\mathbf{V}^T\mathbf{V})_{ib} \geq h_{ab}^{(t)}(\mathbf{V}^T\mathbf{V})_{ab}$. Thus, the matrix inequality in Eq. 5.32 holds and $J(h, h_{ab}^{(t)}) \geq (F_{\mathbf{H}})_{ab}(h)$. □

We can now demonstrate the convergence of Eq. 5.10 under the following updating

rule by replacing Eq. 4.28 with Eq. 5.30:

$$h_{ab}^{(t+1)} = h_{ab}^{(t)} - h_{ab}^{(t)} \frac{F'_{\mathbf{H}}(h_{ab}^{(t)})}{(\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab}} = h_{ab}^{(t)} \frac{(\mathbf{X}\mathbf{V})_{ab}}{(\mathbf{H}\mathbf{V}^T\mathbf{V})_{ab}}$$

Since Eq. 5.30 is an auxiliary function, $F_{\mathbf{H}}$ is non-increasing under this replace rule.

Updating G We focus on updating $\mathbf{G}$ while fixing $\mathbf{V}$ and $\mathbf{H}$. We rewrite the objective function relevant to $\mathbf{G}$ as

$$F_{\mathbf{G}} = \left\|\mathbf{U} - \mathbf{V}\mathbf{G}^T\right\|_F^2 + \lambda_2 \operatorname{tr}(\mathbf{G}^T\mathbf{L}_{\mathbf{U}}\mathbf{G}) \tag{5.33}$$

Considering any element g_{ab} in $\mathbf{G}$, we define the following lemma:

Lemma 5.6. *Function*

$$J(g, g_{ab}^{(t)}) = F_{\mathbf{G}}(g_{ab}^{(t)}) + F'_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)}) + \frac{(\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\mathbf{D}_{\mathbf{G}}\mathbf{G})_{ab}}{g_{ab}^{(t)}}(g - g_{ab}^{(t)})^2 \tag{5.34}$$

is an auxiliary function for $F_{\mathbf{G}}$.

Proof. $J(g, g) = F_{\mathbf{G}}(g)$ is obvious. Then we only need to prove $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$. We compare the Taylor series expansion of $F_{\mathbf{G}}(g)$ as:

$$F_{\mathbf{G}}(g) = F_{\mathbf{G}}(g_{ab}^{(t)}) + F'_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)}) + \frac{1}{2}F''_{\mathbf{G}}(g_{ab}^{(t)})(g - g_{ab}^{(t)})^2 \tag{5.35}$$

where $F'_{\mathbf{G}}(g_{ab}^{(t)}) = (-2\mathbf{U}^T\mathbf{V} + 2\lambda_2\mathbf{L}_{\mathbf{U}}\mathbf{G})_{ab} + 2(\mathbf{G}\mathbf{V}^T\mathbf{V})_{ab}$ and $F''_{\mathbf{G}}(g_{ab}^{(t)}) = 2\lambda_2\mathbf{L}_{\mathbf{U}aa} + 2(\mathbf{V}^T\mathbf{V})_{bb}$. By comparing Eq. 5.34 with Eq. 5.35, we find that $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$ is equivalent to

$$\frac{(\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2\mathbf{D}_{\mathbf{G}}\mathbf{G})_{ab}}{g_{ab}^{(t)}} \geq \frac{1}{2}(F_{\mathbf{G}})''_{ab} = \lambda_2\mathbf{L}_{\mathbf{U}aa} + (\mathbf{V}^T\mathbf{V})_{bb} \tag{5.36}$$

We have

$$(\mathbf{G}\mathbf{V}^T\mathbf{V})_{ab} = \sum_{i=1}^{k} g_{ai}^{(t)}(\mathbf{V}^T\mathbf{V})_{ib} \geq g_{ab}^{(t)}(\mathbf{V}^T\mathbf{V})_{bb}$$

and

$$(\lambda_2 \mathbf{D_G G})_{ab} = \lambda_2 \sum_{p=1}^{l} \mathbf{D_G}_{ap} g_{pb}^{(t)} \geq \lambda_2 \mathbf{D_G}_{aa} g_{ab}^{(t)} \geq \lambda_2 (\mathbf{D_G} - \mathbf{W_U})_{aa} g_{ab}^{(t)} = \lambda_2 (\mathbf{L_U})_{aa} g_{ab}^{(t)}$$

Thus, Eq. 5.36 holds and $J(g, g_{ab}^{(t)}) \geq F_{\mathbf{G}}(g)$. □

We can now replace Eq. 4.28 by $J(g, g_{ab}^{(t)})$ in Eq. 5.34 to derive the update rule:

$$g_{ab}^{(t+1)} = g_{ab}^{(t)} - g_{ab}^{(t)} \frac{F'_{\mathbf{G}}(g_{ab}^{(t)})}{\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2 \mathbf{D_G G})_{ab}} = g_{ab}^{(t)} \frac{(\mathbf{U}^T\mathbf{V} + \lambda_2 \mathbf{W_U G})_{ab}}{(\mathbf{G}\mathbf{V}^T\mathbf{V} + \lambda_2 \mathbf{D_G G})_{ab}}$$

since Eq. 5.34 is an auxiliary function, $F_{\mathbf{G}}$ is non-increasing under this updating rule.

5.4.3 Optimization with Augmented Lagrangian Method

A novel method for solving the ℓ_2,1-norm NMF optimization problem, called Augmented Lagrangian Method [76,192], comparing to the method of Lagrangian Multiplier, converts the original inequality constraints to equality constraints with slack variables and adds an additional quadratic penalty term for each equality constraint to the end of the objective function with a big penalty coefficient. For example, to solve following constrained optimisation problem:

$$\min f(X)$$
$$\text{subject to } h(X) = 0$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$. The standard Lagrangian function can be defined as:

$$\mathcal{L}(X, a) = f(X) + ah(X)$$

where a is the Lagrangian multiplier.

While the Lagrangian function of penalty term based method can be written as:

$$\mathcal{L}(X, \alpha) = f(X) + \alpha p(X),\ p(X) = h^T h$$

where α is the penalty coefficient to regulate the penalty for the violation of equality constraint in $h(X)$ and $\alpha p(X)$ is the penalty term. The standard form of the augmented Lagrangian function is:

$$\mathcal{L}(X, a) = f(X) + \langle \mathbf{Y}, h(X) \rangle + \frac{\alpha}{2} \|h(X)\|_F^2 \tag{5.37}$$

for the optimal Lagrangian multiplier $\mathbf{Y}$, where $\langle \mathbf{A}, \mathbf{B} \rangle = tr(\mathbf{A}^T \mathbf{B})$ and α is the penalty coefficient here.

Now back to our problem. Before we define our augmented Lagrangian function, we introduce some auxiliary variables $\mathbf{P_1} = \mathbf{X} - \mathbf{H}\mathbf{V}^T$, $\mathbf{P_2} = \mathbf{U} - \mathbf{V}\mathbf{G}^T$ and slack variables $\mathbf{Q_1} = \mathbf{V}$, $\mathbf{Q_2} = \mathbf{G}$ to form the necessary equality constraints. Then, the objective function in Eq. 5.1 can be rewritten to fit into the augmented Lagrangian method as follow:

$$\begin{aligned} f_{ALM} = & \underset{\mathbf{P}_1, \mathbf{P}_2, \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{H}, \mathbf{V}, \mathbf{G}}{\arg\min} \mu(\|\mathbf{P}_1\|_{2,1} + \lambda_1 tr(\mathbf{Q}_1^T \mathbf{L}_\mathbf{X} \mathbf{V})) + (1-\mu)(||\mathbf{P}_2||_{2,1} + \lambda_2 tr(\mathbf{Q}_2^T \mathbf{L}_\mathbf{U} \mathbf{G})) \\ & \text{s.t. } \mathbf{P}_1 = \mathbf{X} - \mathbf{H}\mathbf{V}_T, \mathbf{P}_2 = \mathbf{U} - \mathbf{V}\mathbf{G}_T, \mathbf{Q}_1 = \mathbf{V}, \mathbf{Q}_2 = \mathbf{G}, \mathbf{V}^T\mathbf{V} = \mathbf{I} \text{ and } \mathbf{Q}_1, \mathbf{Q}_2 \geq 0 \end{aligned} \tag{5.38}$$

Here, we relax the non-negative constraint of $\mathbf{H}$ since it can be derived from $\mathbf{X}$ is non-negative and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. Then, For the above optimisation problem, we define the augmented Lagrangian function:

$$\begin{aligned} \mathcal{L}_{ALM} = & \underset{\mathbf{P}_1, \mathbf{P}_2, \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{H}, \mathbf{V}, \mathbf{G}}{\arg\min} \mu(||\mathbf{P}_1||_{2,1} + \lambda_1 tr(\mathbf{Q}_1^T \mathbf{L}_\mathbf{X} \mathbf{V})) + (1-\mu)(||\mathbf{P}_2||_{2,1} + \lambda_2 tr(\mathbf{Q}_2^T \mathbf{L}_\mathbf{U} \mathbf{G})) \\ & + \left\langle \mathbf{Y}_1, \mathbf{X} - \mathbf{H}\mathbf{V}^T - \mathbf{P}_1 \right\rangle + \left\langle \mathbf{Y}_2, \mathbf{U} - \mathbf{V}\mathbf{G}^T - \mathbf{P}_2 \right\rangle + \langle \mathbf{Y}_3, (\mathbf{Q}_1 - \mathbf{V}) \rangle + \langle \mathbf{Y}_4, \mathbf{Q}_2 - \mathbf{G} \rangle \\ & + \frac{\alpha}{2}(\left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T - \mathbf{P}_1 \right\|_F^2 + \|\mathbf{Q}_1 - \mathbf{V}\|_F^2 + \left\| \mathbf{U} - \mathbf{V}\mathbf{G}^T - \mathbf{P}_2 \right\|_F^2 + \|\mathbf{Q}_2 - \mathbf{G}\|_F^2) \\ & \text{s.t. } \mathbf{V}^T\mathbf{V} = \mathbf{I} \text{ and } \mathbf{Q}_1, \mathbf{Q}_2 \geq 0 \end{aligned} \tag{5.39}$$

Here, matrices $\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3$ and $\mathbf{Y}_4$ are the Lagrangian multipliers and α is the penalty coefficient. We alternatively update each variable and multiplier while fixing others with following rules:

$$\mathbf{P}_{1i} = \begin{cases} \mathcal{B}_i \cdot (1 - \frac{\mu}{\alpha} \cdot \frac{1}{\|\mathcal{B}_i\|_2}), & \frac{\mu}{\alpha} < \|\mathcal{B}_i\|_2 \\ 0, & \text{otherwise} \end{cases} \tag{5.40}$$

$$\mathbf{P}_{2i} = \begin{cases} \mathcal{C}_p \cdot (1 - \frac{1-\mu}{\alpha} \cdot \frac{1}{\|\mathcal{C}_p\|_2}), & \frac{1-\mu}{\alpha} < \|\mathcal{C}_p\|_2 \\ 0, & \text{otherwise} \end{cases} \tag{5.41}$$

$$\mathbf{Q}_{1ij} = \max(\mathcal{E}_{ij}, 0) \tag{5.42}$$

$$\mathbf{Q}_{2ij} = \max(\mathcal{F}_{ij}, 0) \tag{5.43}$$

$$\mathbf{H} = \left(\mathbf{X} - \mathbf{P}_1 + \frac{1}{\alpha}\mathbf{Y}_1\right)\mathbf{V} \tag{5.44}$$

$$\mathbf{V} = \mathcal{P}\mathcal{Q}^T \tag{5.45}$$

$$\mathbf{G} = \frac{1}{2}\left((\mathbf{U}^T - \mathbf{P}_2^T + \frac{1}{\alpha}\mathbf{Y}_2^T)\mathbf{V} + \mathbf{Q}_2 + \frac{1}{\alpha}\mathbf{Y}_4 + \frac{(1-\mu)\lambda_2}{\alpha}\mathbf{L}_\mathbf{U}\mathbf{Q}_2\right) \tag{5.46}$$

where $\mathcal{B}_i$ is the $i-th$ column of matrix $\mathcal{B} = \mathbf{X} - \mathbf{H}\mathbf{V}^T + \frac{1}{\alpha}\mathbf{Y}_1$; $\mathcal{C}_p$ is the $p-th$ column of matrix $\mathcal{C} = \mathbf{U} - \mathbf{V}\mathbf{G}^T + \frac{1}{\alpha}\mathbf{Y}_2$; $\mathcal{E} = \mathbf{V} - \frac{1}{\alpha}\mathbf{Y}_3 - \frac{\lambda_1}{\alpha}\mathbf{L}_\mathbf{X}\mathbf{V}$; $\mathcal{F} = \mathbf{G} - \frac{1}{\alpha}\mathbf{Y}_4 - \frac{\lambda_2}{\alpha}\mathbf{L}_\mathbf{U}\mathbf{G}$; and $\mathcal{P}, \mathcal{Q}$ are left and right singular matrices of the SVD of $\mathcal{S} = \mathbf{Q}_1 + \frac{1}{\alpha}\mathbf{Y}_3 - \frac{\mu\lambda_1}{\alpha}\mathbf{L}_\mathbf{X}\mathbf{Q}_1 + (\mathbf{X} - \mathbf{P}_1 + \frac{1}{\alpha}\mathbf{Y}_1)^T\mathbf{H} + (\mathbf{U} - \mathbf{P}_2 + \frac{1}{\alpha}\mathbf{Y}_2)\mathbf{G} \in \mathbb{R}^{n\times k}$.

5.4.4 Computation of Augmented Lagrangian Multiplier Method

In the following, we elaborate the computation of each variable when fixing others extracting sub-object function with respect to variables.

Updating $\mathbf{P}_1$ We rewrite the augmented Lagrangian function in Eq. 5.39 relevant to $\mathbf{P}_1$ in brief as:

$$\mathcal{L}_{\mathbf{P}_1} = \arg\min_{\mathbf{P}_1} \frac{1}{2}\left\|\mathbf{X} - \mathbf{H}\mathbf{V}^T - \mathbf{P}_1 + \frac{1}{\alpha}\mathbf{Y}_1\right\|_F^2 + \frac{\mu}{\alpha}\|\mathbf{P}_1\|_{2,1} \tag{5.47}$$

Let $\mathcal{B} = \mathbf{X} - \mathbf{H}\mathbf{V}^T + \frac{1}{\alpha}\mathbf{Y}_1$ herein, Eq. 5.47 can be simplified to $\arg\min_{\mathbf{P}_1} \frac{1}{2}\|\mathbf{P}_1 - \mathcal{B}\|_F^2 + \frac{\mu}{\alpha}\|\mathbf{P}_1\|_{2,1}$. Now, We can use the Proposition 1 presented in [188] and Lemma 3.1 proved in [76] to compute $\mathbf{P}_1$ in Eq. 5.40.

Proof. Eq. 5.47 is equivalent to

$$\mathcal{L}_{\mathbf{P}_1} = \sum_{i=1}^{n} \arg\min_{\mathbf{P}_{1i}} \frac{1}{2} \|\mathbf{P}_{1i} - \mathcal{B}_i\|_F^2 + \frac{\mu}{\alpha} \|\mathbf{P}_{1i}\|_2 \tag{5.48}$$

where, $\|\mathbf{P}_{1i}\|_2 = \sqrt{\mathbf{P}_{1i}^T \mathbf{P}_{1i}}$. According to the element-wise derivation rule, we have $\frac{\partial \|\mathbf{P}_{1i}\|_2}{\partial \mathbf{P}_{1i}} = \begin{cases} \mathbf{P}_{1i} / \sqrt{\mathbf{P}_{1i}^T \mathbf{P}_{1i}}, & \mathbf{P}_{1i} \neq 0 \\ \gamma, & \mathbf{P}_{1i} = 0 \end{cases}$, γ is a subgradient vector for those non-differentiable kinks or cusps and $\|\mathbf{fl}\|_2 \leq 1$. Taking Eq. 5.48 differentiate with respect to $\mathbf{P}_{1i}$, we have

$$\frac{\partial \mathcal{L}_{\mathbf{P}_1}}{\partial \mathbf{P}_{1i}} = \begin{cases} \mathbf{P}_{1i} - \mathcal{B}_i + \frac{\mu}{\alpha} \cdot \mathbf{P}_{1i} / \sqrt{\mathbf{P}_{1i}^T \mathbf{P}_{1i}}, & \mathbf{P}_{1i} \neq 0 \\ \frac{1}{\alpha} \mathbf{fl} - \mathcal{B}_i, & \mathbf{P}_{1i} = 0 \end{cases}$$

Let $\frac{\partial \mathcal{L}_{\mathbf{P}_1}}{\partial \mathbf{P}_{1i}} = 0$, if $\mathbf{P}_{1i} = 0$, we have $\frac{\mu}{\alpha} \mathbf{fl} - \mathcal{B}_i = 0$, which implies $\frac{\mu}{\alpha} \geq \|\mathcal{B}_i\|_2$; if $\mathbf{P}_{1i} \neq 0$, we have $\mathbf{P}_{1i} - \mathcal{B}_i + \frac{1}{\alpha} \cdot \mathbf{P}_{1i} / \sqrt{\mathbf{P}_{1i}^T \mathbf{P}_{1i}} = 0$ that is equivalent to $\mathbf{P}_{1i} = \frac{\|\mathbf{P}_{1i}\|_2}{\|\mathbf{P}_{1i}\|_2 + 1/\alpha} \cdot \mathcal{B}_i$. Let $\delta = \frac{\|\mathbf{P}_{1i}\|_2}{\|\mathbf{P}_{1i}\|_2 + 1/\alpha}$, we can thereby reach $\mathbf{P}_{1i} = \mathcal{B}_i - \frac{1}{\alpha} \cdot \frac{\delta \mathcal{B}_i}{\delta \sqrt{\mathcal{B}_i^T \mathcal{B}_i}} = \mathcal{B}_i (1 - \frac{\mu}{\alpha} \cdot \frac{1}{\|\mathcal{B}_i\|_2})$. □

Updating $\mathbf{P}_2$ We rewrite Eq. 5.39 relevant to $\mathbf{P}_2$ in brief as:

$$\mathcal{L}_{\mathbf{P}_2} = \arg\min_{\mathbf{P}_2} \frac{1}{2} \left\| \mathbf{U} - \mathbf{V}\mathbf{G}^T - \mathbf{P}_2 + \frac{1}{\alpha} \mathbf{Y}_2 \right\|_F^2 + \frac{1-\mu}{\alpha} \|\mathbf{P}_2\|_{2,1} \tag{5.49}$$

Let $\mathcal{C} = \mathbf{U} - \mathbf{V}\mathbf{G}^T + \frac{1}{\alpha} \mathbf{Y}_2$ herein, Eq. 5.50 can be simplified to $\arg\min_{\mathbf{P}_2} \frac{1}{2} \|\mathbf{P}_2 - \mathcal{C}\|_F^2 + \frac{1-\mu}{\alpha} \|\mathbf{P}_2\|_{2,1}$. Thus, referring to the above derivation of $\mathbf{P}_{1i}$, we would have the updating rule in Eq. 5.41 for $p = 1, 2, \cdots, l$, where $\mathcal{C}_p \in \mathbb{R}^{n \times 1}$ is the $\mathcal{C}_p$ is the $p - th$ column of matrix $\mathcal{C}$.

