Pretopology and Topic Modeling for Complex Systems Analysis : Application on Document Classification and Complex Network Analysis

Quang Vu Bui

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à l'École Pratique des Hautes Études

Pretopology and Topic Modeling for Complex Systems Analysis: Application on Document Classification and Complex Network Analysis.

École doctorale de l'EPHE - ED 472

Spécialité INFORMATIQUE, STATISTIQUES ET COGNITION

COMPOSITION DU JURY :

M. Charle Tijus
Université Paris 8
Président du jury

M. Hacène Fouchal
Université de Reims Champagne-Ardenne
Rapporteur

M. Jean Frédéric Myoupo
Université de Picardie Jules Verne
Rapporteur

M. Tu Bao Ho
John von Neumann Institute, Vietnam
Rapporteur

M. Marc Bui
École Pratique des Hautes Études
Directeur de thèse

M. Soufian Ben Amor
Université de Versailles
Saint-Quentin-en-Yvelines
Membre du jury

Soutenue par **Quang Vu BUI**
le 27/09/2018

Dirigée par **Marc BUI**

École Pratique
des Hautes Études

PSL
RESEARCH UNIVERSITY PARIS

Acknowledgments

Without the influence and support of all people around me, I would not be able to achieve the doctorate degree in Compute science, Statistics and Cognition.

First and foremost I would like to thank my advisor, Professor Marc Bui. I knew I can not make this happen without his tremendous guidance and support during the past 4 years. Accomplishing a Ph.D. degree is never easy, and Professor Marc Bui's constant effort and encouragement in my academic research made this journey less difficult than I thought. The knowledge and skills that I have learned from him will continue to shape me in many aspects of my future life.

Secondly, I would like to thank my committee members, Prof. Hacène Fouchal, Prof. Myoupo Jean Frédéric, Professor Tu Bao HO, Professor Charles Tijus, Assis.Prof Soufian Ben Amor, Professor Michel Lamure for your precious time and valuable comments on my dissertation.

Additionally, I want to give thanks to my colleagues and friends at CHart Lab: Manuel, Karim, Julio, Anna, Jeremi, Kaan, Thoa, etc. Because of them, my Ph.D. life was even more precious and memorable. I also want to give thank to my Vietnamese friends for encourage me during the time I do PhD.

Special thanks goes to my family: my parents, two my younger brothers for offering me unconditional support during my bad and good times.

Words cannot express my infinite gratitude to my wife Thanh Hai, two my little girls: Lily and Paris for being a great support and for sharing the moments of joy and happiness during all time we are together. Thank you !

Contents

List of Figures

List of Tables

List of Acronyms

LSA	Latent Semantic Analysis
pLSA	Probabilistic Latent Semantic Analysis
LDA	Latent Dirichlet Allocation
ss-LDA	semi-supervised Latent Dirichlet Allocation
AD-LDA	Approximate Distributed Latent Dirichlet Allocation
ATM	Author-Topic Model
SVD	Singular Value Decomposition
VSM	Vector Space Model
NLP	Natural Language Processing
ML	Maximum Likelihood
MAP	Maximum a Posterior
MCMC	Markov Chain Monte Carlo
SVM	Support Vector Machine
GANIP	General Adaptive Neighborhood Image Processing
tf-idf	term-frequency and inverse document frequency
SP	Stochastic Pretopology
IC model	Independent Cascade model
LT model	Linear Threshold model
PCM	Pretopology Cascade Model
FRS	Finite Random Set
Textual-Homo-IC	Homophily Independent Cascade model based on Textual information
SNA	Social Network Analysis
ABM	Agent-Based Model
NIPS	Neural Information Processing Systems Conference
ARI	Adjusted Rand Index
AMI	Adjusted Mutual Information
MCPTM	Method of Clustering Documents using Pretopology and Topic Modeling
VBM	Vector-Based Measurement
PBM	Probabilistic-Based Measurement

List of Symbols

Notation	Meaning
K	number of topics
V	number of words in the vocabulary
M	number of documents
$\mathcal{D}$	corpus
$\phi_{j=1,\ldots,K}$	distribution of words in topic j
$\theta_{d=1,\ldots,M}$	distribution of topics in document d
$a(.)$	pseudo-closure function
$a_\Omega(.)$	stochastic pseudo-closure function
$i(.)$	iterior function
$cl(.)$	closure function
$F(A)$	closure of A
$O(A)$	opening of A
$\mathcal{F}_e$	family of elementary closed subsets
$\mathcal{F}_m$	family of minimal closed subsets
$\delta(A, B)$	pseudo-distance between A,B
$D_A(x)$	interior-pseudo distance of x in A
$MTP(d)$	major topic of document d
$d_H(P, Q)$	Hellinger distance between P, Q
R_{MTP}	relationship based on major topic
R_{d_H}	relationship based on Hellinger distance
k	number of clusters
$-A = A^c = X/A$	complement of set A in space X

Introduction

The research work carried out around this thesis is mainly focused on the study of Pretopology and Topic Modeling for Complex System analysis with application on Text mining and Complex Networks analysis. In our work, we have addressed the following research's questions?

Q1: What concept of proximity can be used in discrete structures ? (e.g. what are the advantages of *Pretopology* and its pseudo-closure's definition in algorithmics)

Q2: How to apply *Pretopology* for topic modeling in text mining?