Updating $\mathbf{H}$ We rewrite Eq. 5.39 relevant to $\mathbf{H}$ in brief as:

$$\mathcal{L}_{\mathbf{H}} = \arg\min_{\mathbf{H}} \frac{\alpha}{2} \left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T - \mathbf{P}_1 + \frac{1}{\alpha} \mathbf{Y}_1 \right\|_F^2 \tag{5.50}$$

Denote $\mathcal{D} = \mathbf{X} - \mathbf{P}_1 + \frac{1}{\alpha} \mathbf{Y}_1$, Eq. 5.50 can be transformed into

$$\mathcal{L}_{\mathbf{H}} = \arg\min \left\| \mathcal{D} - \mathbf{H}\mathbf{V}^T \right\|_F^2 = tr((\mathcal{D} - \mathbf{H}\mathbf{V}^T)^T (\mathcal{D} - \mathbf{H}\mathbf{V}^T))$$

To minimise $\mathcal{L}_{\mathbf{H}}$, we let $\frac{\partial \mathcal{L}_{\mathbf{H}}}{\partial \mathbf{H}_t} = -2\mathcal{D}\mathbf{V}_t + 2\mathbf{H}_t\mathbf{V}_t^T\mathbf{V}_t = 0$. Considering our input constraint $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, we have the update rule $\mathbf{H} = \mathcal{D}\mathbf{V}$ for $\mathbf{H}$.

Updating $\mathbf{Q}_1$ We rewrite the augmented Lagrangian function relevant to $\mathbf{Q}_1$ as:

$$\mathcal{L}_{\mathbf{Q}_1} = \arg\min_{\mathbf{Q}_1 \geq 0} \lambda_1 tr(\mathbf{Q}_1^T \mathbf{L}_{\mathbf{X}} \mathbf{V}) + tr(\mathbf{Y}_3^T \cdot (\mathbf{Q}_1 - \mathbf{V})) + \frac{\alpha}{2} \|\mathbf{Q}_1 - \mathbf{V}\|_F^2 \tag{5.51}$$

By removing the irrelevant terms, we get the following equivalent equation:

$$\mathcal{L}_{\mathbf{Q}_1} = \arg\min_{\mathbf{Q}_1 \geq 0} tr\left(\frac{2\lambda_1}{\alpha} (\mathbf{Q}_1^T \mathbf{L}_{\mathbf{X}} \mathbf{V}) + \mathbf{Q}_1\mathbf{Q}_1^T - 2\mathbf{Q}_1 (\mathbf{V} - \frac{1}{\alpha}\mathbf{Y}_3)^T \right)$$

Let $\mathcal{E} = \mathbf{V}_t - \frac{1}{\alpha}\mathbf{Y}_3 - \frac{\lambda_1}{\alpha}\mathbf{L}\_\mathbf{X}\mathbf{V}$, the above equation is equivalent to $\arg\min_{\mathbf{Q}_1 \geq 0} \|\mathbf{Q}_1 - \mathcal{E}\|_F^2$. To satisfy the constraint $\mathbf{Q}_1 \geq 0$, we derive the element-wise updating rule of $\mathbf{Q}_1$ as $\mathbf{Q}_{1ij} = \max(\mathcal{E}_{ij}, 0)$.

Updating $\mathbf{Q}_2$ We rewrite the augmented Lagrangian function relevant to $\mathbf{Q}_2$ as:

$$\mathcal{L}_{\mathbf{Q}_2} = \arg\min_{\mathbf{Q}_2 \geq 0} \lambda_2 tr(\mathbf{Q}_2^T \mathbf{L}_{\mathbf{U}} \mathbf{G}) + tr(\mathbf{Y}_4^T \centerdot (\mathbf{Q}_2 - \mathbf{G})) + \frac{\alpha}{2} \|\mathbf{Q}_2 - \mathbf{G}\|_F^2 \tag{5.52}$$

Denote $\mathcal{F} = \mathbf{G} - \frac{1}{\alpha}\mathbf{Y}_4 - \frac{\lambda_2}{\alpha}\mathbf{L}\_\mathbf{U}\mathbf{G}$, similar to above, we can derive the element-wise updating rule as $\mathbf{Q}_{2ij} = \max(\mathcal{F}_{ij}, 0)$ for $\mathbf{Q}_2$.

Updating $\mathbf{V}$ Eq. 5.39 with respect to $\mathbf{V}$ yield the equation below:

$$\begin{aligned}
\mathcal{L}_{\mathbf{V}} = \arg\min_{\mathbf{V}^T\mathbf{V}=\mathbf{I}} & \mu \left(\lambda_1 tr(\mathbf{Q}_1^T \mathbf{L}_{\mathbf{X}} \mathbf{V}) \right) + tr(\mathbf{Y}_1^T \cdot (\mathbf{X} - \mathbf{H}\mathbf{V}^T - \mathbf{P}_1)) \\
& + tr(\mathbf{Y}_2^T \cdot (\mathbf{U} - \mathbf{V}\mathbf{G}^T - \mathbf{P}_2)) + tr(\mathbf{Y}_3^T \cdot (\mathbf{Q}_1 - \mathbf{V})) \\
& + \frac{\alpha}{2} \left(\left\| \mathbf{X} - \mathbf{H}\mathbf{V}^T - \mathbf{P}_1 \right\|_F^2 + \left\| \mathbf{U} - \mathbf{V}\mathbf{G}^T - \mathbf{P}_2 \right\|_F^2 + \|\mathbf{Q}_1 - \mathbf{V}\|_F^2 \right)
\end{aligned} \tag{5.53}$$

We denote $\mathcal{S} = \mathbf{Q}_1 + \frac{1}{\alpha}\mathbf{Y}_3 - \frac{\mu\lambda_1}{\alpha}\mathbf{L}_{\mathbf{X}}\mathbf{Q}_1 + (\mathbf{X} - \mathbf{P}_1 + \frac{1}{\alpha}\mathbf{Y}_1)^T\mathbf{H} + (\mathbf{U} - \mathbf{P}_2 + \frac{1}{\alpha}\mathbf{Y}_2)\mathbf{G}$ and rewrite Eq. 5.53 in the following brief form:

$$\mathcal{L}_{\mathbf{V}} = \arg\min_{\mathbf{V}^T\mathbf{V}=\mathbf{I}} \|\mathbf{V} - \mathcal{S}\|_F^2 \tag{5.54}$$

Notice that when we take the derivative of $\mathbf{V}$ on eq.(38), it is equivalent to Eq. 5.53 by removing irrelevant terms. With the cyclic property and transpose property of the trace, we have:

$$\|\mathbf{V}-\mathcal{S}\|_F^2 = tr((\mathbf{V}-\mathcal{S})^T(\mathbf{V}-\mathcal{S}) = tr(\mathbf{V}^T\mathbf{V}) - 2tr(\mathbf{V}^T\mathcal{S}) + tr(\mathcal{S}^T\mathcal{S})$$

Then, the optimisation problem in Eq. 5.54 is equivalent to

$$\mathcal{L}_{\mathbf{V}} \Rightarrow \max_{\mathbf{V}_t^T\mathbf{V}_t=\mathbf{I}} \langle \mathbf{V}_t, \mathcal{S}\rangle \tag{5.55}$$

To solve this problem with Lagrangian Multiplier Method, we denote the Lagrangian as:

$$\mathcal{L}' = tr(\mathbf{V}_t^T\mathcal{S}) + tr(\Lambda(\mathbf{V}_t^T\mathbf{V}_t - \mathbf{I}))$$

where, Λ is the Lagrangian multiplier matrix. Let $\frac{\partial \mathcal{L}'}{\partial \mathbf{V}} = \mathcal{S} + \mathbf{V}(\Lambda + \Lambda^T) = 0$, we have $\mathcal{S} = \mathbf{V}(\Lambda + \Lambda^T)$ and $\mathbf{V}^T\mathcal{S} = \Lambda + \Lambda^T$. Then we get $\mathcal{S} = \mathbf{V}\mathbf{V}^T\mathcal{S}$, so that we have $\mathbf{V}\mathbf{V}^T = \mathbf{I}$ hold. Meanwhile, we know that $\Lambda + \Lambda^T$ is a symmetric matrix and therefore we have

$$\mathcal{S}^T\mathbf{V} = \mathbf{V}^T\mathcal{S} \tag{5.56}$$

Since $n \gg k$ herein, according to the thin singular-value decomposition (reduced SVD), we have $\mathcal{S} = \mathcal{P}\Sigma\mathcal{Q}^T$, where $\mathcal{P} \in \mathbb{R}^n \times k$, $\mathcal{Q} \in \mathbb{R}^k \times k$ and $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_k) \in \mathbb{R}^{k\times k}$.

As we mentioned in Chapter 2, reduced SVD only calculate k columns of $\mathcal{P}$ corresponding to rows of $\mathcal{Q}$, where columns of $\mathcal{P}$ are orthogonal vectors and $\mathcal{P}$ satisfies $\mathcal{P}^T\mathcal{P} = \mathbf{I}$, while $\mathcal{Q}$ is an orthogonal matrix satisfying $\mathcal{Q}^T\mathcal{Q} = \mathcal{Q}\mathcal{Q}^T = \mathbf{I}$. Now with (5.56), we have:

$$\mathbf{V}^T\mathcal{P}\Sigma\mathcal{Q}^T = \mathcal{Q}\Sigma^T\mathcal{P}^T\mathbf{V} = \mathcal{Q}\Sigma\mathcal{P}^T\mathbf{V}$$

Comparing the two sides, we have $\mathbf{V}_t^T\mathcal{P} = \mathcal{Q} \Rightarrow \mathbf{V}_t\mathbf{V}_t^T\mathcal{P} = \mathbf{V}_t\mathcal{Q}$ which finally leads to the updating rule $\mathbf{V}_t = \mathcal{P}\mathcal{Q}^T$ in Eq. 5.45. Besides, the property $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ holds obviously.

Updating G Eq. 5.39 with respect to **G** is as follow:

$$\mathcal{L}_{\mathbf{G}} = \arg\min_{\mathbf{G}} (1-\mu)\,\lambda_2 tr(\mathbf{Q}_2^T \mathbf{L}_{\mathbf{U}} \mathbf{G}) + tr(\mathbf{Y}_2^T \cdot (\mathbf{U} - \mathbf{V}\mathbf{G}^T - \mathbf{P}_2)) + tr(\mathbf{Y}_4^T \cdot (\mathbf{Q}_2 - \mathbf{G})) + \frac{\alpha}{2}(\left\|\mathbf{U} - \mathbf{V}\mathbf{G}^T - \mathbf{P}_2\right\|_F^2 + \|\mathbf{Q}_2 - \mathbf{G}\|_F^2) \tag{5.57}$$

Combining the constraint $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, we can reconfigure the above function to

$$\arg\min_{\mathbf{G}} (1-\mu)\,\lambda_2 tr(\mathbf{Q}_2^T \mathbf{L}_{\mathbf{U}} \mathbf{G}) + tr(\mathbf{Y}_2^T \cdot (\mathbf{U} - \mathbf{V}\mathbf{G}^T - \mathbf{P}_2)) + tr(\mathbf{Y}_4^T \cdot (\mathbf{Q}_2 - \mathbf{G})) + \frac{\alpha}{2}(\left\|\mathbf{V}^T(\mathbf{U} - \mathbf{P}_2) - \mathbf{G}^T\right\|_F^2 + \|\mathbf{Q}_2 - \mathbf{G}\|_F^2) \tag{5.58}$$

By using Lagrangian multiplier method on Eq. 5.58, we have the updating rule $\mathbf{G} = \frac{1}{2}\left((\mathbf{U} - \mathbf{P}_2 + \frac{1}{\alpha}\mathbf{Y}_2)^T\mathbf{V} + \mathbf{Q}_2 + \frac{1}{\alpha}\mathbf{Y}_4 + \frac{(1-\mu)\lambda_2}{\alpha}\mathbf{L}_{\mathbf{U}}\mathbf{Q}_2\right)$ in Eq. 5.46 for **G**.

In our implementation, we take turn updating variables in the same order as describe above, followed by the computation of other parameters, including Lagrangian multipliers $\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3, \mathbf{Y}_4$ and penalty coefficient α in one iteration. According to [34](page 232-236), we have the following updating rules:

$$\mathbf{Y}_1 = \mathbf{Y}_1 + \alpha(\mathbf{X} - \mathbf{H}\mathbf{V}^T - \mathbf{P}_1) \tag{5.59}$$

$$\mathbf{Y}_2 = \mathbf{Y}_2 + \alpha(\mathbf{U} - \mathbf{V}\mathbf{G}^T - \mathbf{P}_2) \tag{5.60}$$

$$\mathbf{Y}_3 = \mathbf{Y}_3 + \alpha(\mathbf{Q}_1 - \mathbf{V}) \tag{5.61}$$

$$\mathbf{Y}_4 = \mathbf{Y}_4 + \alpha(\mathbf{Q}_2 - \mathbf{G}) \tag{5.62}$$

$$\alpha = \kappa\alpha \tag{5.63}$$

where parameter κ controls the convergence speed as a larger value of κ leads to a bigger step size with less precision and greater error of the objective functions value.

5.5 Robust Hierarchical Ensemble Method

Hierarchical clustering approaches developed among the first generation of clustering methods which can be categorised as either bottom-up agglomerative approach or top-down divisive approach. Intuitively, hierarchical structure enhances the interpretability

Algorithm 5.1: Orthonormal NMF with Augmented Lagrangian/Multiplicative Updating Rule

Input: $\mathbf{X}, \mathbf{U}, k, \mu, \lambda_1$ and λ_2, convergence threshold ϵ and maximum allowed iteration *maxiter*

Output: $\mathbf{V}$

Initialise $\mathbf{V}$ and $\mathbf{G}$ with kmeans($\mathbf{X}$) and kmeans($\mathbf{U}$)
Initialise $\mathbf{H} = \mathbf{XV}$
while $\mathcal{J}_t - \mathcal{J}_{t-1} > \epsilon$ *and iterated less than maxiter* **do**
    Update variables in turn with rules in Eq. 5.6-Eq. 5.11 or Eq. 5.40-Eq. 5.46
    Compute $\mathcal{J}$ for objective function with Eq. 5.1 or Eq. 5.38

of the topic model and presents integrated clustering result for input data, especially for those data with highly sophisticated and mutable distribution. Here we propose a new top-down fashion based Robust Hierarchical Ensemble NMF method (RHE) for document clustering using the above Robust CMF objective with the updating algorithm and an adjustable k-ary tree, where all input documents form the root node in the beginning. In this section, we will present details about how we construct the hierarchical structure, including how we determine which node to split in each step, when the construction will be terminated, the pruning strategy, as well as what standard we take to identify and exclude outliers. Generally, in our RHE method, the topics expand hierarchically as a tree structure, in which each node represents a clurster document of related topics.

5.5.1 Hierarchy Construction

Existing works commonly input a predefined number K representing the number of latent topics in whole corpus, corresponding to K clusters of all input documents no matter what topic detection task they applied to, such as, static data or data streams. Despite how to select the value of K for static dataset, it is inappropriate to keep a constant K for a flow of data. However, the difficulty of setting a proper K has not been resolved. Standard NMF algorithm has been applied to document clustering task in [21] in which each element v_{ij} of $\mathbf{V}$ is treated as the degree of document d_i belonging to cluster c_j, and the hard cluster label cl_i of the document d_i is determined by $cl_i = \text{argmax}_j\, v_{ij}$ in the i-th row of $\mathbf{V}$. As a compromise, our method provides flexible stopping criteria for constructing the hierarchical structure, while the most important thing is our method can offer more

Algorithm 5.2: Robust Hierarchical Ensemble Method

Input: $\mathbf{X}, \mathbf{U}, K, k$, threshold θ and maximum trial allowance γ for outlier verification
Output: CL: cluster labels of documents and $\mathbf{H}$

Create a root node with all input documents $\{X_1, \cdots, X_{n_d}\} \in \mathbf{X}$;
while *number of leaf nodes* $< K$ **do**
  $\mathbb{S} \leftarrow$ set of all (new) leaves;
  $k' \leftarrow min(K -$ number of leaf nodes $+ 1, k)$;
  for s *in* $\mathbb{S}$ **do**
    $p = k'$, $\mathbf{s}\_backup = \mathbf{s}$, $\mathbb{D} \leftarrow \{\}$;
    while $p > 1$ **do**
      $i \leftarrow 0$;
      while $i < \gamma$ **do**
        try to split documents in $\mathbf{s}$ into p clusters $\mathbf{L_s} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots\}$ with proposed Robust CMF;
        if $|\mathbf{L_s}| < p$ **then**
          $p = |\mathbf{L_s}|$, **break**;
        end
        if $\mathbb{D} \neq \{\}$ ***and*** $\mathbb{D}\{end\}$ *cannot be verified as outliers* **then**
          $\mathbf{s} = \mathbf{s} \cup \mathbb{D}\{end\}$;
          split $\mathbf{s}$ into p clusters $\mathbf{L_s} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_p\}$ with Robust NMF method in Algorithm 5.1;
          break;
        end
        if $|\mathbf{s}_j| < \theta \cdot |\mathbf{s}|$ *and* $score(\mathbf{s}_j) <$ *the minimal score of existing leaves* ***and*** $score(\mathbf{s} - \mathbf{s}_j) > score(\mathbf{s})$ **then**
          drop $\mathbf{s}_j$ as a suspected outlier: $\mathbf{s} = \mathbf{s} - \mathbf{s}_j, \mathbb{D} = \mathbb{D} \cup \mathbf{s}_j, i = i + 1$;
        else
          break;
        end
      end
      if $i == \gamma$ **then**
        $\mathbf{s} = \mathbf{s}\_backup$, $p = p - 1$, $Priority(\mathbf{s}) = 1$;
      else
        $Priority(\mathbf{s}) = p$, **break**;
      end
    end
  end
  $\mathbb{C} \leftarrow$ leaves with $\max(Priority)$;
  $\mathbf{c} \leftarrow$ leaf with $\max(score)$ in $\mathbb{C}$;
  actually split $\mathbf{c}$ into clusters $\mathbf{L_c} = \left\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{\max(priority)}\right\}$;
  add $\mathbf{L_c}$ to the tree as children of $\mathbf{c}$;
end
Clusters = K leaves;
$\mathbf{CL} \leftarrow$ cluster labels of all documents according to Clusters;
$\mathbf{V} =$ onehot$(\mathbf{CL})$;
$\mathbf{H} = \mathbf{XV}$

potential to practice on new datasets without the prior knowledge.

Generally, our algorithm consists of a series of sub-clustering of the candidatures starting from the root node. At first, the root node takes all documents. And then, the algorithm repeatedly select a node to split until the algorithm terminate. We pre-defined a $k \geq 2$ representing the maximum number of wanted sub-clusters for each split step, which does not have to be 2. Therefore, the current candidature, the root node, will be divided into at most k child nodes. Note that, different from the mentioned pre-defined cluster number K before, here k only indicates a upper bound of the number of each sub-clustering, and the real resulting sub-clustering number is adaptive depending on the data distribution and our hierarchy algorithm. Then in each following step, we selected a candidature, one current leaf node, to partition until the tree meets the stop criterion. When the algorithm terminates, all the generated leaf nodes represent the final clusters of documents.

In the following of this section, we denote a node containing a set of document examples $\mathcal{N}$ with k children as a pair $(k, \mathcal{N})$, and accordingly its i-th child node as $(\sim, \mathcal{CN}_i)$ where symbol $\sim$ refers to an unknown number before the partition of this child node.

5.5.2 Candidature Selection

When it comes to the essential issue, how to choose the next node to split, we have the following principles:

1. A set of documents, representing by a current leaf node in the hierarchical structure, which contains more potential clusters, up to k evidently, is more valuable to be partitioned as early as possible.
2. For those nodes with the same number of potential clusters, we calculate a score indicating the usefulness of a node that if it is chosen for the following splitting.

The score is derived from the product of modified Normalized Discounted Cumulative Gain (mNDCG) of all child nodes of $(k, \mathcal{N})$ as $\text{score}(k, \mathcal{N}) = \prod_{i=1}^{k} \text{mNDCG}(\sim, \mathcal{CN}_i)$, which can be referred in previous works [90].

Specifically, each time we select a candidature, we first need to know how many children each leaf would potentially have. For example, for a new leaf $(\sim, \mathcal{N})$, we try clustering its documents, using proposed robust OCNMF algorithm to obtain the representa-

tion of its children. Assume the document-word matrix and the topic vector of the node $(\sim, \mathcal{N})$ is $\mathbf{X}_{(\sim,\mathcal{N})}$ and $H_{(\sim,\mathcal{N})}$, respectively. By OCNMF, we obtained the approximation $\mathbf{X}_{(\sim,\mathcal{N})} \approx \mathbf{H'V'^T}$, where its k children $(\sim, \mathcal{CN}_1), \cdots, (\sim, \mathcal{CN}_k)$ are found according to the k clusters in $\mathbf{V}$ and the corresponding k topic vectors $H_1, \cdots, H_k$, as well. Then, we compute mNDCG scores for all of its child nodes $(\sim, \mathcal{CN}_i)|_{i=1}^k$ by comparing ranked words of the father node $H_{(\sim,\mathcal{N})}$ and the child node $H_{(\sim,\mathcal{CN}_i)}$. In fact, mNDCG$(\sim, \mathcal{CN}_i)$ reflects the extent to which the topic of this child is related to its father's topic and the varying degree of topics between two generations, allowing score$(k, \mathcal{N})$ possessing the following properties:

- score$(k, \mathcal{N})$ will be large if child nodes describe well-separated topics and all of which are highly related to the parent node.
- score$(k, \mathcal{N})$ will be small if child nodes describe close topics, indicating the father should not be split any more.
- score$(k, \mathcal{N})$ will be small if only some child nodes are relevant to their parent node and others describe entirely different topics, which indicates outliers were involved in the father node from its previous splitting.

5.5.3 Pruning Strategy

NMF performs well on interpretation of detected topics and the corresponding document clusters, however, a problem appears that the number of clusters obtained from above rules (Eq. **??**-Eq. **??** or Eq. **??**-Eq. **??**) may be less than the wanted k because of the complex data distribution, which has not been discussed in previous works, but does occur in the real data sets among our experiments. Herein we hypothesise that if a set of data points can only be clustered to $k' < k$ arts, their subsets are unlikely to be able to be divided into k clusters while up to k clusters could be obtained in the next partition, which further implies that if only $k' < k$ clusters be found among the data points in this node, its k' children (clusters) are highly homogeneous already and no further partition is needed. Therefore, we stop building branch for these children. Once a child node's branch is pruned, it will no longer be considered as a candidate of the further hierarchy construction, but it is still a leaf node representing a cluster of documents, i.e., a separate topic, in the final result, and score$(k', \mathcal{CN}_i) < 0$ will be assigned to it for distinguishing it

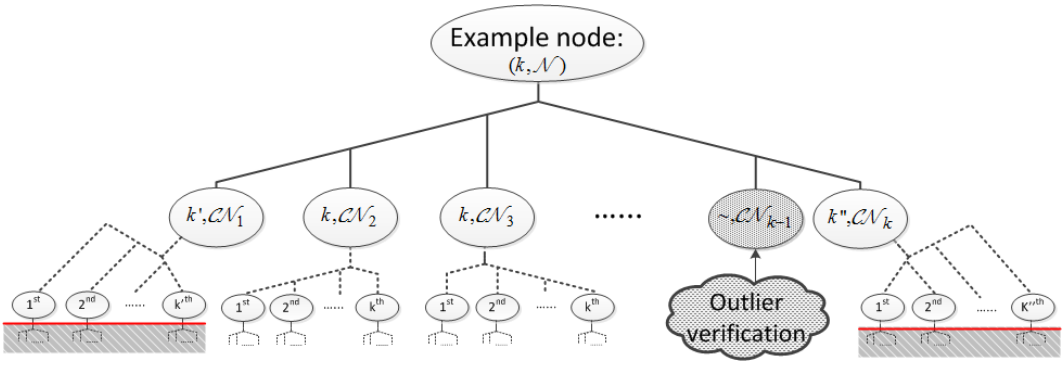


Figure 5.1: An illustration of cases that may emerge in a partition process of the node $(k, \mathcal{N})$ containing a set of document examples $\mathcal{N}$ by k clusters. The solid lines depict the actions in this step, while the dashed lines indicate actions may happen in the further steps. Three cases are demonstrated: 1) PRUNING: terminate nodes represented by the two-side child nodes $(k', C\mathcal{N}_1)$ and $(k'', C\mathcal{N}_k)$ because of $k', k'' < k$. They will be labelled as indivisible with a negative score and lose the opportunity to be chosen for further splitting. 2) OUTLIERS: represented by the second right node $(\sim, C\mathcal{N}_{k-1})$. Therefore, the verification process and more trial splitting on $(k, \mathcal{N})$ ensue. 3) QUALIFIED CANDIDATURES: represented by $(k, C\mathcal{N}_2)$ and $(k, C\mathcal{N}_3)$. The one with the largest score among them will be selected to fulfill the following hierarchy construction before it terminates.

from other real candidates.

In Subsection 5.5.5, we will introduce another pruning case accompanied by outlier verification.

5.5.4 Stopping Criteria

We propose two stopping criteria for our algorithm. i) One is that the incremental number of leaf nodes reaches the upper limit K of the topic number. When one leaf node is selected to split, the increment of leaf nodes is, at best, $k - 1$ and in the last round of split, the number of wanted clusters will be specified to $K - k + 1$ because we do not restrict the tree to be a binary tree so that a change of k should be allowed. ii) For applications in which we have no idea about the distribution of document clusters, a feasible and practical alternative is that the construction of the hierarchical structure terminates following our above pruning strategy until when all leaf nodes are pruned.

The datasets we used to conducted experiments in this work are well predefined with the specific K clusters; therefore, for ease of demonstrating and comparing the performance, we use the first stopping criterion.

5.5.5 Verification of Outliers

Outlier verification is two-step in each round of split. First, the algorithm identifies the outliers. A Node will be considered as outliers with two following conditions:

1. Too few documents were divided into this node, or
2. The score of this node is smaller than the current minimum score of candidates.

In the former case we apply a threshold θ[1] to judge whether a cluster is too small compared to its parent node. And avoiding the latter case guarantees the relative effectiveness of a new leaf node among all existing leaf nodes.

Then a verification of the outlier suspects will be applied to confirm the outliers. And we will illustrate the details of this process by example. As shown in Fig. **??**, the parent node $(k, \mathcal{N})$ contains a set of documents $\mathcal{N}$, and a subset of documents $C\mathcal{N}_{k-1} \subset \mathcal{N}$ is divided into the (k-1)-th child node (the second right child) while $|C\mathcal{N}_{k-1}| < \theta|\mathcal{N}|$. Since the number of documents in this child node is too small, we do not actually split this node but temporarily assign score$(\sim, C\mathcal{N}_{k-1}) = 0$ to it, which results in this document set $C\mathcal{N}_{k-1}$ being identified as a suspected outlier and excluded from the whole sample set of the parent node for now, $\mathcal{N}' = \mathcal{N} - C\mathcal{N}_{k-1}$. Therefore, one more tentative split step will be arranged. At the next trial, ideally, k new child nodes will be generated and the verification of $C\mathcal{N}_{k-1}$ will be carried out here by comparing two scores between $(k, \mathcal{N})$ and $(k, \mathcal{N}')$. If the new score declines instead of increasing, we countermand the last exclusion and recover the sample set to the previous state $\mathcal{N}$. Otherwise, a new round of outlier identification will be carried on the k new child nodes until no outliers are detected or the predefined trial allowance γ reaches.

Before selecting the next candidature node, we adopt this two-step verification policy to enhance the credibility of confirmed outliers. If outliers of this node $(k, \mathcal{N})$ still exist after γ times of trial splits[2], in other words, no valid partition under k obtained, the algorithm will try to split the node, to better fit the distribution of $\mathcal{N}$ with a smaller k.

The overall algorithm of hierarchy is summarized in Algorithm 5.2.

[1] θ is configured depending on the scale of datasets and k.
[2] We set γ to 3 in all our experiments.

Table 5.1: Summary of datasets and corresponding hierarchy parameters k.

Dataset	#Sample Size (n)	# of Feature (m)	# of Users (l)	# of Topics (K)	k for hierarchy
CS5	403	19,455	66,537	5	2,3,5
CS10	1,451	29,903	149,121	10	2,3,5,7,10
TS5	724	20,218	66,019	5	2,3,5
TS10	1,554	33,771	133,227	10	2,3,5,7,10
MS5	1,189	46,231	25,717	5	2,3,5
MS7	2,459	54,488	40,217	7	2,3,5,7
MS8	2505	54693	40,684	8	2,3,5,7,8
TDT2	9394	36771	-	5,7,10,15,20	2,3,5,7,10,11,13,17,20

5.5.6 Complexity Analysis

As each leaf node in the hierarchy is tried for split, expect the last generated leaves, and we ignore the constant times of outlier verfication, the complexity of building the hierarchy is $O(K-k)$, where K is the total number of clusters and k is the low rank for CMF algorithm. Then considering the integrated CMF algorithm, we suppose the optimisation stops after t iterations, therefore the cost of the CMF is $O(tmnk + tnlk)$, where n, m and l are numbers of documents, features and involved active users, respectively. To enhance weak connections denoting by $\phi(\mathbf{V})$ and $\phi(\mathbf{G})$, an extra cost of $O(n^2m + l^2n)$ is needed. Above all, the overall complexity of RHE framework is $O((K-k)\cdot(tmnk + tnlk + n^2m + l^2n))$.

5.6 Experiments and Evaluations

Extensive experiments were conducted to evaluate our proposed hierarchical method and analyse the effectiveness of $\ell_{2,1}$-norm and F-norm based optimisation function of NMF. We demonstrate our experimental settings and results in the rest of this section.

5.6.1 Datasets Description

To evaluate the proposed methods, we experiment with 8 textual datasets. Tab. 5.1 summarises the description of these datasets. The first six are real datasets published in [81][3], including real social context and Twitter users for evaluating CMF methods. Each of

[3]https://github.com/kjanani/LTECS

them contains 14 days press articles from 80 international news sources. The special thing is a list of tweets linking to each article posted in 12 hours after the press release is included. The datasets were reorganised in three categories according to the type of topics: 1) Textual Stable (TS) topics do not evolve too much in terms of text content, but keep attracting different attentions along the incremental evolution during the period of observation; 2) Community Stable (CS) topics spreads amongst some settled communities, but diverse vocabulary arises; 3) Mixed Stable (MS) topics show stability in both text content and communities. Our experiments are conducted in the consecutive 14 days and the average performance of the 14 days will be reported. Dataset 8 is a benchmark dataset consisting of TDT series datasets (TDT2 in short) that are extracted from the NIST Topic Detection and Tracking (TDT2) corpus[4] removing documents labelled with more than one topic. We test 50 random runs for each $K = 5, 7, 10, 15, 20$ using the generating code in [10] and report the average performance will be reported.

5.6.2 Evaluation Metrics

To evaluate the algorithm performances on our two different tasks, topic detection and documents clustering, we adopt two and three evaluation metrics accordingly.

Metrics measuring the performances of topic detection

To find the most similar topics between detected result and ground truth, a permutation mapping will be conducted on columns of terms-topics matrix **H**.

Mean Average Precision (MAP) reflects the algorithms global performance by calculating the mean of the Average Precision (AP) for all discovered topics. $AP_i = (\sum_{j=1}^{R} \frac{j}{rank_j})/p$ assumes the binary relevance between relevant j terms and the topic i which means a term is either of relevant of not. Since we use the top ten ranking terms to denotes the discovered topics, the number of relevant term $p = 10$. $rank_j$ is the rank position of the relevant term j in the discovered terms list. $\frac{j}{rank_j} = 0$ means the relevant term j is not discovered in the top ten terms list of the topic. For corpus with K topics, if k topics are

[4] http://www.itl.nist.gov/iad/mig/tests/tdt/1998/

detected by the algorithm, MAP will be calculated as follow:

$$MAP = (\sum_{i=1}^{k} AP_i) / \max(k, K)$$

The Ideal situation would be that the algorithm outputs $k = K$ topics. Under our first stop criterion, we always know that $k \leq K$.

Normalized Discounted Cumulative Gain (NDCG) can show multiple levels of relevance or the graded usefulness of a term regard to a topic using the cumulated gain of the term. Ideally, the most relevant term lists at the front position, therefor the gain is accumulated starting at the top of the ranking and be discounted at the lower ranks. Typically, the discount is set to $1/\log_2 rank_j$. In a corpus containing K topics, to compute groundtruth DCG or Ideal DCG (IDCG) for a topic i, we assume that the word $j \in$ top ten representative words relevant to topics with the same relevancy rel_j. We have $IDCG_i = 1 + \sum_{j=2}^{R} \frac{rel_j}{\log_2 rank_j}$. Then for detected words-topics matrix with k topics, the DCG of a topic i is defined as $DCG_i = 1 + \sum_{j=2}^{R} \frac{rel_j}{\log_2 rank_j}$. It also indicates that if a relevent word is not detected as the top ten words, the relevancy $rel_j = 0$. Then the NDCG of the corpus can be defined as:

$$NDCG = (\sum_{i=1}^{k} \frac{DCG_i}{IDCG_i}) / \max(k, K).$$

Metrics measuring the results of document clustering

To map each cluster label to the equivalent ground truth label, a permutation mapping on columns of documents-topics matrix $\mathbf{V}$ is necessary before we compute the following clustering evaluation metrics. The mapping was implemented with the Hungarian algorithm. Let cl_i denote the cluster label of document i after mapping.

Accuracy (AC) measures the degree of correspondence between a predicted cluster label of a data sample and its closest class labelling in the ground truth, defined as follow:

$$AC = \frac{\sum_{i=1}^{n} \delta(cl_i, cl\_gnd_i)}{n}$$

where $\delta(cl_i, cl\_gnd_i)$ is the delta function that equals to 1 if $cl_i = cl\_gnd_i$ and equals to 0 otherwise, and n is the total number of documents.

Purity (PU) transparently evaluates the extent to which the data points from one ground truth class are assigned to one predicted cluster. It is computed by counting the number of correctly assigned documents and dividing by the total number of documents n, given as follow formally:

$$Purity(C\_gnd, C) = \frac{1}{n}\sum_{i=1,j=1}^{K} \max_{j} \left|c_i \cap c'_j\right|$$

where, $c_i \in C\_gnd$ and $c'_j \in C$ are the sets of documents belong to predicted clusters and ground truth classes correspondingly. Purity(C_gnd,C) is bounded in the range of $[0, 1]$ and the large the better.

Normalized Mutual Information (NMI) determines the mutual dependence between the set of predicted clusters and the set of ground truth classes, and bounds the value range to $[0, 1]$ for computation and comparison with following expression:

$$NMI(C\_gnd, C) = \frac{MI(C\_gnd, C)}{\max(H(C\_gnd), H(C))}$$

where $H(C\_gnd)$ and $H(C)$ are the entropies of C_gnd and C, and $MI(C\_gnd, C)$ is the mutual information of the compared pair. $MI(C\_gnd, C) = \sum_{i=1}^{K}\sum_{j=1}^{k} \frac{\left|c_i \cap c'_j\right|}{n_d} \log_2 \frac{n_d \left|c_i \cap c'_j\right|}{|c_i|\left|c'_j\right|}$, where $c_i \in C\_gnd$ and $c'_j \in C$ are the sets of documents, estimates the information sharing between C_gnd and C, that is the higher $MI(C\_gnd, C)$, the more information they share. However, it is not always less than 1 and hard to interpret among different pairs. Therefore, we normalise it with the maximum entropy.

From above expressions, we observes that NMI considers the entire distribution of the corpus in the computation while PU only involve the largest cluster with the most correctly predicted documents and AC averages out the one-to-one corresponding relationship for each document.

5.6.3 Experimental Setup

We involved five baseline methods plus our proposed three methods in the experiments as follows:

- RHE series methods (RHEs in short): It is the proposed hierarchical ensemble method in our work, in which three optimization functions are used to form three algo-

rithms: RHE_ALM, RHE_MUL21 and RHE_MULF. To verify the outliers during the hierarchy construction, we set $\theta = 0.1$ and $\gamma = 3$. We tune μ in the range of $\{0.1, \cdots, 0.9\}$, λ_1, λ_2 in the range of $\{0, 10^0, \cdots, 10^4\}$ for RHE_MUL21 and RHE_MULF, while λ_1, λ_2 in the range of $\{0, 10^{-9}, \cdots, 10^{-5}\}$ for RHE_ALM. We set k corresponding to the datasets, presented in the right column in Tab. 5.1.