Q3: How to apply *Pretopology* and topic modeling for complex networks analysis? then,

Q4: How to revisit some problems in data analysis such as structure analysis, classification, clustering, etc. with *Pretopology*?

The framework of the research

Developments on *Pretopology* [22, 23] find their foundations in problems relevant from the field of social sciences. Very often, in domains such as economy or sociology, the scientists try to build the models that can follow a transformation process step by step. To deal with such process, usual topology does not seem very adequate due to the fact that *closure function* is an **idempotent** one in topology. Therefore, to follow intermediate steps between a set A and $F(A)$ - closure of A, we have to relax this assumption of idempotence. This is exactly what *Pretopology* proposes.

The pretopology formalism comes from classical topology but has weaker axioms (the "closure" operator is not necessary *idempotent* and only the *extensivity* axiom is required). In consequence, pretopology can be generalized topology space for modeling the concept of proximity. Different to *closure opertor* with having only step from a set A to its closure, $F(A)$, *pseudo-closure* function, with its flexibility, can follow step by step the growth of a set, then it can be applied to solve some problems in complex systems such as diffusion of aerial pollution in a given geographic area, disease diffusion, information diffusion in complex networks, etc. For such processes, it is more interesting to see not only the final result but also how these processes take place step by step.

For this reason, pretopology is an efficient and pioneering theoretical framework to solve different problems: economical modeling [19, 66], game theory [16], shape recognition [69, 13, 31, 74, 73], image analysis [106, 14, 61, 27, 28], structural analysis [29, 30, 111], classification and clustering [154, 113, 114, 124, 71], supervised learning [117, 74, 73, 116], text mining [4, 53, 48, 115], generalization of graph theory [57, 58, 59, 60], complex networks modeling [119, 120, 125, 20, 21]; complex systems modeling (percolation processes [12, 9, 10], pollution phenomena [3, 107, 11], modeling

the impact of geographic proximity on scientific collaborations [111], analyzing the evolution of a communicable disease [20], analyzing communities of the web [122], modeling discrete signal machines [126] and modeling gesture trajectories [42]).

We recall here two applications that we will try to improve. The first is the work of Professor Michel Lamure's group [113, 114] when they proposed a method to find automatically a number k of clusters and k centroids for k-means clustering by results from structural analysis of minimal closed subsets algorithm [29, 30] and also proposed to use pseudo-closure distance constructed from the relationships family to examine the similarity measure for both numeric and categorical data. The second is the works of Michel Lamure's group and Marc Bui's group when they showed *Pretopology* as an extension of graph theory [57, 60] and applied in complex networks modeling [119, 120, 21], complex systems modeling such as percolation processes [12, 9, 10], pollution phenomena [3, 107, 11], etc.

Our concern at that time is how to improve these works? More detail, for the first work, the authors just illustrated the method with a toy example about the toxic diffusion between 16 geographical areas using only one relationship. Therefore, it did not show the advantage of pretopology that can work with a family of relationships for multi-criteria clustering. For the second work, they did not deal with either the random factor that often occurs in real life complex networks or the dynamic processes on complex networks such as information diffusion. These are exactly what things we propose to study in this thesis.

Pretopology and Topic Modeling for Text Mining

The first part of the Ph.D. work is in text mining field where we try to find the solutions the questions Q2 and Q4, given in the previous section. We first begin with *Vector Space Model* (VSM) in which each document is represented by a tf-idf (term-frequency (tf) and inverse document frequency (idf)) matrix. This matrix can be used as an input for some clustering algorithms such as k-means or the hierarchical clustering. However, the big issue of VSM is the high-dimensional matrix of document representation. To overcome this challenge, one of the possible solutions is to represent the text as a set of topics and use the topics as an input for a clustering algorithm. This is the idea of topic modeling. Topic Modeling is one of the most popular probabilistic clustering algorithms which has gained increasing attention recently. The main idea of topic modeling [26, 93, 82, 176] is to create a probabilistic generative model for the corpus of text documents. In topic models, documents are mixture of topics, where a topic is a probability distribution over words. The three main topic models are Latent Semantic Analysis (LSA) [109] which uses the Singular-value decomposition methods to decompose high-dimensional sparse matrix; Probabilistic Latent Semantic Analysis (pLSA) [93], a probabilistic model that treats the data as a set of observations coming from a generative model; *Latent Dirichlet Allocation* (LDA) [26] is a Bayesian extension of pLSA.

Therefore, in our work, we first reduce the dimensionality by decomposing the document matrix into latent components using the LDA [26] method. Each document is represented by a probability distribution of topics and each topic is characterized by a probability distribution over a finite vocabulary of words. We then use the probability distribution of topics as the input for k-means clustering. This approach called *LDA+k-means*. For document clustering using *LDA+k-means*, we face up to three challenges:

Q5: How to choose a "good" distance measure for *probability distribution* to get the most accurate clusters ?

Q6: How to choose the number of cluster for input of k-means ?

Q7: How to work with multi-criteria clustering ?

For the first challenge Q5, we investigate the efficiency of eight distance measures represented for eight distance families categorized by [52] when clustering with *LDA+k-mean*. These measures are based on two approaches: i) Vector-based measurement (VBM) with Euclidean distance, Sørensen distance, Tanimoto distance, Cosine distance and ii) Probabilistic-based measurement (PBM) with Bhattacharyya distance, Probabilistic Symmetric χ^2 divergence, Jensen-Shannon divergence, Taneja divergence. Experiments on two datasets (20Newsgroup, WebKB) with two evaluation criteria (Adjusted Rand Index, Adjusted mutual information) demonstrate the fact that the efficiency of PBM clustering is better than the VPM clustering including Euclidean distance.