- NMF [94]: Standard NMF method implemented with multiplicative updating rules and F-norm formulation.
- GNMF [24]: Geometric information is utilized as a p-nearest neighbour graph extracted from the original data set. We search 5 nearest neighbours of each data sample to build the Laplacian graph by the author and tune the regularisation parameters λ in the grid $\{0, 0.1, 1, 10, 100, 1000, 10000\}$ for the best and stable performance.
- FastR2 [90]: A rank-2 hierarchical NMF method was solved as alternating non-negative least squares with an active set type algorithm. The outlier threshold and trial allowance are set to 0.1 and 3 for outlier verification of hierarchy.
- LETCS [81]: A multi-view collected matrix factorization method models topic evolution with a transition matrix, which is applied to linearly associate the current topic matrix with the previous one. We tune its regularization parameters in the range of $10^{[0,7]}$ by the author.
- NMFCP [176]: A multi-relation collected matrix factorization method introduced a constraint propagation mechanism to enhance inherent pairwise relations among original data points. The regularization parameters are set as same as our RHEs with multiplicative updating rules.

5.6.4 User Selection

Unlike the number of common words is easily controlled and organised within one corpus, the number of involved users sometimes may become super large, displaying fluctuations with regard to topic type, stage of topic evolution, event propagation and even the influence of participating opinion leader. Subject to computing resource and the purpose of removing noises and discover really relevant users, we generate a subset of the whole user set forming $\mathbf{U}'$ with two strategies. First, we only consider those users who

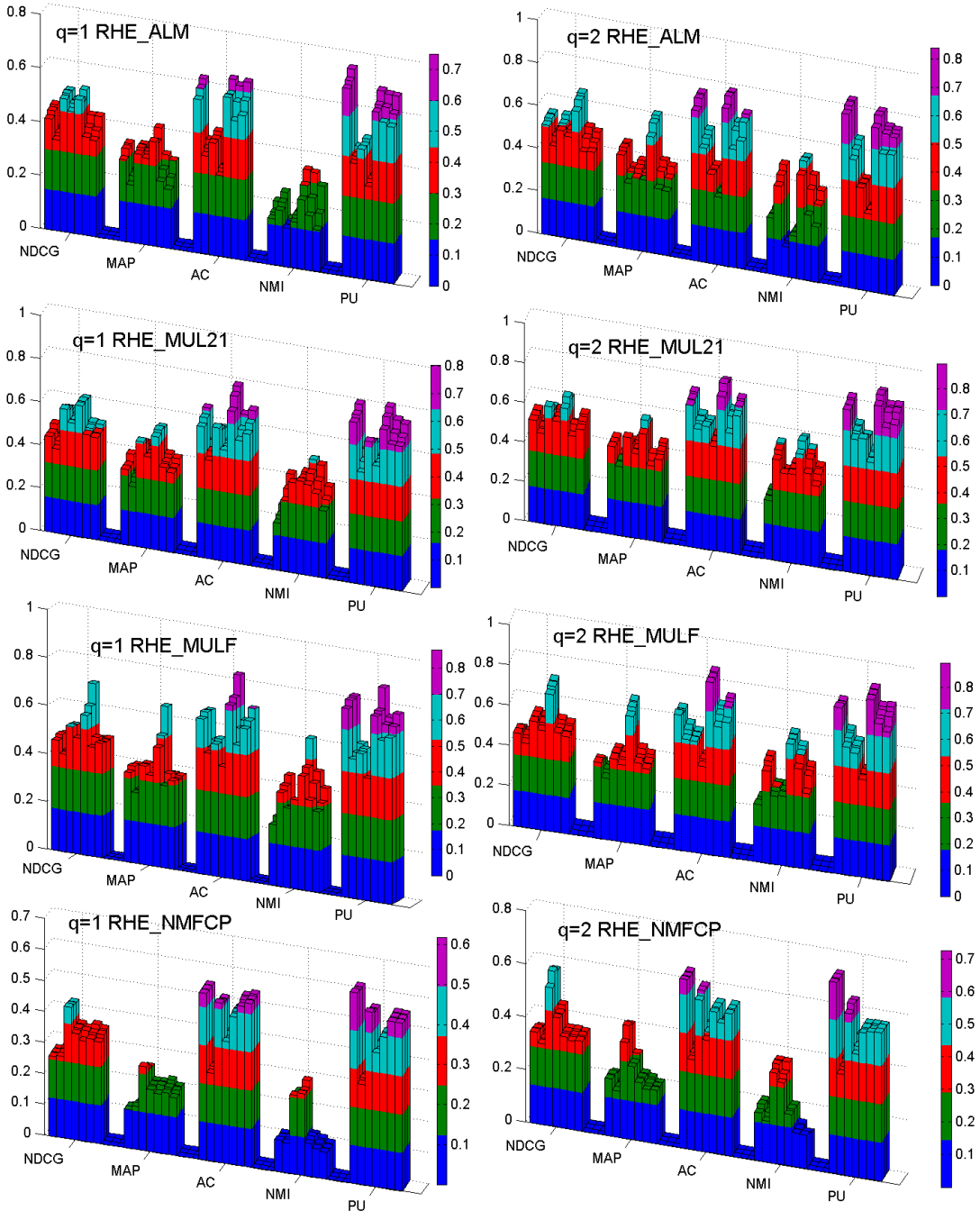


Figure 5.2: The performance on different user scales. Performances of four methods (RHE_ALM, RHE_MUL21, RHE_MULF and NMFCP) are shown from top to down, containing five metrics from left to right in each graph. And for each metrics, 7 bars correspond to results on 7 datasets sequentially mentioned in Table 5.1. We illustrate three user scales (500, 1000, 2000) for each dataset, corresponding to the three rows of bars from the front to the back. Results of users selected from $q = 1, 2$ consecutive days are placed on the left side and right side, respectively.

are active in q consecutive time steps and secondly we keep user scale under a predefined number, max $\_user \leq 2000$ for example. After the random selection of max $\_user$ users, we calculate the pairwise weight on $\mathbf{U}'$ which contains fewer but more representative users. This compromise is based on our observation that compared to a random user who occasionally read or reposts an article about a topic, those users who pay sustained attention to a topic in a period are more likely to be truly interested in this topic. The time step can be set to a day, an hour or any other time interval as required. Under the predefined corpus, the data of texture information and community information in a time-step comprises the entire set of our input data of the time-step. In datasets 1-7, we take each day as a time step. Fig. 5.2 revealed the performance of the number of users and the quality of users for four methods where community information are involved and number of users is adjustable, which are RHE_ALM, RHE_MUL21, RHE_MULF and NMFCP from up to down, on seven out of eight datasets we mentioned above. Each bar from the left to the right of each bunch of bars in each subgraph corresponds to a dataset of datasets 1-7, and from the front to the back, each bar corresponds to the user scale 500, 1000 and 2000. A clear trend is that more users lead to better performances on all of the metrics; meanwhile, those results with users of higher quality, selected by $q = 2$ in the right side subgraphs, tend to be better than those $q = 1$ in the left subgraphs. This result shows that it is of great necessity to carefully select high-quality users for advanced performance under limited computing resource.

Table 5.2: Statistics of involved users in consecutive q days.

q		CS5	CS10	TS5	TS10	MS5	MS7	MS8
1	Avg. # of users	5,830	15,823	6,608	14,238	2,636	5,009	5,112
	Max. # of users	11,189	22,050	13,732	26,653	5,672	7,744	7,762
2	Avg. # of users	798	2,938	1,090	2,739	478	1,288	1,336
	Max. # of users	1,800	4,150	1,966	5,810	1,611	2,246	2,662
3	Avg. # of users	383	1,447	594	1,497	141	610	640
	Max. # of users	466	2,077	851	2,062	188	808	862

We display the statistics of users of datasets 1 to 7 in Tab. 5.2. For reducing the loss of less users in one single day, we set $q = 3$ and max $\_user \leq 2000$ for following experiments.

5.6.5 Result Analysis

We show the topic detection performance through the left two columns of the following tables and the clustering results are discussed through the right three columns. Tab. 5.3 - Tab. 5.4 show the results on real CMF datasets. The best performance of each method is shown with the corresponding parameters. The best results of baseline methods are indicated with daggers and the best results among all methods are emphasised in bold. On datasets TDT2 (shown in Fig. 5.5 - Fig. 5.5) without multi-timesteps and multi-view, LETCS reduces to the standard NMF; therefore, we do not test LETCS on TDT2. Since

Table 5.3: Performance over three MS datasets. The best result is indicated in **bold**.

		Topic Detection		Document Clustering		
MS5	parameters	NDCG	MAP	AC	NMI	PU
NMF		0.4489	0.3115	0.6618	0.3835	0.7012
GNMF	$\lambda = 10^4$	0.5812	0.4942	0.7993	0.5566	0.8582
FASTR2		0.5214	0.4223	0.7218	0.5046	0.8082
LETCS	$\mu = 0.3, \lambda = 1$	0.6756†	0.6317†	0.8714†	0.6490†	0.9295†
NMFCP	$\mu = 0.9, \lambda_1 = 10^4, \lambda_2 = 10^2$	0.6159	0.5106	0.8125	0.5739	0.8710
RHE_MULF	$k = 5, \mu = 0.5, \lambda_1 = 10^1, \lambda_2 = 10^2$	**0.7025**	**0.6695**	**0.9129**	**0.6980**	**0.9522**
RHE_MUL21	$k = 5, \mu = 0.3, \lambda_1 = 10^1, \lambda_2 = 10^1$	0.6627	0.6120	0.8822	0.6784	0.9464
RHE_ALM	$k = 5, \mu = 0.1, \lambda_1 = 10^{-5}, \lambda_2 = 10^{-8}$	0.6589	0.6134	0.8044	0.5677	0.8641
MS7						
NMF		0.4718	0.3812	0.6020	0.4382	0.7432
GNMF	$\lambda = 10^3$	0.5037	0.4391	0.6380	0.5008	0.7725
FASTR2		0.4998	0.4232	0.5879	0.4708	0.7572
LETCS	$\mu = 0.9, \lambda = 1$	0.5993†	0.5465†	0.7970†	0.6505†	**0.9336**†
NMFCP	$\mu = 0.9, \lambda_1 = 10^3, \lambda_2 = 10^1$	0.5622	0.4808	0.6348	0.5115	0.7734
RHE_MULF	$k = 3, \mu = 0.9, \lambda_1 = 1, \lambda_2 = 10^2$	**0.6247**	**0.5706**	**0.8072**	**0.6531**	0.8935
RHE_MUL21	$k = 2, \mu = 0.5, \lambda_1 = 10^1, \lambda_2 = 10^1$	0.5365	0.4572	0.6496	0.5506	0.8429
RHE_ALM	$k = 7, \mu = 0.7, \lambda_1 = 10^{-7}, \lambda_2 = 10^{-6}$	0.5622	0.4808	0.6348	0.5115	0.7734
MS8						
NMF		0.5085	0.4230	0.6348	0.3248	0.7170
GNMF	$\lambda = 10^3$	0.5183	0.4473	0.6707	0.3990	0.7573
FASTR2		0.5081	0.4332	0.6306	0.3572	0.7433
LETCS	$\mu = 0.3, \lambda = 1$	0.5643	0.4840†	0.7545†	0.5497†	0.9032†
NMFCP	$\mu = 0.9, \lambda_1 = 10^4, \lambda_2 = 10^2$	0.5743†	0.4684	0.6556	0.4033	0.7633
RHE_MULF	$k = 3, \mu = 0.7, \lambda_1 = 1, \lambda_2 = 10^2$	**0.6470**	**0.5860**	**0.8535**	**0.5669**	**0.9127**
RHE_MUL21	$k = 8, \mu = 0.1, \lambda_1 = 10^2, \lambda_2 = 10^1$	0.5218	0.4372	0.7328	0.4646	0.8301
RHE_ALM	$k = 2, \mu = 0.1, \lambda_1 = 10^{-9}, \lambda_2 = 10^{-6}$	0.5738	0.4985	0.7802	0.4763	0.8388

Table 5.4: Performance over two TS (up) and CS (down) datasets.

		Topic Detection		Document Clustering		
TS5	parameters	NDCG	MAP	AC	NMI	PU
NMF		0.5089	0.3212	0.4709	0.2503	0.5398
GNMF	$\lambda = 10^4$	0.5917	0.4770	0.6527	0.4728	0.6978
FASTR2		0.5791	0.4782†	**0.6856**†	**0.5675**†	**0.7747**†
LETCS	$\mu = 0.7, \lambda = 10^3$	0.4127	0.2412	0.4919	0.2049	0.5324
NMFCP	$\mu = 0.9, \lambda_1 = 10^4, \lambda_2 = 1$	0.6375†	0.4350	0.6348	0.4270	0.6646
RHE_MULF	$k = 2, \mu = 0.9, \lambda_1 = 10^2, \lambda_2 = 10^4$	0.6453	0.5809	0.6792	0.5057	0.7413
RHE_MUL21	$k = 2, \mu = 0.7, \lambda_1 = 10^2, \lambda_2 = 10^1$	**0.6686**	**0.6067**	0.6800	0.5640	0.7660
RHE_ALM	$k = 3, \mu = 0.9, \lambda_1 = 10^{-9}, \lambda_2 = 10^{-8}$	0.5829	0.4248	0.4938	0.2432	0.5577
TS10						
NMF		0.4704	0.2728	0.3999	0.2876	0.4511
GNMF	$\lambda = 10^3$	0.4659	0.3437	0.5422	0.4711	0.6016
FASTR2		0.5150	0.4147†	0.5566†	0.4974†	0.6246†
LETCS	$\mu = 0.9, \lambda \geq 10^2$	0.3169	0.1292	0.3072	0.1718	0.3535
NMFCP	$\mu = 0.5, \lambda_1 = 10^4, \lambda_2 = 0$	0.5645†	0.4027	0.5245	0.4450	0.5742
RHE_MULF	$k = 2, \mu = 0.9, \lambda_1 = 10^1, \lambda_2 = 10^4$	0.5902	0.5087	0.5933	0.5176	0.6420
RHE_MUL21	$k = 2, \mu = 0.7, \lambda_1 = 10^1, \lambda_2 = 10^4$	**0.6064**	**0.5316**	**0.6006**	**0.5475**	**0.6630**
RHE_ALM	$k = 2, \mu = 0.9, \lambda_1 = 10^{-5}, \lambda_2 = 10^{-8}$	0.5021	0.3619	0.3403	0.2550	0.4132
CS5						
NMF		0.5851†	0.4196†	0.5947	0.1018	0.6415
GNMF	$\lambda = 10^4$	0.5022	0.4133	0.5607	0.1163	0.6312
FASTR2		0.4641	0.3833	0.5589	0.1048	0.6340
LETCS	$\mu = 0.9, \lambda = 10^1$	0.4097	0.2366	0.6642	0.2110	0.7311
NMFCP	$\mu = 0.1, \lambda_1 = 1, \lambda_2 = 10^3$	0.4234	0.3085	**0.7660**†	0.3024†	0.8313†
RHE_MULF	$k = 2, \mu = 0.9, \lambda_1 = 0, \lambda_2 = 10^3$	**0.5876**	**0.5562**	0.7588	0.3377	0.8763
RHE_MUL21	$k = 2, \mu = 0.1, \lambda_1 = 1, \lambda_2 = 0$	0.5636	0.5118	0.7638	0.3235	0.8740
RHE_ALM	$k = 2, \mu = 0.9, \lambda_1 = 10^{-6}, \lambda_2 = 10^{-8}$	0.5501	0.4818	0.7502	**0.3410**	**0.8809**
CS10						
NMF		0.4250†	0.2890†	0.3001	0.1383	0.3524
GNMF	$\lambda = 10^3$	0.3880	0.2705	0.3200	0.1983	0.3967
FASTR2		0.3197	0.2034	0.2984	0.1515	0.3549
LETCS	$\mu = 0.1, \lambda \geq 10^2$	0.3569	0.2338	0.4874	0.3645	0.5707
NMFCP	$\mu = 0.5, \lambda_1 = 0, \lambda_2 = 10^3$	0.3680	0.2531	0.5596†	0.4275†	0.6109†
RHE_MULF	$k = 3, \mu = 0.1, \lambda_1 = 10^2, \lambda_2 = 10^2$	**0.5267**	**0.4481**	0.6211	**0.5669**	0.7459
RHE_MUL21	$k = 10, \mu = 0.1, \lambda_1 = 10^1, \lambda_2 = 10^2$	0.4564	0.3479	**0.6736**	0.5667	**0.7519**
RHE_ALM	$k = 10, \mu = 0.7, \lambda_1 = 0, \lambda_2 = 10^{-8}$	0.4958	0.4014	0.5985	0.5304	0.7003

there is no information of user concerns, only λ_1 is needed for NMFCP, RHE_MULF and RHE_MUL21. To all experiments across TDT2, we simply set $\lambda_1 = 10^2$ for all NMF methods of Lagrangian Multiplier and $\lambda_1 = 10^{-7}$ for RHE_ALM method of Augmented Lagrangian. In each dataset, results in bold under a metric represent a significant improvement in performance using the Student-t significant test and $p < 0.05$.

Comparing the results of two categories of metrics, we can see some slight incongruities between them. For example, on TS5, FASTR2 has the best performance on clustering but does not obtain a consistently good value on topic detection metrics, and on CS, NMF performs best in topic detection but relatively poor in document clustering among baseline methods. However, in most cases, all the metrics are generally positively correlated and accordant.

Overall, we observe that our RHEs achieve significantly better performances than other compared methods when considering both tasks of topic detection and document clustering, especially on datasets with a larger K, demonstrating the robustness of our RHEs methods. For example, FASTR2 performs rather good in document clustering on TS, however, does not obtain a consistently good value on other datasets; similarly, LETCS outperforms other baselines on MS datasets but lags behind on other datasets, especially bad on TS. Besides, in some particular cases, RHEs trail closely behind the best-performed method for either topic detection or document clustering. For instance, NMFCP tends to perform well on TDT2 $K = 5, 7$, followed by RHE_ALM and RHE_MULF.

For the difference between $\ell_{2,1}$-norm and F-norm based objective functions, our results of both topic detection and document clustering do not actively support the remarkable robustness brought by $\ell_{2,1}$-norm mentioned by [3], [18] that mainly worked on image data. For example, on TDT2 datasets, RHE_MULF performs as good as RHE_ALM; however, RHE_MUL21 even performs worse than some of our baseline methods, NMFCP or standard NMF for instance. The improvement of F-norm based RHE_MULF method evidences the limitations of simply applying $\ell_{2,1}$-norm to the objective function due to the complexity of semantic circumstance for textual data to some extent, indicating the necessity of more robust learning method and the advantage of our ensemble method as well. In addition, it is also worth noticing that for $\ell_{2,1}$-norm based objective function, adopting different updating method seriously impact the experimental results, especially

Table 5.5: Clustering results over TDT2 datasets. The best result is indicated in **bold**.