For the two remain challenges Q6,Q7, we propose a **M**ethod of **C**lustering Documents using **P**retopology and **T**opic **M**odeling (MCPTM) that can cluster documents with multi-criteria. This result extends the method proposed in Le et al. [114] in two directions: firstly, we exploit this idea in document clustering and integrate structural information from LDA using the pretopological concepts of pseudo-closure and minimal closed subsets. Secondly, we show that Pretopology theory can apply to multi-criteria clustering by defining the pseudo-closure distance built from multi-relationships. In this work, we cluster documents by using two criteria: one based on the major topic of document (qualitative criterion) and the other based on Hellinger distance (quantitative criterion).

LDA is one of the most used topic models to discover complex semantic structure in the natural language processing (NLP) area. However, this method is unsupervised and therefore does not allow to include knowledge to guide the learning process. For this reason, we present a semi-supervised version of LDA (ss-LDA) based on the works of [161, 132]. The supervision of the process is within two levels: word level and document level. By connecting the ss-LDA and *Random Forest* classifier, we propose a new methodology for a multilayer soft classification for web pages. Our methodology allow us to obtain good accuracy results on a Data collection from *dmoz* [1].

In information retrieval systems we need both speed and accuracy. When we implemented LDA in python, it worked well. However, for massive corpora of text, the iterations of Gibbs sampling are extremely slow and can require days or even months of execution time. Clusters of computers can resolve this problem. From the work of Newman [148], we propose a **distributed version of Gibbs sampling built on Spark**. Spark [187] is a cluster computing and data-parallel processing platform for applications that focus on data-intensive computations. The main idea of the proposed algorithm is to make local copies of the parameters across the processors and synchronize the global counts matrices that represent the coefficients of LDA mixtures. We show empirically with a set of experimentations that our parallel implementation with Spark has the same predictive power as the sequential version and has a considerable speedup. We finally document an analysis of the scalability of our implementation and the super-linearity that we obtained.

Pretopology and Topic Modeling for Complex Network Analysis

Complex System is a relatively new and broadly interdisciplinary field that deals with systems composed of many interacting units often called “agents, such that the collective

[1] http://dmoz-odp.org/

behavior of its parts together is more than the "sum" of their individual behaviors [149]. The current theories of complex systems typically envisage a large collection of agents interacting in some specified way. To quantify the details of the system one must specify first its topology (who interact with whom) and then its dynamics (how the individual agents behave and how they interact).

The topology of complex systems is often specified in terms of complex networks that are usually modeled by graphs, composed by vertices and edges. Graph theory has been widely used in the conceptual framework of network models, such as random graphs, small world networks, scale-free networks [150, 67]. However, having more complicated non-regular topologies, complex systems need a more general framework for their representation [149].

For this reason, the second part of my Ph.D. work is applications of Pretopology and Topic Modeling in Complex Networks field where we tried to find the solutions for the questions Q3, given in the previous section.

We start with the definition of Graph in the *Claude Berge sense* [24], in which a graph, denoted by $G = (V, \Gamma)$, is a pair consisting of a set V of vertices and a **multivalued function** Γ **mapping** V **into** $\mathcal{P}(V)$. From this point, we show that Graph is a special case of pretopology space (V, a) in which pseudo-closure function $a(A) = \{x \in V | \Gamma(x) \cap A \neq \emptyset\}$ where $\Gamma(x) = \{y \in V | x\, R\, y\}$ built from a binary relation R on V. Therefore, we can generalize the definition of complex network by using *Pretopology network*, denoted by $G^{(Pretopo)} = (V, a)$, is a pair consisting of a set V of vertices and a pseudo-closure function $a(.)$ **mapping** $\mathcal{P}(V)$ **into** $\mathcal{P}(V)$.

Our research's questions are:

Q8: How to use *Pretopology* to deal with the **random factor** that often occurs in real life problems that take place over a complex networks structure ?

Q9: Can **dynamic processes** on complex networks such as information diffusion be modeled with the concepts of *Pretopology* ?

Q10: How to take benefits of Topic Modeling for Complex Networks analysis when nodes of these **networks contain textual information** ?

To overcome the first question Q8, we propose *Stochastic Pretopology* (SP), a result of the combination of *Pretopology theory* and *Random Sets* [144, 152], as a more general network modeling framework for complex system representation. In *SP*, given a subset A of the space, its pseudo-closure $a(A)$ is considered as a random set. So, we considered **the pseudo-closure not only as a set transform but also as a random correspondence**. After giving some examples for building stochastic pretopology in many situations, we show how this approach generalizes graph, random graph, multi-relational networks, etc.

For the second question Q9, we firstly represent *Independent Cascade model* [78] and *Linear Threshold model* [81] under stochastic pretopology language and then propose *Pretopology Cascade Model* as a general model for information diffusion process that can take place in more complex networks such as multi-relational networks or stochastic graphs. *Stochastic graphs* present in this work are defined by extending the definition of graph in the *Claude Berge sense* [24] $G = (V, \Gamma)$. In this approach, by considering Γ function as a finite random set defined from a degree distribution, we give a general graph-based network model in which *Erdős–Rényi model* and *scale-free networks* are special cases.

For the third question Q10, we propose an agent-based model for analyzing dynamic social network associated to textual information using author-topic model, namely *Textual-ABM. Author-topic model* [163] (ATM) is chosen to estimate topic's distribution

transformation of agents in the agent-based model since it models the content of documents and also interests of authors. *Textual-ABM* can be utilized to discover dynamic of a social network which includes not only network structure but also node's properties over time. Furthermore, we introduce a homophily independent cascade model based on textual information, namely *Textual-Homo-IC*. The infected probability associated with each edge is homophily or similarity which measured based on topic's distribution extracted from LDA or ATM. We have applied our methodology to two collected datasets from NIPS and social network platform *Twitter* and experimental results demonstrated that the effectiveness of *Textual-Homo-IC* on the static network outperforms *Random-IC*. In addition, experiments also illustrated the fluctuation of the active number on dynamic agent's network instead of obtaining and remaining a steady state in a static network.

The structure of the thesis

This thesis is divided into three parts: the first about the *state of the art of Pretopology* while the remaining two parts are related to applications stemming from the developed theory in document clustering and complex network analysis.

Specifically, the first part of this thesis - chapter 1 - reviews some basic concepts of pretopology theory, state of the art of pretopology as well as the two applications in data analysis and complex networks modeling that we will apply in our work which is presented in the two next parts. We also present the "algorithmic consequence" of mathematical works by proposing the practical way to build pretopological spaces for application and giving the state of the art of pretopology by briefly recalling most of the works related to Pretopology both in theory as well as in application.

The second part - composed of chapters 2 and 3 - develops a framework for document clustering by combining topic modeling and Pretopology. After presenting the *LDA* and the two extended versions: distributed version of LDA implemented in Spark and ss-LDA, an LDA version for learning process, in chapter 2, we proposed in chapter 3 the ways to solve three challenges when appling *LDA+k-means* for document clustering: choosing the "good" distance measure by comparing the effect of eight distance or similarity measures which divided into two appoaches: Probabilistic-based measurement (PBM) and Vector-based measurement (VBM); choosing the number of clusters by the result from structure analysis using minimal closed subsets and clustering with multi-criteria by using pseudo-closure distance defining from the pseudo-closure function to connect documents with multi-relations.

The final part - composed of chapters 4 and 5 - presents some applications of pretopology and topic modeling in complex network field. Chapter 4 develops a general framework for complex networks modeling by using *stochastic pretopology* and also proposes *Pretopology Cascade Model* as a general method for information diffusion. Chapter 5 proposes *Textual-ABM* as an agent-based model for analyzing dynamic social network associated to textual information using author-topic model and introduces *Textual-Homo-IC*, a homophily independent cascade model based on textual information, in which homophily or similarity measured based on topic's distribution extracted from LDA or ATM.

Some of the results of our work have been published

I. Related to Application of Pretopology and Topic Modeling for Text Mining :

1. Quang Vu Bui, Karim Sayadi, Soufian Ben Amor, Marc Bui. Integrating Latent Dirichlet Allocation and K-means for Documents Clustering: Effect of Probabilistic Based Distance

Measures, *Intelligent Information and Database Systems - 9th Asian Conference, ACIIDS 2017, Kanazawa, Japan, April 3-5, 2017, Proceedings, Part I. Lecture Notes in Computer Science 10191*, 2017, ISBN 978-3-319-54471-7, pages: 248-257.

2. Karim Sayadi, Quang Vu Bui, Marc Bui. Distributed Implementation of the Latent Dirichlet Allocation on Spark, *Proceedings of the Seventh Symposium on Information and Communication Technology, SoICT 2016, Ho Chi Minh City, Vietnam, December 8-9, 2016.* ACM 2016, ISBN 978-1-4503-4815-7, pages 92-98.

3. Quang Vu Bui, Karim Sayadi, Marc Bui. A multi-criteria document clustering method based on topic modeling and pseudoclosure function, *Informatical Journal*, Vol 40, No 2 (2016), pages:169-180.

4. Quang Vu Bui, Karim Sayadi, Marc Bui. A multi-criteria document clustering method based on topic modeling and pseudoclosure function, *Proceedings of the Sixth International Symposium on Information and Communication Technology, Hue City, Vietnam, December 3-4, 2015.* ACM 2015, ISBN 978-1-4503-3843-1, pages: 38-45.

5. Karim Sayadi, Quang Vu Bui, Marc Bui. Multilayer classification of web pages using Random Forest and semi-supervised Latent Dirichlet Allocation, *15th International Conference on Innovations for Community Services, I4CS 2015, Nuremberg, Germany, July 8-10, 2015.* IEEE 2015, ISBN 978-1-4673-7327-2, pages: 01-07.

II. Related to Application of Pretopology and Topic Modeling for Complex Networks Analysis :

6. Quang Vu Bui, Soufian Ben Amor, Marc Bui. Stochastic Pretopology as a Tool for Topological Analysis of Complex Systems, *Intelligent Information and Database Systems - 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018, Proceedings, Part II. Lecture Notes in Computer Science 10752*, 2018, ISBN 978-3-319-75419-2, pages: 102-111.