			Topic Detection		Document Clustering		
Dataset		k	NDCG	MAP	AC	NMI	PU
$K = 5$	NMF		0.7761	0.6989	0.8033	0.6805	0.8933
	GNMF		0.6559	0.5342	0.9096	0.7932	0.9346
	FastR2		0.7148	0.6294	0.7672	0.6669	0.8923
	NMFCP		**0.8957**†	**0.8614**†	**0.9367**†	**0.8394**†	**0.9605**†
	RHE_MULF	2	0.7841	0.7173	0.7669	0.7117	0.9135
		3	0.7852	0.7172	0.7686	0.6988	0.9101
		5	**0.9075**	**0.8724**	0.8842	**0.8131**	**0.9498**
	RHE_MUL21	2	0.7723	0.6951	0.7131	0.6573	0.9113
		3	0.6624	0.5529	0.6544	0.5981	0.8681
		5	0.7510	0.6597	0.7542	0.6523	0.8852
	RHE_ALM	2	0.7381	0.6507	0.8276	0.7203	0.8852
		3	0.8252	0.7687	0.8713	0.7525	0.9061
		5	**0.8965**	**0.8621**	0.8778	**0.8209**	**0.9603**
$K = 7$	NMF		0.8105	0.7474	0.8227	0.7240	0.9084
	GNMF		0.6210	0.4998	0.8755	0.7903	0.9361
	FastR2		0.6188	0.5181	0.6850	0.6205	0.8715
	NMFCP		**0.8815**†	**0.8454**†	**0.9101**†	**0.8386**†	0.9531†
	RHE_MULF	2	0.7076	0.6207	0.6793	0.6646	0.8956
		3	0.6916	0.6023	0.6874	0.6553	0.8813
		5	0.7711	0.6991	0.7709	0.7398	0.9110
		7	**0.8777**	**0.8329**	0.8466	0.7981	0.9439
	RHE_MUL21	c 2	0.6778	0.5772	0.6100	0.6214	0.8842
		3	0.7120	0.6164	0.6649	0.6472	0.8796
		5	0.6673	0.5536	0.6477	0.6037	0.8546
		7	0.7247	0.6291	0.7334	0.6513	0.8758
	RHE_ALM	2	0.6639	0.5647	0.7984	0.6999	0.8608
		3	0.6872	0.5932	0.8352	0.7341	0.8678
		5	0.8106	0.7520	0.8722	0.8020	0.9201
		7	**0.8979**	**0.8638**	0.8633	**0.8157**	**0.9619**
$K = 10$	NMF		0.7564	0.6837	0.7265	0.6919	0.8616
	GNMF		0.5704	0.4441	0.7804	0.7439	0.9062
	FastR2		0.5824	0.4761	0.6291	0.6000	0.8324
	NMFCP		0.8045†	0.7468†	0.8466†	**0.7924**†	0.9199†
	RHE_MULF	2	0.6831	0.5879	0.6533	0.6808	0.8744
		3	0.6899	0.5946	0.6808	0.6940	0.8727
		5	0.6845	0.5870	0.7024	0.7015	0.8592
		7	0.7301	0.6452	0.7608	0.7473	0.8877
		10	**0.8581**	**0.8090**	**0.8332**	**0.7881**	**0.9272**
	RHE_MUL21	2	0.6578	0.5534	0.5864	0.6316	0.8585
		3	0.7095	0.6135	0.6200	0.6582	0.8799
		5	0.6183	0.4983	0.6119	0.6329	0.8253
		7	0.6560	0.5372	0.6368	0.6376	0.8274
		10	0.7119	0.6129	0.7526	0.6898	0.8548
	RHE_ALM	2	0.6313	0.5259	0.7481	0.6831	0.8355
		3	0.6462	0.5475	0.7648	0.6869	0.8325
		5	0.6834	0.5920	0.7972	0.7287	0.8592
		7	0.7504	0.6796	**0.8502**	**0.7955**	0.8986
		10	**0.8528**	**0.8050**	0.7963	**0.7734**	**0.9376**

Table 5.6: Clustering results over TDT2 datasets. The best result is indicated in **bold**.

			Topic Detection		Document Clustering		
Dataset		k	NDCG	MAP	AC	NMI	PU
$K = 15$	NMF		0.7538	0.6811	0.6565	0.6719	0.8391
	GNMF		0.5290	0.4004	0.6798	0.7025	0.8748†
	FastR2		0.5378	0.4248	0.5459	0.5737	0.7981
	NMFCP		0.7585†	0.6955†	0.7403†	0.7141†	0.8678
	RHE_MULF	2	0.6637	0.5692	0.6002	0.6685	0.8622
		3	0.6146	0.5108	0.6028	0.6511	0.8373
		5	0.6452	0.5431	0.6364	0.6816	0.8509
		7	0.6501	0.5479	0.6625	0.6965	0.8510
		11	0.7349	0.6564	**0.7713**	**0.7708**	**0.8893**
		13	0.7962	0.7347	**0.7838**	**0.7853**	**0.9073**
		15	**0.8398**	**0.7877**	**0.7767**	**0.7798**	**0.9208**
	RHE_MUL21	2	0.6323	0.5171	0.5288	0.6280	0.8481
		3	0.6414	0.5265	0.5546	0.6396	0.8459
		5	0.6229	0.5088	0.6100	0.6458	0.8221
		7	0.5533	0.4191	0.5937	0.6277	0.7806
		11	0.5818	0.4505	0.5651	0.5841	0.7814
		13	0.6468	0.5287	0.6061	0.5950	0.7913
		15	0.6583	0.5387	0.6742	0.6210	0.7932
	RHE_ALM	2	0.5820	0.4675	0.6434	0.6688	0.8103
		3	0.5608	0.4392	0.7141	0.6587	0.8032
		5	0.6211	0.5161	0.7822	0.7187	0.8366
		7	0.6387	0.5380	**0.7843**	0.7292	0.8442
		11	0.7142	0.6339	**0.8001**	**0.7778**	**0.8794**
		13	**0.8243**	**0.7689**	**0.7842**	**0.7662**	**0.9054**
		15	**0.8386**	**0.7863**	0.7321	0.7369	**0.9106**
$K = 20$	NMF		0.7629†	0.6934†	0.6221	0.6715	0.8182
	GNMF		0.5073	0.3876	0.6501	0.7013†	0.8569†
	FastR2		0.5194	0.4007	0.5133	0.5845	0.7820
	NMFCP		0.7303	0.6631	0.6995†	0.7088†	0.8416
	RHE_MULF	2	0.6297	0.5250	0.5758	0.6743	0.8414
		3	0.6160	0.5110	0.5984	0.6831	0.8362
		5	0.5810	0.4693	0.6119	0.6789	0.8202
		7	0.6328	0.5307	0.6590	0.7151	0.8412
		11	0.6319	0.5291	0.7406	**0.7628**	**0.8563**
		13	0.6751	0.5853	**0.7678**	**0.7859**	**0.8714**
		17	**0.7824**	**0.7203**	**0.7913**	**0.8044**	**0.9020**
		20	**0.8458**	**0.7966**	**0.7729**	**0.7912**	**0.9148**
	RHE_MUL21	2	0.6163	0.5004	0.5233	0.6484	0.8388
		3	0.6616	0.5562	0.5685	0.6650	0.8497
		5	0.6832	0.5808	0.6482	0.6887	0.8433
		7	0.5823	0.4604	0.6569	0.6915	0.7921
		11	0.5085	0.3637	0.6152	0.6411	0.7460
		13	0.5481	0.4101	0.5978	0.6226	0.7590
		17	0.6319	0.5116	0.6214	0.6217	0.7736
		20	0.6616	0.5464	0.6736	0.6304	0.7760
	RHE_ALM	2	0.5976	0.4856	0.6506	0.6857	0.8063
		3	0.5448	0.4228	0.6707	0.6440	0.7714
		5	0.5756	0.4615	0.7608	0.6924	0.8017
		7	0.5611	0.4451	0.7699	0.7169	0.8128
		11	0.5898	0.4819	**0.7797**	**0.7602**	0.8327
		13	0.6351	0.5365	**0.7742**	**0.7756**	0.8453
		17	**0.7950**	**0.7320**	**0.7570**	**0.7552**	**0.8793**
		20	**0.8110**	**0.7495**	0.7003	0.7232	**0.8892**

on TDT2 datasets. We interpret this result as a possibility of the non-applicability of Multiplicative updating rule at simple textual context, which results in the unexpected bad result of RHE_MUL21 on TDT2. However, on real datasets integrating community information, minor difference indicates an improvement in the non-applicability situation of RHE_MUL21.

Besides, we can find that though RHE_ALM performs well on TDT2, it is not always suitable for real datasets with community information; on the contrary, RHE_MUL21 obtained rather good results on TS5 and TS10.

5.6.6 Parameter Analysis

In the following, we present the discussion of parameter analysis for essential parameters of RHEs methods. Representatives of Three types of parameters are included below.

Regularisation parameters λ_1 and λ_2 control the degree of the effect caused by soft constraint terms. A larger value represents more weight on this term and vice versa. Fig. 5.3, Fig. 5.4 and Fig. 5.5 show how the performances of our RHEs varied with λ_1

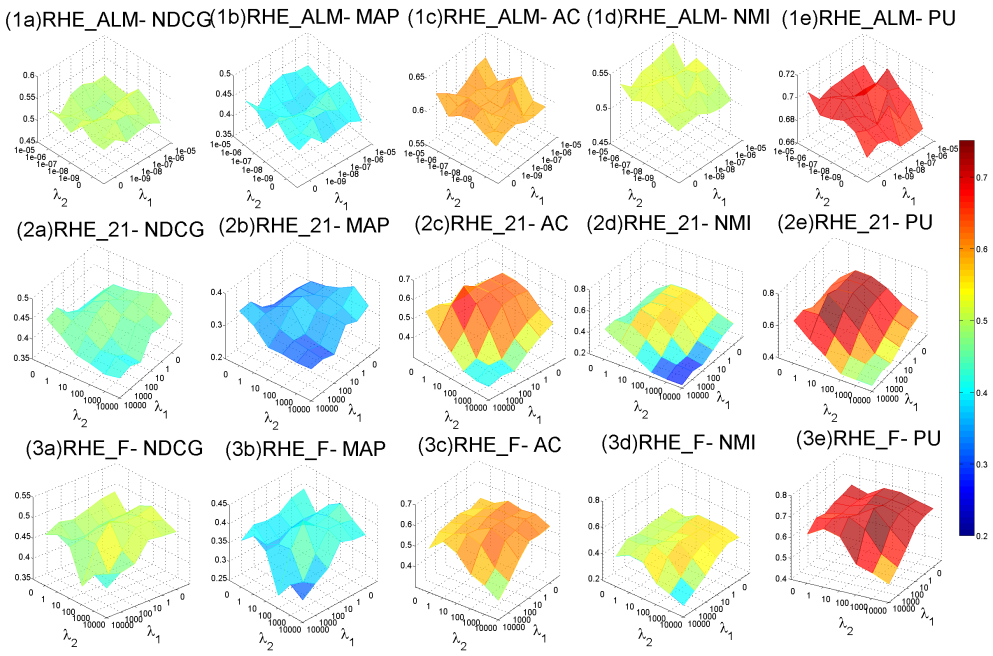


Figure 5.3: Parameters λ_1 and λ_2 on CS10.

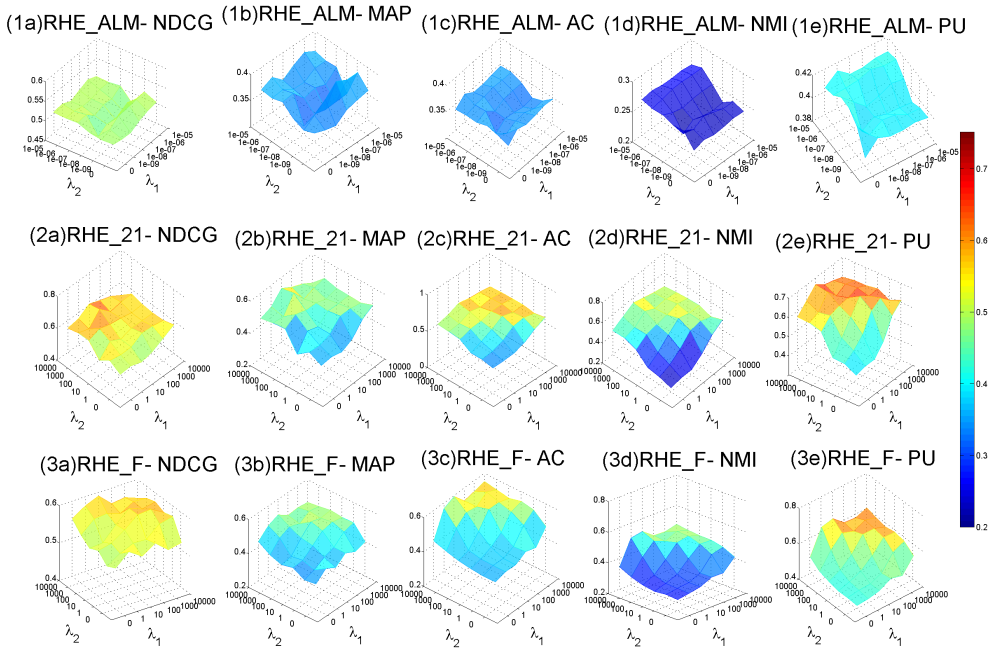


Figure 5.4: Parameters λ_1 and λ_2 on TS10.

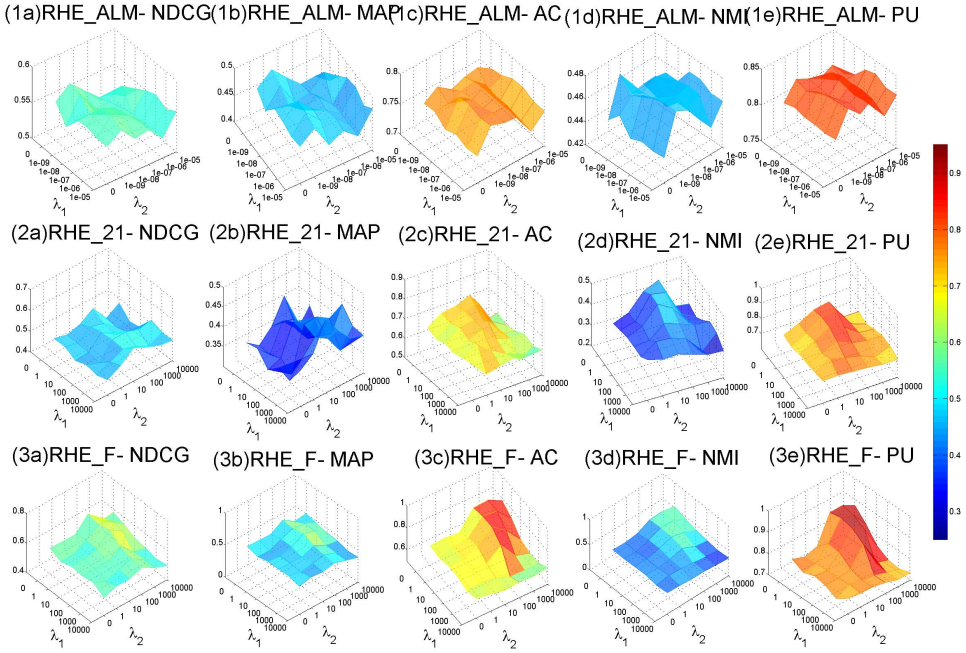
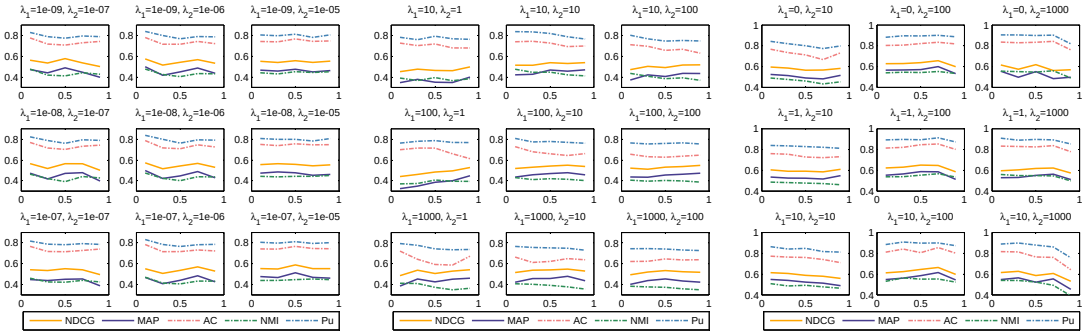
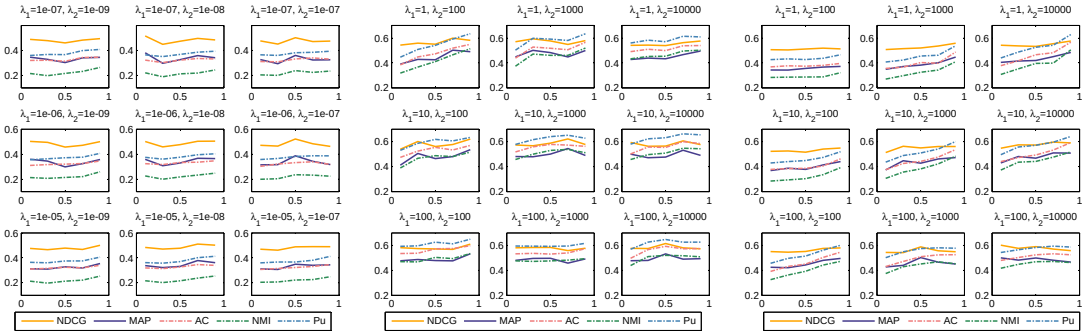


Figure 5.5: Parameters λ_1 and λ_2 on MS8.