7. Kim Thoa Ho, Quang Vu Bui, Marc Bui. Dynamic Social Network Analysis Using Author-Topic Model, *18th International Conference on Innovations for Community Services, I4CS 2018, Zilina, Slovakia, June 18-20, 2018, Proceedings. Communications in Computer and Information Science 863*, Springer 2018, ISBN: 978-3-319-93407-5, pages: 47-62.

8. Thi Kim Thoa Ho, Quang Vu Bui, Marc Bui. Homophily Independent Cascade Diffusion Model Based On Textual Information, *10th International Conference on Computational Collective Intelligence (ICCCI 2018), Bristol, UK, September 05-07, 2018, Proceedings, Part II. Lecture Notes in Computer Science.*

9. Quang Vu Bui, Soufian Ben Amor, Marc Bui. Stochastic pretopology as a tool for complex networks analysis, *Journal of Information and Telecommunication, 2018, ISSN: 2475-1839 (Print) 2475-1847 (Online), DOI: 10.1080/24751839.2018.1542562.*

Part I

State of The Art

Chapter 1

Pretopology Theory

1.1 Introduction

Developments on *Pretopology* [22, 23] find their foundations in problems relevant to the field of social sciences. Very often, in domains such as economy or sociology, the scientists try to build the models that can follow a transformation process step by step. For example, it is the case when one studies diffusion processes such as diffusion of aerial pollution in a given geographic area, disease diffusion, information diffusion in complex networks, etc. It is more interesting to see not only the final result but also how these processes take place step by step.

To deal with such process, usual topology does not seem very adequate due to the fact that closure function is an idempotent one in topology. Therefore, to follow intermediate steps between a set A and $F(A)$, closure of A, we have to relax this assumption of idempotence. This is exactly what *pretopology* proposes.

The pretopology formalism comes from classical topology but has weaker axioms. The "closure" operator is not necessary idempotent and only the extensivity axiom is required. Marcel Brissaud, who must be considered as the "father of pretopology", gave the first definition of a pretopological space [36] in 1975 based on the previous works of Fréchet spaces [72], Kuratowski's closure axioms [105] and Čech closure operator [51]. Starting from this prerequisite, pretopology has been developed important theories during the 1970's and 1980's. We can recall the works of Gérard Duru [65, 66], Jean-Paul Auray [16, 19], Hubert Emptoz [69], Michel Lamure [106], Nicolas Nicoloyannis [154], Marcel Brissaud [37, 18, 38, 39]. In 1993, the first book about Pretopology [22] was written by Z.Belmandt[1]. These works were continued with many mathematicians and computer scientists up to now. Thus, at the moment, pretopology is a theory with a quite complete corpus of results on the ground of theory, and in the same time has played an important role in various applications: economical modeling [19, 66], games theory [16], shape recognition [69, 13, 31, 74, 73], image analysis [106, 14, 61, 27, 28], data analysis (structural analysis [29, 30, 111], classification and clustering [154, 113, 114, 124, 71], supervised learning [117, 74, 73, 116], text mining [4, 53, 48, 115, 46]), generalization of graph theory [57, 58, 59, 60], complex networks modeling [56, 119, 120, 125, 20, 21, 43], complex systems modeling (percolation processes [12, 9, 10], pollution phenomena [3, 107, 11], modeling the impact of geographic proximity on scientific collaborations [111], analyzing the evolution of a communicable disease [20], analyzing communities of

[1] collective name for a group of mathematicians and computer scientists working on Pretopology, namely Marcel **B**rissaud, Michel **L**amure, Jean-Jacques **M**ilan, Jean Paul **A**uray, Nicolas **N**icoloyannis, Gérard **D**uru, Michel **T**errenoire, Daniel **T**ounissoux, Djamel **Z**ighed. This group with new members, namely Stéphane Bonnevay, Le Thanh Van, Marc Bui, Soufian Benmor, Vincent Levorato worked together again and published the second book about pretopology in 2011 [23]

the web [122], modeling discrete signal machines [126] and modeling gesture trajectories [42]), smart grid simulation [157, 86], fuzzy pretopology [68, 76], software [123, 121].

In this chapter, after reviewing the state of the art of pretopology, we propose to the reader basic concepts related to pretopology as well as a landscape of the two applications in data analysis and complex networks modeling that we will apply in our work which is presented in the two next parts. We also present the "algorithmic consequence" of mathematical works by proposing the practical way to build pretopological spaces for applications.

Firstly, we present in the section 2 a general overview of the literature related to Pretopology in both theorical and application works. We then show, in section 3, how pretopology can be generalized topology space for modeling the concept of proximity. Similar to the topology space, we present some basic concepts in pretopology such as pseudo-closure operator, closed and open set, closure and opening, neighborhood, etc. We can also show how we build in pretopology space the concepts of continuity - which formalizes the transport between sets of a topological structure; compactness - which, combined with the notion of continuity, makes it possible to solve problems of existence of particular points such as a fixed point or an optimum; connectedness - which allows to model the concept of "homogeneity" of a part of a given set.

From the point of view the computer science, we need to find the way to build some algorithms that provide an easy-to-use framework for implementing basics concepts of pretopology and their associated operators in computers. We, therefore, present in section 4 the practical way to build pretopological spaces for applications. We propose in this section the way we build *pseudo-closure* functions from the *basis of neighborhoods* and illustrate this method in many situations such as Grid simulation ($\mathcal{Z}^2$), metric space, space equipped binary or valued relations or space equipped elements' neighborhood functions.

For applications of pretopology, we recall two applications that we will apply for our works in the two next parts. The first related to data analysis presented in section 5. In this section, we show how we can apply pretopology for structure analysis via the concept of *mimimal closed subset* and clustering via *the method of classification using pretopology with Reallocation* (MCPR). The second related to complex network modeling presented in section 6. We show pretopology as an extension of Graph theory which leads us to use pretopology for modeling the network with complex interactions. We also see how pretopology build some concepts similarly to graph such as path, chain, path-connectedness, partition, etc.

This chapter is organized as follows. Firstly, we review the state of the art of pretopology in section 2. Section 3 describes Pretopology as a mathematical tool for modeling the concept of proximity. We then present the practical way to build pretopological space for application in section 4. Section 5 presents pretopology as a tool for data analysis that we will apply in part 2. Another application of pretopology on network modeling which will be applied in part 3 is presented in section 6. Then it is followed by conclusions in section 7.

1.2 State of the art

1.2.1 Theoretical works

Pretopology can be considered as a mathematical tool for modeling the concept of proximity for complex systems. The mathematical theory aims to model the concept of proximity is topology [51] that allows for the definition of concepts such as continuity - which formalizes the transport between sets of a topological structure; compactness -

which, combined with the notion of continuity, makes it possible to solve problems of existence of particular points such as a fixed point or an optimum; connectedness - which allows modeling the concept of "homogeneity" of a part of a given set. Topology space can be defined via open sets or via neighborhoods. Another way to define a topological space is by using the Kuratowski closure axioms [105], which define the closed sets as the fixed points of an operator on the power set of space X. A Kuratowski Closure Operator [105] is an assignment $\mathrm{cl} : \mathcal{P}(X) \to \mathcal{P}(X)$ with the following properties:

(i) $\mathrm{cl}(\emptyset) = \emptyset$ (Preservation of Nullary Union)

(ii) $A \subseteq \mathrm{cl}(A)$ for every subset $A \subseteq X$ (Extensivity)

(iii) $\mathrm{cl}(A \cup B) = \mathrm{cl}(A) \cup \mathrm{cl}(B)$ for any subsets $A, B \subseteq X$ (Additive)

(iv) $\mathrm{cl}(\mathrm{cl}(A)) = \mathrm{cl}(A)$ for every subset $A \subseteq X$ (Idempotence)

In fact, in many practical situations such as for example that of social sciences, the axiomatic of topology was often incompatible with the requirements of the ground. Hence the idea of considering the construction of a theory with an axiomatic less restrictive than that of topology to obtain the "weakened structures": this is what pretopology proposes.

Brissaud [36], "father of pretopology", gave the first definition of a pretopological space in 1975 using pseudo-closure function satisfied only two axioms (i), (ii) from four axioms of Kuratowski's closure axioms. He told the story about the origins of pretopology and the questions that led to its development, Retour sur les origines de la prétopologie [40], at the second days of studies on pretopology, organized on 18-20 October 2007, in Sainte-Croix, Ain, France.

After the work of Brissaud, Gérard Duru, in his state doctoral thesis [66], presented new elements of Pretopology [65] and applied pretopology to study structures of complex systems in social sciences .

Jean-Paul Auray, in his state doctoral thesis "Contribution à l'étude des structures pauvres" [16], proposed some "weakened structures" definitions that we called V-type, V_D-type, V_S-type pretopological spaces which reminded again in [17]. He also worked with Brissaud and Duru to summarize their work about pretopology [18].

Brissaud introducted the connectedness, compactness for pretopological spaces [37] and applied them for analysis of the recovery of a repository [38] in economic. He also proposed pretopological analysis by using adherence and multicriteria acceptability [39] and introducted elements of generalized pretopology in [37, 41].

Dalud-Vincent, Brissaud, and Lamure [58] presented different concepts of closures in pretopological spaces which appear to us as quite important due to the fact that, on these concepts depend definitions of path and chain in pretopology. They also focused on modeling possibilities they offer in the field of social sciences, in comparison with tools from graph theory or topology.

In 1993, Z. Belmandt[2], a group of mathematicians and computer scientists working on pretopology, published the first book about Pretopology and its application [22] in French. Joined with some new members, this group worked together again in 2010 and published English version [23] after updated some advance works about Pretopology. In these two books, we can then show that, with a very limited axiomatic compared to that of the topology, and therefore more able to model field situations, we can generalize

[2]collective name for a group of mathematicians and computer scientists working on Pretopology, namely Marcel Brissaud, Michel Lamure, Jean-Jacques Milan, Jean-Paul Auray, Nicolas Nicoloyannis, Gérard Duru, Michel Terrenoire, Daniel Tounissoux, Djamel Zighed and Stéphane Bonnevay, Le Thanh Van, Marc Bui, Soufian Ben Amor, Vincent Levorato

in pretopology the basic concepts of the topology (adhesion and interior, closure and opening, ...), neighborhood, continuity, compactness, connectivity, space products.

By its convenience, pretopology is an efficient and pioneering theoretical framework to solve different problems that we present in following: economic modeling, image analysis, pattern recognition, data analysis (structure analysis, clustering, classification, text mining), generalization of graph theory and application for modelling complex networks, complex systems modelling (diffusion process, congestion and phase transition phenomena, propagation dynamics, ...), smart grid, etc.