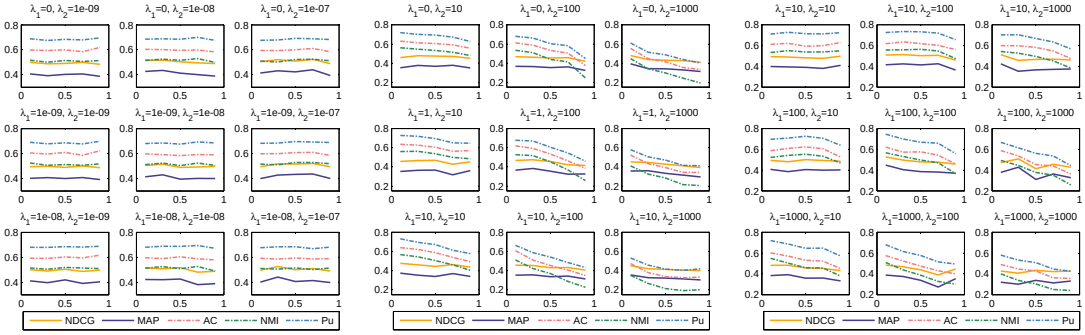
(a) RHE_ALM (b) RHE_MUL21 (c) RHE_MULF

Figure 5.6: Performances of RHEs vs. the parameters μ on MS8: Solid lines depict the topic detection performances while dash-dot lines specify metrics of the clustering result



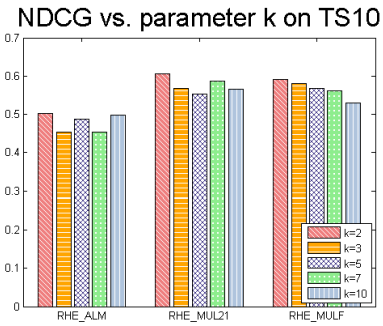
(a) RHE_ALM (b) RHE_MUL21 (c) RHE_MULF

Figure 5.7: Performances of RHEs vs. the parameters μ on TS10

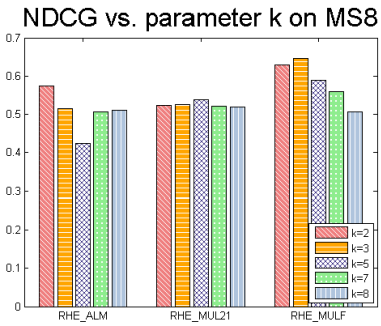


(a) RHE_ALM (b) RHE_MUL21 (c) RHE_MULF

Figure 5.8: Performances of RHEs vs. the parameters μ on CS10

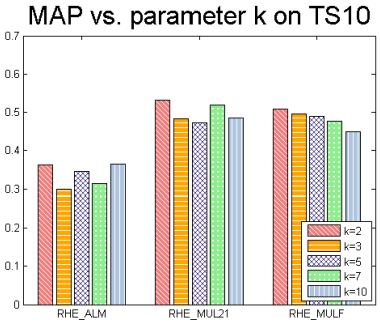


(a) TS10 _ NDCG

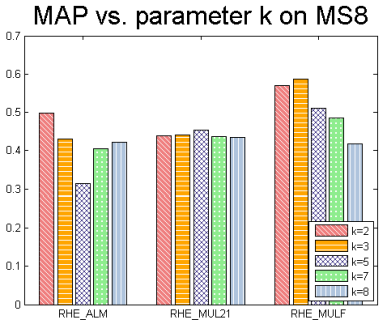


(b) MS8 _ NDCG

Figure 5.9: Performance of NDCG on TS10 (left) and MS8 (right) vs. the parameter k

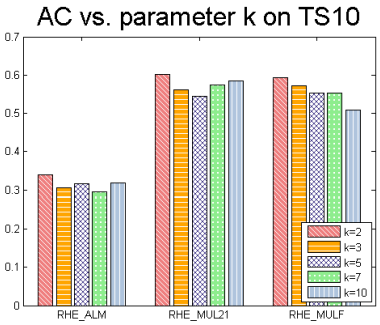


(a) TS10 _ MAP

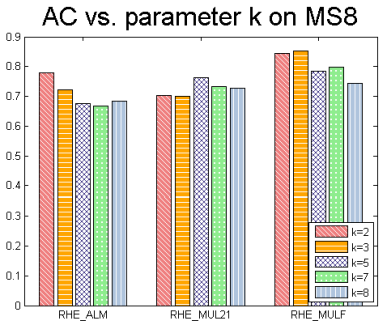


(b) MS8 _ MAP

Figure 5.10: Performance of MAP on TS10 (left) and MS8 (right) vs. the parameter k



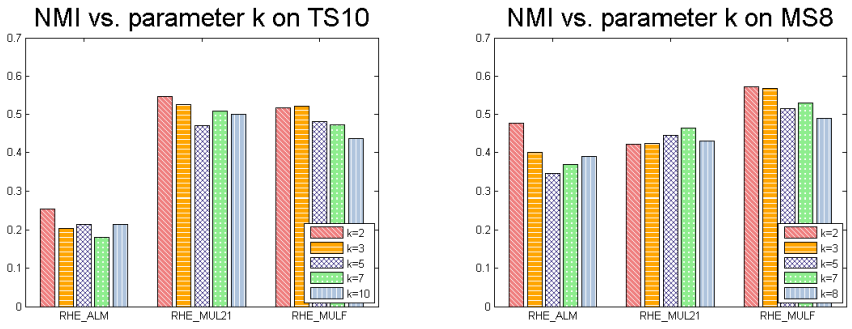
(a) TS10 _ AC



(b) MS8 _ AC

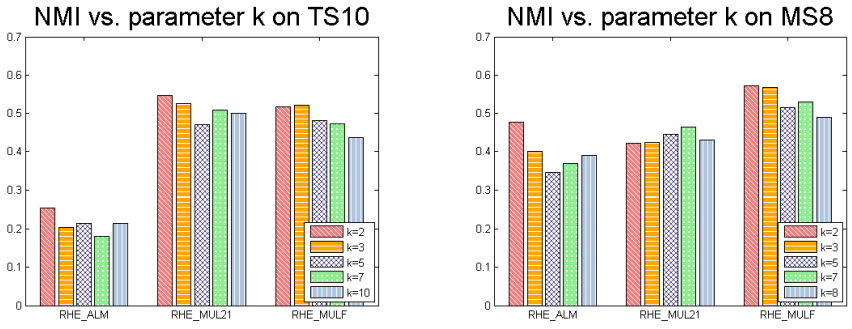
Figure 5.11: Performance of AC on TS10 (left) and MS8 (right) vs. the parameter k

and λ_2 on CS10, TS10 and MS8, respectively. We can find that our HER series methods are sensitive to λ_1 and λ_2, demonstrating that choosing proper values for them is very crucial to the performance, which has also attracted much attention in previous work-



(a) TS10 _ NMI (b) MS8 _ NMI

Figure 5.12: Performance of NMI on TS10 (left) and MS8 (right) vs. the parameter k



(a) TS10 _ PU (b) MS8 _ PU

Figure 5.13: Performance of PU on TS10 (left) and MS8 (right) vs. the parameter k

s [10], [37]. Compared to topic detection performance, the clustering results vary more regularly with the changes of λ_1 and λ_2, especially in multiplicative updating methods RHE_MUL21 and RHE_MULF. In particular, on TS10, larger values of λ_1 and λ_2 boost the topic detection and document clustering results simultaneously. As in Fig. 5.4, all metrics increase when λ_1 and λ_2 varying from 0 to bigger values. While on CS10, Fig. 5.3, this trend can only be observed in the performances of RHE_ALM, and the performances of RHE_MUL21 and RHE_MULF asymptotically improve with λ_1 and λ_2 increasing until they reach peak followed by reductions afterwards. After all, the two types of updating rules are different.

Trade-off parameter μ regularises the relevant portion between two relation matrices during the approximation. Fig. 5-7 show the performances of RHEs related with μ over MS8, TS10 and CS10, under three different settings of λ_1, λ_2 and k corresponding

to those values of best performance on each dataset. In fig.6 and 7, we can observe two opposite trends that the performances in fig. 7 (CS10) decline consistently as μ increases which correspond to the reduction of involved community information from $\mathbf{U}$, while the performances in fig. 6 (TS10) raise along with μ. Though there are not clear monotonous trends shown in fig. 5, we still can observe enhancements brought by a certain proportion of community information when μ increases. However, too much community information will either not help, or it may even reduce momentum.

Hierarchy parameter k affects the structure of hierarchy. We select results on TS10 and MS8 to display below. Fig. 8-9 show the performances of topic detection followed by fig. 10-11 that show the results of document clustering. We can see that in most cases, the two types of metrics vary consistently with each other; that is to say, good result of document clustering corresponds to the good performance of topic detection. For example, on TS10, all methods obtain the best performances when $k = 2$, and RHE_MUL21 get the lowest value of all metrics when $k = 5$, while RHE_MULF reaches the worst performance when $k = 10$. Intuitively, k depends on the data distribution of the original data space. However, some of our results show less coordinated with datasets which might be influenced by the selection of other parameters. For example, on MS8, only RHE_ALM always obtains best results when $k = 2$, while RHE_MUL21 performs best in NDCG and MAP when $k = 3$, but does not maintain the edge in document clustering results.

5.7 Conclusions

In this Chapter, we propose a robust hierarchical ensemble NMF framework for topic detection through hierarchical document clustering and collective NMF in seeking a reduction of the effect caused by outliers and semantic diversity. To better adapt to the data distribution in the latest coming corpus, our method uses a dynamic cluster number to build an adjustable k-ary tree for hierarchy. Further pruning and outlier verification strategies are introduced. To enhance the robustness of NMF, we propose our objective function with both $\ell_{2,1}$-norm and F-norm, adopting two different optimization manners, augmented Lagrangian multiplier and Multiplicative updating rules, and deriving RHE series methods, RHE_ALM, RHE_MUL21 and RHE_MULF. Experiments on both syn-

thetic and real datasets including different application scenarios validate the robustness and effectiveness of our RHE series methods.

For future work, we intend to embed our ensemble framework to document and corresponding online comments streams to explore the changing trend of data distribution in continuous text data.

Chapter 6
Conclusions and Future Directions

This chapter summarises the contributions of the thesis towards enhancing robust detecting techniques for text content in the online social network context. Possible research extensions and directions are also discussed base on the conclusion of research findings and working experiences.

6.1 Summary and Conclusions

Text mining techniques are oriented around unstructured data that is widely distributed in peoples' daily life, revealing meaningful knowledge and associations from the mass of human natural language materials which is complicated and less regular. The advent of social network services facilitating interactions among people even poses more significant challenges to existing text mining models as the content and public sentiment of the information are more widespread and evolve quickly. More types of additional information will be generated and accessed through the social network which help broaden the scope of data and promote the development of methodologies applied to it. On the other hand, robustness gradually becomes the major concern of text mining influenced by the continuously changing social context.

In this thesis, we first introduced the background of this project in terms of text mining and social network analysis in Chapter 1, followed by the identification of specific challenges that have arisen in this developmental stage of text mining surrounded by online social network context. The challenges are broken into three major aspects, including challenges in the vast dynamic data universe, challenges in combining textual content with the social network context and challenges in achieving robust performance, plus additional thoughts about other challenges. The research problems and key contributions are then listed and explained accordingly. The outline is also illustrated at the end

of Chapter 1 to give an overall structure of the thesis. Chapter 2 presented a literature review for the existing works of topic models and document clustering and classification, which are the two core tasks of text mining this thesis focuses on, as well as relevant techniques of social network analysis and stream processing. Chapter 3 to Chapter 5 presented the specific contributions with theoretical analysis and experimental verifications in details, which target the research problems from the following components:

- To identify the changing distribution in an imbalanced data stream and maintain an up-to-date classifier for it, we proposed a Multi-Window based Ensemble Learning (MWEL) framework in Chapter 3 for imbalanced streaming data which comprehensively improves the classification performance. The principal part is a multi-window monitoring mechanism that maintains four windows for the current batch of instances, latest positive instances, sub-classifiers of the ensemble classifier and instances employed to train existing sub-classifiers, respectively. The sensitivity of the MWEL is guaranteed by an effective updating strategy for weights corresponding to existing sub-classifiers and an efficient updating strategy for sub-classifiers which keep the ensemble classifier being adaptive to the evolving data distribution and up-to-date when a change of data concept is detected and renewal is necessary. The imbalanced issue is tackled by an impartial re-sampling mechanism for both positive and negative instances which generate a new training set with an ideal imbalance ratio for updating sub-classifiers. Experiments conducted on real datasets covering five different application scenarios and synthetic datasets created following three distributions demonstrate that the proposed multi-windows method can efficiently and effectively classify imbalanced streaming data with outstanding performances across comprehensive evaluation criteria compared to baseline approaches.

- In Chapter 4, we proposed a semi-supervised collective learning method for topic detection combining the text content of the input corpus with the corresponding social context to combine existing text content representation with the social context in network individual level and to extract the inherent geometric structure from the data distribution and maximise the effect. Specifically, a collective non-negative matrix factorization based topic model is introduced to connect the corpus, the tex-

tual content and its social context. The network individuals are organised by a user preference representation according to their reactions to a collected document in an observation period. A constraint propagation scheme was designed to adequately exploit and enhance the hidden geometrical structure of original data space. Since the hidden geometrical structure is naturally very sparse and weak, it always cannot be thoroughly discovered and used without enhancement. The scheme propagates pairwise constraint between data points in both vertical and horizontal directions through the whole data space and generates weights for data point pair; then the weights will be used to minimise distances between two data points in the following approximation procedure of the NMF. Experiments conducted on real and semi-synthetic data sets demonstrate that the proposed method outperforms baselines and state-of-the-art approaches in most cases of topic detection and feature selection for further documents clustering with k-NN algorithm. At that time, we have noticed that some outliers may easily dominated the F-norm based objective function during the optimisation process; therefore, we extended the NMFCP with a locally weighted scheme to seek better approximation of certain parts of the data matrix in each iteration.

- In Chapter 5, to adaptively detect robust correct topics in circumstance with sever outliers and noise issue and obtain topics with more specific information, we proposed a Robust Hierarchical Ensemble (RHE) framework for topic detection via document hierarchy in text corpus and the corresponding social context. A robust multiplicative updating rule based NMF algorithm with an orthonormal constraint added on the output cluster indicator matrix is developed for document clustering so that detected topics can be generated at the same time. To enhance robustness, on the one hand, we conducted a comparative analysis in objective function level for the severe outliers and noise issue in complicated social context and high-dimensional text data. On the other hand, we designed a top-down hierarchical algorithm, including the candidature selection policy, the pruning strategy, the verification of outliers and two practical stopping criteria, which also facilitate the robustness of the ensemble framework. The proposed method that constructs document hierarchy structure also can flexibly cluster documents to adapt to the

changing distribution of the input corpus, achieving a logical and specific interpretation for clusters of the output. We also discussed how to extract the most valuable social context with experiments by selecting a subset of relevant network individuals for the purpose of removing some noises from the surroundings and efficiency. Through extensive experiments, RHE framework exhibits the robust and remarkable performance for both topic detection and document clustering.

6.2 Possible Future Directions

This section gives some insights into promising research directions and problems, which are core tasks of text mining but not restricted to the range of topic detection and document clustering and classification. As the scope of text mining has been broadened by social network services, although much attention has been devoted to tasks of text mining influenced by the ever-changing social context, there are still several gaps that are possible for extending the work presented in this thesis.

6.2.1 Hierarchy Evolving Tracking in Sequential Time Steps

The current hierarchy structure is constructed on the whole set of the corpus at a time step containing all input documents. This corpus is actually a segmentation of the continuous data stream according to time with the batch method of stream processing. However, the segmentation may break the consistency lying in the stream, troubling the model of topic evolving and fading process within consecutive time steps, worse yet, hindering the evolutionary information been taken advantage of. Besides, the hierarchy reflects the clustering process of documents, where each of the nodes indicates a collection of documents that share similar topics. And therefore, only nodes in the bottom level can be treated as the final classes of documents, revealing topics accordingly which are completely depend on the orthonormal relation between cluster indicators. However, although it is not difficult to track the changes of document concepts, it is hard to arrange the evolving with the existing hierarchy when a new corpus arrives in the next time step. In other words, the evolution of the corpus hierarchy has not been considered, while the information that may describe the connection or the trend of topics between two consecutive time steps

has not been involved in our current framework. Since the robustness enhanced by the hierarchy structure has been validated, we will keep working on this hierarchical model, considering formulating relation between two consecutive hierarchies with a concise evolution term that expresses the changing of concepts within the corpus. The items of this evolution term should be capable of interpreting the trace of the evolution as well as deducing the corresponding changes from the social context. Of course, it will complicate the model, increasing time and space consumption. The efficiency issue will be discussed in the next subsection.

6.2.2 Efficiency Issue of the Hierarchical model

The complexity of building the current hierarchy is $O(K-k)$. Then considering the integrated collective NMF algorithm and the constraint propagation for geometric structure, the overall complexity of the RHE framework is $O((K-k)\cdot(tmnk+tnlk+n^2m+l^2n))$, where n, m and l are numbers of documents, features and involved active users, respectively. There are two components of the RHE framework which may cause the efficiency issue. The first one is the constraint propagation across all data points, the result of which is presented by a sparse matrix, however, the computing process cannot avoid generating matrices of intermediate processes which are space consuming. This problem can be tackled with a more refined social content selection strategy to some extent. The strategy was discussed in Chapter 5 with an initial method proposed in Section 5.6.4, through which the most valuable social context can be filtered and this procedure can be implemented with parallel computing since it is independent of the hierarchy construction and NMF algorithm. The second step that may cause more time consumption is the construction of hierarchy, where multiple trials clustering are required for candidature selection and verification for outliers may be needed that will invoke the function of constraint propagation. However, as we realised that outliers and noises are top problems that threaten the robustness of text mining model in the continuously changing context, we plan to solve this problem in the future by finding a more efficient implementation method to construct hierarchy and update hierarchy for topic evolving. There is also some effort can be made to optimise the quick approximation of robust manifold NMF algorithm.

6.2.3 Content Evolving-base Dynamic Community Detection

Recently, more research works discuss the topic detection and tracking in the social network analysis field with a fine-grained term, event detection or even extraction. From the literature, we can hardly differentiate the two terms "topic" and "event" and find many works taking them as the same thing and named a task as "Topic Events Detection" [83, 97, 125]. We regard the event as a fine-grained level of topic. Normally, it is used more often for the short text mining in social network analysis, telling stories from different aspect with the same subjects, location and objects; while the content of a topic evolves quickly that may jump to other events with the same concerns. For example the tragic event of "Indonesia lion air crash on 29 October 2018" can evolve to "the development and operation of budget airlines", "the competition of aircraft companies", "the design of aircraft types" and even the "the search and rescue efforts"; while the evolution of the event will around the event itself, for example, "the black box", "the injuries and deaths". Therefore, we can say that both topic detection and event detection care about the content of a story and its evolution but serve different needs. Once we make the difference clear, we can detect and analyse the community structure of networks for different purposes, and even can help us to extract the needed social context better. Online communities consist of a variety of network individuals under the scope of social network analysis providing insights into how social influence spread within or beyond a certain range of the network and how does the network function. As the group of individual preferences used to present social context in our work helps in detecting topics, the evolution law of communities will also benefit from tracking the topic evolving on the topic or event level. To discover different types of online community, we can focus on topic level or event level, and the corresponding communities will offer more precise and valuable social context in turn.