1.2.2 Economic Modeling

The first application of pretopology is in the field of economics. Auray, Duru and Mougeot [19, 66] proposed a pretopological analysis of input-output model. An economic structure can be modeled by an Input-Output table. From this table, one can define different types of influence: direct influence (quantity), indirect influence (quantity), direct influence (price), global influence (price). Given a type of influence, one get a valued graph defined on the set of economic sectors. This kind of influence graph cannot be modeled by a topology. But, any graph of this kind is characteristic of a V_S pretopology. This V_S pretopological space can be built from dominance relationship R: jRk means: "sector j is influenced by sector k" (influence greater than a given threshold t). From this pretopology space, one can analyze the influence ratio by using interior ratio and pseudo-closure ratio defined from interior and pseudo-closure functions. Please refer [22], chapter 5 for more details.

The second application of pretopology in economic proposed by Auray et al. [16, 18]. Their work proposed a general modeling of the concept of the economic channel. For that, they based themselves on the concept of the minimal closed subset in V-type pretopological space. From the result of the minimal closed subset, one can analyze the structure of an economy.

1.2.3 Images analysis and pattern recognition

After the works of Brissaud [36], Duru [65, 66] and Auray [16], Pretopology theory became a mathematical modeling tool for the concepts of proximity suitable to discrete spaces. By this motivation, several researchers tried to develop some pretopological tools well adapted to image processing and pattern recognition. The choice of the pretopology is motivated by the fact that it has fewer axioms than the topology which facilitates its adaptation to discrete spaces and in particular image processing. For example, in a pretopological space, the adherence (pseudo-closure) function is not idempotent, what allows it successive applications on objects and then the mathematical modeling of segmentation by region growing. Pretopological closures are the best candidates to partition the image using connected and homogenous (by the means of the criterion associated with the pretopological structure) subsets. The pretopological closure is made by successive applications of the adherence function on initial germs chosen in the image. We recall in the following some groups that pioneered to apply pretopology for image processing and pattern recognition.

1.2.3.1 Pattern Recognition

The first related to the work of Hubert Emptoz and his group. H. Emptoz (1983) [69], in his doctoral thesis entitled "Modèles prétopologiques pour la reconnaissance des formes: application en Neurophysiologie", used pretopology to model the concept of proximity and thus gave meaning to the notion of neighborhood and then applied it in

pattern recognition for the formalization and implementation of some new methods of automatic classification. Selmaoui, Leschi and Emptoz [171, 172] proposed a new process for extreme lines detection based on the pretopological model. They set a new algorithm to analyze images of lines (images of the third type), its partly uses the principle of functioning of clustering algorithms proposed by H. Emptoz in his thesis [69], but has a different "philosophy" to interpret the results. Meziane, Iftene, and Selmaoui [140] proposed two algorithms based on the mathematical pretopology and the structuring functions for detecting crests lines in a grey level image at a very high definition. The first algorithm is based on a method of grouping by relaxing propagation on the definition of a pretopological structure on the set to be classified. The second algorithm consists of grouping by extraction of a new pretopology from the one defined initially. Dealing with the pretopological approach to the supervised pattern classification problem, Frélicot and Emptoz [74] proposed a learning process which results in a reduced number of neighborhoods to be learned and extended this approach with reject options in [73].

1.2.3.2 Image Analysis

The advance works in image analysis is related to the work of M. Lamure and his group. Some first result has been given on binary images [14] then M. Lamure [106], in his thesis, has proposed to increase these results to gray-level images. He has built a pretopological space according to the decomposition of a 256 gray-level image to 8 levels corresponding to the binary writing of 256. Lamure and Milan [108] used some pretopological operators (adherence, interior, border, derivative, coherency, etc) for the development of an interactive system "SAPIN" for the binary image structures detection. Dapoigny, Lamure et al [61] improved this system by developing a parallel implementation for pretopological transformations of binary images. For gray-level image analysis, Stéphane and Lamure [27, 28] proposed four structures corresponding to four new dilations (pseudo-closures) and four new erosions (interiors). These operators extend mathematical morphology operators in regard to the number of elements used to operate transformations. These new operators create new image transformations finer or deeper than mathematical morphology ones.

Other works are related to D.Mammass, F.Nouboud and their group. In the work of [135], they presented an approach based on a pretopological formalism that allows the mathematical modeling of image segmentation by region growing. In their approach, the pretopological adherence (pseudo-closure) function associated to the pretopological structure is defined by a criterion of homogeneity. They applied their approach to the extraction of handwritten information on check background with images of a scene and to edge detection. They also proposed a multi-criterion pretopological formalism that allows the mathematical modeling of image segmentation [136]. The multi-criteria pretopology is based on a multicriteria adherence that allows a process of aggregation to present the segmented image as a partition of all the multi-criterion pretopological closures (or classes of equivalence).

A recent research is related to the work of Johan Debayle and Jean-Charles Pinoli (2011) [62]. Their work introduced pretopological image filtering in the context of the General Adaptive Neighborhood Image Processing (GANIP) approach. Pretopological filters act on a gray level image while satisfying some topological properties. The GANIP approach enables to get an image representation and mathematical structure for adaptive image processing and analysis. Then, the combination of pretopology and GANIP leads to efficient image operators. They enable to process images while preserving region structures without damaging image transitions.