6.2.4 Content Evolving-base Sentiment Prediction

Sentiment analysis delivers the latent attitudes in the text which may varying quickly in the social network. Traditional sentiment analysis in long text towards a document is statistic data-oriented, which is short on catching the changes among multiple kinds

of sentiment quickly and revealing what causes these changes and always roughly classify the sentiment to three statuses, positive, negative and neutral. Recently, sentiment analysis in the social network has been emphasised with the group intelligence, which claims that network individuals would be influenced by the information they received, often unconsciously, at the time they present their own opinions, especially when using the online social network services. The latent sentiment information spreads along with the social network structure. Mike Thelwall analysed over two million public comments associated with 2,990 pairs of U.K. and U.S. in 'MySpace Friends', then found statistically significant evidence for a weak correlation between the strength of positive emotion exchanged between Friends, which verified that sentiment is contagious and people tend to be a friend of others with similar levels of feeling [162]. Chenhao Tao et al. proposed a model based on their empirically confirmed assumption that connected users are more likely to have similar opinions which utilised the relationship between users of Twitter, including the follower-followee and homophily, to complement their utterances. The group intelligence also inspires us to connect the evolution of topic, event and online community with the prediction of sentiment spreading across the social network, making the relation between status of topics and events comprehensible and having things under control, what needs more works on monitoring and detecting the emerging, evolving and transforming of the sentiment. The detection, analysis and prediction of latent sentiment are also closely related work to social context extraction for the specific corpus. Therefore, it is regarded as part of our future directions.

6.2.5 Word Embedding and Topic Detection

As we mentioned in Chapter 2, almost all widely employed topic models are BoW-based methods that ignore the ordering of words and semantic information among the word sequence. It has been a serious bottleneck in improving the performance of topic models. Recently, some works have come rather to the front that integrates word embeddings to topic models by replacing discrete topic distribution in traditional LSA and LDA models with a multivariate Gaussian distribution on the embedding space [39, 98, 189]. Shi et al. claims that distributed Representation models and Latent topic models are complementary in how they represent the meaning of word occurrences, but some previous

works either using word embeddings to improve the quality of latent topics or using the latent topic model to improving word embeddings cannot take advantage of the mutual interaction between them [146]. They proposed a unified framework to learn word embeddings and latent topics simultaneously which inspired us to consider a model for the mutual benefit.

6.3 Final Remarks

Text mining reveals concepts, topics, and other meaningful knowledge in large collections of human natural language resources, helping people identify latent facts and relationships from the mass of big textual data. This research investigates challenges posed by the complicated online social network context in recent years, presenting novel algorithms for robust text mining model towards continuous changing context. The research outcome of this thesis is instrumental in promoting research and innovation of extensive text mining techniques influenced by the development of online social network continuing apace.

Bibliography

[1] C. C. Aggarwal and C. Zhai, "A Survey of Text Clustering Algorithms," in *Mining Text Data*, C. C. Aggarwal and C. Zhai, Eds. Boston, MA: Springer US, 2012, pp. 77–128.

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, E. P. Xing, and T. Jaakkola, "Mixed membership stochastic block models for relational data with application to protein-protein interactions," in *In Proceedings of the International Biometrics Society Annual Meeting*, 2006.

[3] C. Alippi, G. Boracchi, and M. Roveri, "Just in time classifiers: Managing the slow drift case," in *International Joint Conference on Neural Networks, 2009. IJCNN 2009*, Jun. 2009, pp. 114–120.

[4] I. Alsmadi and I. Alhami, "Clustering and classification of email contents," *Journal of King Saud University - Computer and Information Sciences*, vol. 27, no. 1, pp. 46–57, Jan. 2015.

[5] L. AlSumait, D. Barbar, and C. Domeniconi, "On-line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking," in *2008 Eighth IEEE International Conference on Data Mining*, Dec. 2008, pp. 3–12.

[6] R. anculef, I. Flaounas, and N. Cristianini, "Efficient classification of multi-labeled text streams by clashing," *Expert Systems with Applications*, vol. 41, no. 11, pp. 5431–5450, Sep. 2014.

[7] D. Andrzejewski and X. Zhu, "Latent Dirichlet Allocation with Topic-in-set Knowledge," in *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning*

for Natural Language Processing, ser. SemiSupLearn '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 43–48.

[8] T. Anwar and M. Abulaish, "A Social Graph Based Text Mining Framework for Chat Log Investigation," *Digit. Investig.*, vol. 11, no. 4, pp. 349–362, Dec. 2014.

[9] V. D. Badal, P. J. Kundrotas, and I. A. Vakser, "Natural language processing in text mining for structural modeling of protein complexes," *BMC Bioinformatics*, vol. 19, no. 1, p. 84, Mar. 2018.

[10] Q. Bai, Q. Hu, F. Fang, and L. He, "Topic Detection with Danmaku: A Time-Sync Joint NMF Approach," in *Database and Expert Systems Applications*, ser. Lecture Notes in Computer Science, S. Hartmann, H. Ma, A. Hameurlain, G. Pernul, and R. R. Wagner, Eds. Springer International Publishing, 2018, pp. 428–435.

[11] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2014, pp. 238–247.

[12] S. Beliga, A. Metrovi, and S. Martini-Ipi, "An Overview of Graph-Based Keyword Extraction Methods and Approaches," *Journal of Information and Organizational Sciences*, vol. 39, no. 1, pp. 1–20, Jun. 2015.

[13] Y. Bengio, R. Ducharme, and P. Vincent, "A Neural Probabilistic Language Model," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, ser. NIPS'00. Cambridge, MA, USA: MIT Press, 2001, pp. 893–899.

[14] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A Neural Probabilistic Language Model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[15] D. P. Bertsekas, *Constrained optimization and lagrange multiplier methods*, 1982.

[16] A. Bifet and R. Gavald, "Learning from time-changing data with adaptive windowing," in *In SIAM International Conference on Data Mining*, 2007.

[17] A. Bifet and R. Gavaldà, "Adaptive Learning from Evolving Data Streams," in *Advances in Intelligent Data Analysis VIII*, N. M. Adams, C. Robardet, A. Siebes, and

J.-F. Boulicaut, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 249–260.

[18] D. M. Blei, "Probabilistic Topic Models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.

[19] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 113–120.

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press.

[22] R. B. Bradford, "Application of Latent Semantic Indexing in Generating Graphs of Terrorist Networks," in *Intelligence and Security Informatics*, ser. Lecture Notes in Computer Science, S. Mehrotra, D. D. Zeng, H. Chen, B. Thuraisingham, and F.-Y. Wang, Eds. Springer Berlin Heidelberg, 2006, pp. 674–675.

[23] D. Cai, X. He, and J. Han, "Locally Consistent Concept Factorization for Document Clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 902–913, Jun. 2011.

[24] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[25] B. Cao, D. Shen, J.-T. Sun, X. Wang, Q. Yang, and Z. Chen, "Detect and Track Latent Factors with Online Nonnegative Matrix Factorization." in *IJCAI*, vol. 7, 2007, pp. 2689–2694.

[26] S. Castano, A. Ferrara, and S. Montanelli, "Exploratory analysis of textual data streams," *Future Generation Computer Systems*, vol. 68, pp. 391–406, Mar. 2017.

[27] S. Chakravarthy, A. Venkatachalam, and A. Telang, "A Graph-Based Approach for Multi-folder Email Classification," in *2010 IEEE International Conference on Data Mining*, Dec. 2010, pp. 78–87.

[28] N. V. Chawla, "Data Mining for Imbalanced Datasets: An Overview," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer US, 2005, pp. 853–867.

[29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002.

[30] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting," in *Knowledge Discovery in Databases: PKDD 2003*, ser. Lecture Notes in Computer Science, N. Lavra, D. Gamberger, L. Todorovski, and H. Blockeel, Eds. Springer Berlin Heidelberg, 2003, no. 2838, pp. 107–119.

[31] P. Chen, N. L. Zhang, L. K. M. Poon, and Z. Chen, "Progressive EM for Latent Tree Models and Hierarchical Topic Detection," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. Phoenix, Arizona: AAAI Press, 2016, pp. 1498–1504.

[32] S. Chen and H. He, "Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach," *Evolving Systems*, vol. 2, no. 1, pp. 35–50, Nov. 2010.

[33] J. Choo, C. Lee, C. K. Reddy, and H. Park, "UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 1992–2001, Dec. 2013.

[34] P.-H. Chou, R. T.-H. Tsai, and J. Y.-j. Hsu, "Context-aware sentiment propagation using LDA topic modeling on Chinese ConceptNet," *Soft Computing*, vol. 21, no. 11, pp. 2911–2921, Jun. 2017.

[35] T. Chou and M. C. Chen, "Using Incremental PLSI for Threshold-Resilient Online Event Analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 3, pp. 289–299, Mar. 2008.

[36] B. Choudhary and P. Bhattacharyya, "Text Clustering using Semantics," Hawaii, USA, 2002, p. 4.

[37] D. Cohn and T. Hofmann, "The missing link-a probabilistic model of document content and hypertext connectivity," *Advances in neural information processing systems*, pp. 430–436, 2001.

[38] R. Collobert and J. Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 160–167.

[39] R. Das, M. Zaheer, and C. Dyer, "Gaussian LDA for Topic Models with Word Embeddings," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, 2015, pp. 795–804.

[40] V. Dasondi, M. Pathak, and N. P. Singh, "An implementation of graph based text classification technique for social media," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, Mar. 2016, pp. 1–7.

[41] S. Deerwester, S. T. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing By Latent Semantic Analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391–407, Sep. 1990.

[42] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle, "A Case-Based Technique for Tracking Concept Drift in Spam Filtering," in *Applications and Innovations in Intelligent Systems XII*, P. A. M. B. CEng, R. E. B. MSc, and D. T. Allen, Eds. Springer London, 2005, pp. 3–16.

[43] L. Deng, B. Xu, L. Zhang, Y. Han, B. Zhou, and P. Zou, "Tracking the evolution of public concerns in social media," in *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service*. ACM, 2013, pp. 353–357.

[44] C. Ding, T. Li, and W. Peng, "On the equivalence between Non-negative Matrix Factorization and Probabilistic Latent Semantic Indexing," *Computational Statistics & Data Analysis*, vol. 52, no. 8, pp. 3913–3927, Apr. 2008.

[45] C. H. Q. Ding and X. He, "Cluster merging and splitting in hierarchical clustering algorithms," in *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, 2002, pp. 139–146.

[46] C. H. Q. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix T-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 126–135.

[47] C. H. Q. Ding, D. Zhou, X. He, and H. Zha, "R_1-PCA: Rotational invariant L_1-norm principal component analysis for robust subspace factorization," in *Proceedings of the 23rd International Conference on Machine Learning*.

[48] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, 2014, pp. 2083–2090.

[49] P. Domingos and G. Hulten, "Mining High-speed Data Streams," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '00. New York, NY, USA: ACM, 2000, pp. 71–80.

[50] R. Du, D. Kuang, B. Drake, and H. Park, "DC-NMF: nonnegative matrix factorization based on divide-and-conquer for fast clustering and topic modeling," *Journal of Global Optimization*, vol. 68, no. 4, pp. 777–798, Aug. 2017.

[51] S. T. Dumais, "Latent Semantic Indexing (LSI): TREC-3 Report," in *Proceedings of the Text REtrieval Conference (TREC-3)*, 1995, pp. 219–230.

[52] S. T. Dumais and J. Nielsen, "Automating the Assignment of Submitted Manuscripts to Reviewers," in *In Research and Development in Information Retrieval*. ACM Press, 1992, pp. 233–244.

[53] R. Elwell and R. Polikar, "Incremental Learning of Concept Drift in Nonstationary Environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011.

[54] G. Erkan and D. R. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, Dec. 2004.

[55] E. Erosheva, S. Fienberg, and J. Lafferty, "Mixed-membership models of scientific publications," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5220–5227, Apr. 2004.

[56] R. Feldman, "Techniques and Applications for Sentiment Analysis," *Commun. ACM*, vol. 56, pp. 82–89, Apr. 2013.

[57] J. R. Firth, "The Technique of Semantics." *Transactions of the Philological Society*, vol. 34, no. 1, pp. 36–73, Nov. 1935.

[58] ——, *A Synopsis of Linguistic Theory, 1930-1955*. Oxford: Oxford University Press, 1957, vol. In Special Volume of the Philological Society., google-Books-ID: T8LDtgAACAAJ.

[59] J. Foulds, S. Kumar, and L. Getoor, "Latent topic networks: A versatile probabilistic programming framework for topic models," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 777–786.

[60] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, Jun. 2005.

[61] M. Gamon, "Graph-Based Text Representation for Novelty Detection," in *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing on the First Workshop on Graph Based Methods for Natural Language Processing - TextGraphs '06*. New York, NY, USA: New York, NY, USA, Jun. 2006, pp. 17–24.

[62] E. Gaussier and C. Goutte, "Relation Between PLSA and NMF and Implications," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '05. New York, NY, USA: ACM, 2005, pp. 601–602.

[63] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 4–es, Mar. 2007.

[64] A. Hassan and A. Mahmood, "Deep Learning approach for sentiment analysis of short texts," in *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, Apr. 2017, pp. 705–710.

[65] H. Hassan, A. Hassan, and S. Noeman, "Graph based semi-supervised approach for information extraction," in *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing on the First Workshop on Graph Based Methods for Natural Language Processing - TextGraphs '06*. New York, NY, USA: Association for Computational Linguistics, Jun. 2006, pp. 9–16.

[66] C. He, T. Zhuo, D. Ou, M. Liu, and M. Liao, "Nonlinear Compressed Sensing-Based LDA Topic Model for Polarimetric SAR Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 3, pp. 972–982, Mar. 2014.

[67] H. He and E. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[68] S. Hensman, "Construction of Conceptual Graph Representation of Texts," in *HLT-NAACL 2004: Student Research Workshop*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2004, pp. 49–54.

[69] G. E. Hinton, *Learning distributed representations of concepts*, ser. Parallel distributed processing: Implications for psychology and neurobiology. New York, NY, US: Clarendon Press/Oxford University Press, 1989.

[70] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 89–101, Jan. 2012.

[71] P. Hoffman, M. A. L. Ralph, and T. T. Rogers, "Semantic diversity: A measure of semantic ambiguity based on variability in the contextual usage of words," *Behavior Research Methods*, vol. 45, no. 3, pp. 718–730, 2013.

[72] T. Hofmann, "Probabilistic Latent Semantic Indexing," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99. New York, NY, USA: ACM, 1999, pp. 50–57, citation Key: hofmann_probabilistic_1999.

[73] ——, "Unsupervised Learning by Probabilistic Latent Semantic Analysis," *Machine Learning*, vol. 42, no. 1-2, pp. 177–196, Jan. 2001.

[74] W. Hong, X. Zheng, J. Qi, W. Wang, and Y. Weng, "Project Rank: An Internet Topic Evaluation Model Based on Latent Dirichlet Allocation," in *2018 13th International Conference on Computer Science Education (ICCSE)*, Aug. 2018, pp. 1–4.

[75] P. O. Hoyer, "Non-negative Matrix Factorization with Sparseness Constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Dec. 2004.

[76] J. Huang, F. Nie, H. Huang, and C. H. Q. Ding, "Robust manifold nonnegative matrix factorization," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 3, pp. 11:1–11:21, 2014.

[77] S. Huang, H. Wang, T. Li, T. Li, and Z. Xu, "Robust graph regularized nonnegative matrix factorization for clustering," *Data Mining and Knowledge Discovery*, vol. 32, no. 2, pp. 483–503, Mar. 2018.

[78] G. Hulten, L. Spencer, and P. Domingos, "Mining Time-changing Data Streams," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 97–106.

[79] J. L. Hurtado, A. Agarwal, and X. Zhu, "Topic discovery and future trend forecasting for texts," *Journal of Big Data*, vol. 3, no. 1, p. 7, Apr. 2016.

[80] M. L. Joshi, "A Review on Semantic Graph based Text Mining," *International Research Journal of Computers and Electronics Engineering (IRJCEE)*, vol. 3, p. 5, 2015.

[81] J. Kalyanam, A. Mantrach, D. Sáez-Trumper, H. Vahabi, and G. R. G. Lanckriet, "Leveraging social context for modeling topic evolution," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 517–526.

[82] S. P. Kasiviswanathan, H. Wang, A. Banerjee, and P. Melville, "Online L1-Dictionary Learning with Application to Novel Document Detection," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 2258–2266, citation Key: kasiviswanathan_online_2012.

[83] N. Keane, C. Yee, and L. Zhou, "Using Topic Modeling and Similarity Thresholds to Detect Events," in *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*. Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 34–42.

[84] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751.

[85] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware Neural Language Models," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. Phoenix, Arizona: AAAI Press, 2016, pp. 2741–2749.

[86] J. Z. Kolter and M. A. Maloof, "Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts," *J. Mach. Learn. Res.*, vol. 8, pp. 2755–2790, Dec. 2007.

[87] D. Kong, C. H. Q. Ding, and H. Huang, "Robust nonnegative matrix factorization using l21-norm," in *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM 2011)*, 2011, pp. 673–682.

[88] T. Koo, X. Carreras, and M. Collins, "Simple Semi-supervised Dependency Parsing," in *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, 2008, pp. 595–603.

[89] A. Krause, "Robust L1 Norm Factorization in the Presence of Outliers and Missing Data by Alternative Convex Programming," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, Jun. 2015, citation Key: krause_robust_nodate.

[90] D. Kuang and H. Park, "Fast rank-2 nonnegative matrix factorization for hierarchical document clustering," in *Proceedings of the 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2013)*, 2013, pp. 739–747.

[91] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological review*, vol. 104, no. 2, p. 211, 1997.

[92] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, no. 2-3, pp. 259–284, Jan. 1998.

[93] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[94] ——, "Algorithms for Non-negative Matrix Factorization," in *In NIPS*. MIT Press, 2000, pp. 556–562.

[95] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: a new benchmark collection for text categorization research," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[96] C. Li, A. Sun, and A. Datta, "Twevent: Segment-based Event Detection from Tweets," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, ser. CIKM '12. New York, NY, USA: ACM, 2012, pp. 155–164.

[97] C. Li, Y. Ye, X. Zhang, D. Chu, S. Deng, and X. Xu, "Clustering Based Topic Events Detection on Text Stream," in *Intelligent Information and Database Systems*, ser. Lecture Notes in Computer Science, N. T. Nguyen, B. Attachoo, B. Trawiski, and K. Somboonviwat, Eds. Springer International Publishing, 2014, pp. 42–52.

[98] X. Li, J. Chi, C. Li, J. Ouyang, and B. Fu, "Integrating Topic Modeling with Word Embeddings by Mixtures of vMFs," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 151–160.

[99] X. Li, L. Guo, and Y. E. Zhao, "Tag-based Social Interest Discovery," in *Proceedings of the 17th International Conference on World Wide Web*, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 675–684.

[100] R. N. Lichtenwalter and N. V. Chawla, "Adaptive Methods for Classification in Arbitrarily Imbalanced and Drifting Data Streams," in *New Frontiers in Applied Data Mining*, ser. Lecture Notes in Computer Science, T. Theeramunkong, C. Nattee, P. J. L. Adeodato, N. Chawla, P. Christen, P. Lenca, J. Poon, and G. Williams, Eds. Springer Berlin Heidelberg, 2010, no. 5669, pp. 53–75.

[101] D. Lin and X. Wu, "Phrase Clustering for Discriminative Learning," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Computational Linguistics, 2009, pp. 1030–1038.

[102] H. Liu, Z. Wu, D. Cai, and T. S. Huang, "Constrained nonnegative matrix factorization for image representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1299–1311, 2012.

[103] W. Liu, S. Chawla, D. A. Cieslak, and N. V. Chawla, "A robust decision tree algorithm for imbalanced data sets," in *in SIAM International Conference on Data Mining, 2010*, 2010, pp. 766–777.

[104] W. Liu, L. Wang, and M. Yi, "Simple-Random-Sampling-Based Multiclass Text Classification Algorithm," *The Scientific World Journal*, vol. 2014, pp. 1–7, 2014.

[105] X. Liu, W. Wang, D. He, P. Jiao, D. Jin, and C. V. Cannistraci, "Semi-supervised community detection based on non-negative matrix factorization with node popularity," *Information Sciences*, vol. 381, pp. 304–321, Mar. 2017.

[106] Y. Lu, Q. Mei, and C. Zhai, "Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA," *Information Retrieval*, vol. 14, no. 2, pp. 178–203, Apr. 2011.

[107] Z. Lu and H. H. Ip, "Constrained spectral clustering via exhaustive and efficient constraint propagation," in *Proceedings of the 11th European Conference on Computer Vision (ECCV 2010)*, 2010, pp. 1–14.

[108] X. Luo, J. Xuan, J. Lu, and G. Zhang, "Measuring the Semantic Uncertainty of News Events for Evolution Potential Estimation," *ACM Trans. Inf. Syst.*, vol. 34, no. 4, pp. 24:1–24:25, Jun. 2016.

[109] X. Ma and D. Dong, "Evolutionary Nonnegative Matrix Factorization Algorithms for Community Detection in Dynamic Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 5, pp. 1045–1058, May 2017.

[110] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.

[111] T. Mikolov, M. Karafit, L. Burget, J. Cernock, and S. Khudanpur, "Recurrent neural network based language model," vol. 2, Jan. 2010, pp. 1045–1048.

[112] J. Miller, D. Gefen, and V. K. Narayanan, "Seeing the Forest: Applying Latent Semantic Analysis to Smartphone Discourse," *BLED 2016 Proceedings*, Jan. 2016.

[113] S. Miller, J. Guinness, and A. Zamanian, "Name tagging with word clusters and discriminative training," in *Proceedings of HLT-NAACL*, 2004, pp. 337–342.

[114] M. M. Miroczuk and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," *Expert Systems with Applications*, vol. 106, pp. 36–54, Sep. 2018.

[115] G. Mishne and N. Glance, "Leave a Reply: An Analysis of Weblog Comments," in *In Third annual workshop on the Weblogging ecosystem*, 2006.

[116] B. Mitra, F. Diaz, and N. Craswell, "Learning to Match using Local and Distributed Representations of Text for Web Search," in *Proceedings of the 26th International Conference on World Wide Web - WWW '17*. Perth, Australia: ACM Press, 2017, pp. 1291–1299.

[117] A. Mnih and G. Hinton, "Three New Graphical Models for Statistical Language Modelling," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 641–648.

[118] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen, "Joint latent topic models for text and citations," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 542–550.

[119] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in *Proceedings of the 21st International Conference on Machine Learning*, 2004, pp. 78–, citation Key: ng_feature_2004.

[120] S. Osiski, J. Stefanowski, and D. Weiss, "Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition," in *Intelligent Information Processing and Web Mining*, M. A. Kopotek, S. T. Wierzcho, and K. Trojanowski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 359–368, citation Key: 10.1007/978-3-540-39985-8_37.

[121] J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan, "Nested Hierarchical Dirichlet Processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 256–270, Feb. 2015.

[122] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in Social Media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, May 2012.

[123] N. Papanikolaou, G. A. Pavlopoulos, T. Theodosiou, and I. Iliopoulos, "Protein-protein interaction predictions using text mining methods," *Methods (San Diego, Calif.)*, vol. 74, pp. 47–53, Mar. 2015.

[124] Y. Papanikolaou and G. Tsoumakas, "Subset Labeled LDA: A Topic Model for Extreme Multi-label Classification," in *Big Data Analytics and Knowledge Discovery*, ser. Lecture Notes in Computer Science, C. Ordonez and L. Bellatreche, Eds. Springer International Publishing, 2018, pp. 152–162.

[125] R. Papka and J. Allan, "Topic Detection and Tracking: Event Clustering as a Basis for First Story Detection," in *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*, ser. The Information Retrieval Series, W. B. Croft, Ed. Boston, MA: Springer US, 2000, pp. 97–126.

[126] P. Parveen, Z. R. Weger, B. Thuraisingham, K. Hamlen, and L. Khan, "Supervised Learning for Insider Threat Detection Using Stream Mining," in *Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, ser. ICTAI '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1032–1039.

[127] C. Patel and M. Gadhavi, "A Model for Document classification using Kernel Discriminant Analysis(KDA) and semantic analysis," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, pp. 783–786, Apr. 2017.

[128] M. Pavlinek and V. Podgorelec, "Text classification method based on self-training and LDA topic models," *Expert Systems with Applications*, vol. 80, pp. 83–93, Sep. 2017.

[129] X. Pei, C. Chen, and W. Gong, "Concept Factorization With Adaptive Neighbors for Document Clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 343–352, Feb. 2018.

[130] M. Peng, S. Ouyang, J. Zhu, J. Huang, H. Wang, and J. Yong, "Emerging Topic Detection from Microblog Streams Based on Emerging Pattern Mining*," in *2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))*, May 2018, pp. 259–264.

[131] M. Peng, J. Zhu, X. Li, J. Huang, H. Wang, and Y. Zhang, "Central topic model for event-oriented topics mining in microblog stream," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1611–1620.

[132] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543.

[133] F. Pereira, N. Tishby, and L. Lee, "DISTRIBUTIONAL CLUSTERING OF ENGLISH WORDS," in *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*. Columbus, Ohio, USA: Association for Computational Linguistics, Jun. 1993, pp. 183–190.

[134] P. Pham, P. Do, and C. D. C. Ta, "W-PathSim: Novel Approach of Weighted Similarity Measure in Content-Based Heterogeneous Information Networks by Applying LDA Topic Modeling," in *Intelligent Information and Database Systems*, ser. Lecture Notes in Computer Science, N. T. Nguyen, D. H. Hoang, T.-P. Hong, H. Pham, and B. Trawiski, Eds. Springer International Publishing, 2018, pp. 539–549.

[135] R. Polikar, L. Upda, S. Upda, and V. Honavar, "Learn++: an incremental learning algorithm for supervised neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 31, no. 4, pp. 497–508, Nov. 2001.

[136] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 1 - EMNLP '09*, vol. 1. Singapore: Association for Computational Linguistics, 2009, p. 248.

[137] K. Reing, D. C. Kale, G. V. Steeg, and A. Galstyan, "Toward interpretable topic discovery via anchored correlation explanation," *arXiv preprint arXiv:1606.07043*, 2016.

[138] R. Rifkin and A. Klautau, "In Defense of One-Vs-All Classification," *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, Dec. 2004.

[139] A. Saha and V. Sindhwani, "Learning evolving and emerging topics in social media: a dynamic NMF approach with temporal regularization," in *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, 2012, pp. 693–702.

[140] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.

[141] K. Sasaki, T. Yoshikawa, and T. Furuhashi, "Online topic model for Twitter considering dynamics of user interests and topic trends." in *EMNLP*, 2014, pp. 1977–1985.

[142] H. Sayyad, "Event Detection and Tracking in Social Streams," in *Proceedings of the Third International ICWSM Conference (2009)*, San Jose, CA, USA, May 2009, pp. 311–314.

[143] L. Sha and D. Schonfeld, "Dual graph regularized sparse coding for image representation," in *Proceedings of 2017 IEEE Visual Communications and Image Processing (VCIP)*, 2017, pp. 1–4.

[144] W. Shao, L. He, C. Lu, X. Wei, and P. S. Yu, "Online unsupervised multi-view feature selection," in *Proceedings of IEEE 16th International Conference on Data Mining (ICDM 2016)*, 2016, pp. 1203–1208.

[145] X. Shen, M. Boutell, J. Luo, and C. Brown, "Multilabel machine learning and its application to semantic scene classification," vol. 5307, Dec. 2003, pp. 188–199.

[146] B. Shi, W. Lam, S. Jameel, S. Schockaert, and K. P. Lai, "Jointly Learning Word Embeddings and Latent Topics," *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '17*, pp. 375–384, 2017, arXiv: 1706.07276.

[147] J. Shi, X. Tian, Z. Jiang, D. Zhao, and M. Lu, "Sparsity-constrained probabilistic latent semantic analysis for land cover classification," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2016, pp. 5453–5456.

[148] J. Shi and Z. Luo, "Nonlinear dimensionality reduction of gene expression data for visualization and clustering analysis of cancer tissue samples," *Computers in Biology and Medicine*, vol. 40, no. 8, pp. 723–732, Aug. 2010.

[149] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 650–658.

[150] R. Sinha and R. Mihalcea, "Unsupervised Graph-basedWord Sense Disambiguation Using Measures of Word Semantic Similarity," in *International Conference on Semantic Computing (ICSC 2007)*, Sep. 2007, pp. 363–369.

[151] H. Sperr, J. Niehues, and A. Waibel, "Letter N-Gram-based Input Encoding for Continuous Space Language Models," in *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 30–39.

[152] D. Spina, J. Gonzalo, and E. Amig, "Learning Similarity Functions for Topic Detection in Online Reputation Monitoring," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, ser. SIGIR '14. New York, NY, USA: ACM, 2014, pp. 527–536.

[153] S. S.Sonawane and P. A. Kulkarni, "Graph based Representation and Analysis of Text Document: A Survey of Techniques," *International Journal of Computer Applications*, vol. 96, no. 19, pp. 1–8, Jun. 2014.

[154] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, "Exploring Topic Coherence over Many Models and Many Topics," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 952–961.

[155] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, "Probabilistic author-topic models for information discovery." ACM, Aug. 2004, pp. 306–315.

[156] W. N. Street and Y. Kim, "A Streaming Ensemble Algorithm (SEA) for Large-scale Classification," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 377–382.

[157] S. Suh, J. Choo, J. Lee, and C. K. Reddy, "L-ensnmf: Boosted local topic discovery via ensemble of nonnegative matrix factorization," in *Proceedings of IEEE 16th International Conference on Data Mining (ICDM 2016)*, 2016, pp. 479–488.

[158] Y. Sun, A. K. C. Wong, and Y. Wang, "Parameter Inference of Cost-Sensitive Boosting Algorithms," in *Machine Learning and Data Mining in Pattern Recognition*, ser. Lecture Notes in Computer Science, P. Perner and A. Imiya, Eds. Springer Berlin Heidelberg, 2005, no. 3587, pp. 21–30.

[159] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112.

[160] J. Tang, R. Jin, and J. Zhang, "A Topic Modeling Approach and Its Integration into the Random Walk Framework for Academic Search," in *2008 Eighth IEEE International Conference on Data Mining*, Dec. 2008, pp. 1055–1060.

[161] D. Tao, D. Tao, X. Li, and X. Gao, "Large Sparse Cone Non-negative Matrix Factorization for Image Annotation," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 3, pp. 37:1–37:21, Apr. 2017.

[162] M. Thelwall, "Emotion homophily in social network site messages," *First Monday*, vol. 15, no. 4, Apr. 2010.

[163] I. Tomek, "Two modifications of CNN," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, 1976.

[164] L. Tsekouras, I. Varlamis, and G. Giannakopoulos, "A Graph-based Text Similarity Measure That Employs Named Entity Information," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., 2017, pp. 765–771.

[165] M. Tsytsarau, T. Palpanas, and K. Denecke, "Scalable discovery of contradictions on the web," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 1195–1196.

[166] D. Tu, L. Chen, M. Lv, H. Shi, and G. Chen, "Hierarchical Online NMF for Detecting and Tracking Topic Hierarchies in a Text Stream," *Pattern Recogn.*, vol. 76, no. C, pp. 203–214, Apr. 2018.

[167] S. Tuarob, S. Bhatia, P. Mitra, and C. L. Giles, "AlgorithmSeer: A System for Extracting and Searching for Algorithms in Scholarly Big Data," *IEEE Transactions on Big Data*, vol. 2, no. 1, pp. 3–17, Mar. 2016.

[168] C. K. Vaca, A. Mantrach, A. Jaimes, and M. Saerens, "A time-based collective factorization for topic discovery and monitoring in news." ACM Press, 2014, pp. 527–538.

[169] C. Wang, D. Blei, and D. Heckerman, "Continuous time dynamic topic models," *arXiv preprint arXiv:1206.3298*, 2012.

[170] D. Wang, F. Nie, and H. Huang, "Fast robust non-negative matrix factorization for large-scale human action data clustering," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 2016, pp. 2104–2110.

[171] D. Wang, X. Gao, and X. Wang, "Semi-Supervised Nonnegative Matrix Factorization via Constraint Propagation," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 233–244, Jan. 2016.

[172] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining Concept-drifting Data Streams Using Ensemble Classifiers," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03. New York, NY, USA: ACM, 2003, pp. 226–235.

[173] S. Wang, L. Minku, and X. Yao, "A learning framework for online class imbalance learning," in *2013 IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL)*, Apr. 2013, pp. 36–45.

[174] S. Wang, J. Tang, and H. Liu, "Embedded Unsupervised Feature Selection." in *AAAI*, 2015, pp. 470–476.

[175] X. Wang, Y. Jia, R. Chen, H. Fan, and B. Zhou, "Improving Text Categorization with Semantic Knowledge in Wikipedia," *IEICE TRANSACTIONS on Information and Systems*, vol. E96-D, no. 12, pp. 2786–2794, Dec. 2013.

[176] Y. Wang, Y. Zhang, B. Zhou, and Y. Jia, "Semi-supervised collective matrix factorization for topic detection and document clustering," in *Proceedings of IEEE 2nd International Conference on Data Science in Cyberspace (DSC 2017)*, 2017, pp. 88–97.

[177] K. Wegba, A. Lu, Y. Li, and W. Wang, "Interactive Movie Recommendation Through Latent Semantic Analysis and Storytelling," *arXiv preprint arXiv:1701.00199*, 2017.

[178] D. C. Wheeler, "Geographically Weighted Regression," in *Handbook of Regional Science*, M. M. Fischer and P. Nijkamp, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1435–1459.

[179] G. Widmer and M. Kubat, "Learning in the Presence of Concept Drift and Hidden Contexts," *Mach. Learn.*, vol. 23, no. 1, pp. 69–101, Apr. 1996.

[180] P. Xie, D. Yang, and E. Xing, "Incorporating Word Correlation Knowledge into Topic Modeling," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 725–734.

[181] W. Xie, F. Zhu, J. Jiang, E. Lim, and K. Wang, "TopicSketch: Real-Time Bursty Topic Detection from Twitter," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2216–2229, Aug. 2016.

[182] E. S. Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas, "Dealing with Concept Drift and Class Imbalance in Multi-label Stream Classification," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, ser. IJCAI'11. Barcelona, Catalonia, Spain: AAAI Press, 2011, pp. 1583–1588.

[183] 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 Informaion Retrieval*, ser. SIGIR '03. New York, NY, USA: ACM, 2003, pp. 267–273.

[184] C. Yang, J. Yang, H. Ding, and H. Xue, "A Hot Topic Detection Approach on Chinese Microblogging," in *Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012*, ser. Lecture Notes in Electrical Engineering, Z. Zhong, Ed. Springer London, 2013, pp. 411–420.

[185] P. Yang, X. Su, L. Ou-Yang, H.-N. Chua, X.-L. Li, and K. Ning, "Microbial community pattern detection in human body habitats via ensemble clustering framework," *BMC systems biology*, vol. 8 Suppl 4, p. S7, 2014.

[186] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical Attention Networks for Document Classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1480–1489.

[187] S. Yu, S. Van Vooren, L.-C. Tranchevent, B. De Moor, and Y. Moreau, "Comparison of vocabularies, representations and ranking algorithms for gene prioritization by text mining," *Bioinformatics (Oxford, England)*, vol. 24, no. 16, pp. i119–125, Aug. 2008.

[188] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[189] A. Zhang, G. Xun, Y. Li, Wayne Xin Zhao, and J. Gao, "A Correlated Topic Model Using Word Embeddings," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 4207–4213.

[190] D. Zhang, H. Shen, T. Hui, Y. Li, J. Wu, and Y. Sang, "A Selectively Re-train Approach Based on Clustering to Classify Concept-Drifting Data Streams with Skewed Distribution," in *Advances in Knowledge Discovery and Data Mining*, ser. Lecture Notes in Computer Science, V. S. Tseng, T. B. Ho, Z.-H. Zhou, A. L. P. Chen, and H.-Y. Kao, Eds. Springer International Publishing, May 2014, no. 8444, pp. 413–424.

[191] L. Zhang, Q. Zhang, B. Du, D. Tao, and J. You, "Robust manifold matrix factorization for joint clustering and feature extraction," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, 2017, pp. 1662–1668.

[192] L. Zhang, Q. Zhang, B. Du, J. You, and D. Tao, "Adaptive manifold regularized matrix factorization for data clustering," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*, 2017, pp. 3399–3405.

[193] X. Zhang, J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 649–657.

[194] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li, "Comparing Twitter and Traditional Media Using Topic Models," in *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ser. ECIR'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 338–349.

[195] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16*. MIT Press, 2004, pp. 321–328.

[196] F. Zhou, F. Zhang, and B. Yang, "Graph-based text representation model and its realization," in *Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering(NLPKE-2010)*, Aug. 2010, pp. 1–8.

[197] C. Zhu, H. Zhu, Y. Ge, E. Chen, Q. Liu, T. Xu, and H. Xiong, "Tracking the evolution of social emotions with topic models," *Knowledge and Information Systems*, vol. 47, no. 3, pp. 517–544, Jun. 2016.

[198] W. Y. Zou, R. Socher, D. Cer, and C. D. Manning, "Bilingual Word Embeddings for Phrase-Based Machine Translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1393–1398.