1.2.4 Data analysis

The first application of pretopology in data analysis proposed by Auray et al. [16, 18]. Based on the concept of elementary closed subsets - the closures of singletons - which allow us to observe the links between groups in the structure, they proposed the concept of the minimal closed subset in V-type pretopological space. From the result of minimal closed subset, one can analyse the structure of a given set. Bonneway, Largeon, Lamure and Nicoloyannis extended these works by proposing a pretopological approach for structural analysis: firstly for structuring data in non-metric spaces [29], then analyzing data based on minimal closed subsets [30] and summarizing their work in [111].

For classification and clustering, Nicolas Nicoloyannis [154] has developed a self-tuning algorithm and a software namely *Demon* for automatic classification based on pretopology in his thesis entitled "Structures prétopologiques et classification automatique: Le logiciel Demon". This work extended by Lamure, Bui and their group [113, 114, 124, 32, 124]. Le [113], in her thesis, proposed three methods of data clustering methods founded on the pretopology concepts. They function in two stages: the first one consists in structuring the population by the minimal closed subsets method and the second one consists in building a clustering of the population, starting from the previously obtained structuring. In the work of [124], they proposed a pretopological classification algorithm exploiting the notion of distance based on the complexity of Kolmogorov.

For text mining, the authors of this work [4] built a vector space with Latent Semantic Analysis (LSA) and used the pseudo-closure function from Pretopological theory to compare all the cosine values between the studied documents represented by vectors and the documents in the labeled categories. A document is added to a labeled category if it has a maximum cosine value.

1.2.5 Pretopology as an extension of Graph theory and its application in modeling complex networks

The advances of these works are related to the work of Michel Lamure and Marc Bui's group when they showed pretopology as an extension of graph theory and applied for complex network modeling. Firstly, by defining the pseudo-closure from a family of relationships, they showed how pretopology recovered the Graphs Theory [57, 60] and then they presented the various types of connectivities and their capacity to define (or not) a partition in components of a pretopological space [59]. They also showed the links between Matro¨ides and pretopology on one hand and between Hypergraphs and pretopology on the other hand [60].

Complex network modeling is generally based on graph theory, which allows for studying of dynamics and emerging phenomena. However, in terms of neighborhood, the graphs are not necessarily adapted to represent complex interactions, and the neighborhood of a group of vertices can be inferred from the neighborhoods of each vertex composing that group. As a generalization of the graph theory, pretopology can apply for modeling complex networks [119, 20] by modeling group in social networks, generalizing some measures used in social network analysis (degree, betweenness, and closeness) by considering a group as a whole entity [120], detecting of communities in directed networks based on strongly p-connected components [125].

1.2.6 Complex systems modeling

Different to *closure opertor* with having only step from a set A to its closure, $F(A)$, *pseudo-closure* function, with its flexibility, can follow step by step the growth of a set, then it can be applied to solve some problems in complex systems. Bui, Amor et al. [12, 9, 10] proposed a generalization of percolation processes in $\mathcal{Z}^2$ using the pretopology theory. They formalized the notion of a neighborhood by extending it to the concept of proximity, expressing different types of connections that may exist between the elements of a population. A modeling and simulation of forest fire using this approach showed the efficiency of this formalism to provide a realistic and powerful modeling of complex systems. Lamure and Bui's group [12, 9, 10] proposed a model of aerial pollution of an urban area, the city of Ouagadougou. They developed a mathematical model which based both on the concept of pretopological structure and the concept of random sets of the spatial area from the pollution point of view. We can recall some other results in complex systems modeling such as modeling the impact of geographic proximity on scientific collaborations [111], analyzing the evolution of a communicable disease [20], analyzing communities of the web [122], modeling discrete signal machines [126] and modeling gesture trajectories [42].

1.3 Pretopology as a mathematical tool for modeling the concept of proximity

1.3.1 Topological space

In topology and related branches of mathematics, a topological space may be defined as a set of points, along with a set of neighborhoods for each point, satisfying a set of axioms relating points and neighborhoods. The definition of a topological space relies only upon set theory and is the most general notion of a mathematical space that allows for the definition of concepts such as continuity, connectedness, and convergence. The utility of the notion of a topology is shown by the fact that there are several equivalent definitions of this structure. Thus one chooses the axiomatization suited for the application. We recall here four definitions of topological space based on terms of open sets, closed sets, neighborhoods and Kuratowski closure axioms. The most commonly used is that in terms of open sets, but perhaps more intuitive is that in terms of neighborhoods and so this is given first.

1.3.1.1 Definition via neighborhoods

This axiomatization is due to Felix Hausdorff. Let X be a set; the elements of X are usually called points, though they can be any mathematical object. We allow X to be empty. Let $\mathcal{N} : X \to \mathcal{P}(\mathcal{P}(X))$ be a function assigning to each $x \in X$ a non-empty collection $\mathcal{N}(x)$ of subsets of X. The elements of $\mathcal{N}(x)$ will be called neighborhoods of x with respect to $\mathcal{N}$ (or, simply, neighborhoods of x). The function $\mathcal{N}$ is called a *neighbourhood topology* if the axioms below [51] are satisfied; and then X with $\mathcal{N}$ is called a topological space.

1. If N is a neighborhood of x (i.e., $N \in \mathcal{N}(x)$) then $x \in N$.
2. If M is a neighborhood of x and $M \subseteq N \subseteq X$, then N is a neighbourhood of x.
3. The intersection of two neighborhoods of x is a neighborhood of x.
4. If N is a neighborhood of x, then N contains a neighborhood M of x such that N is a neighborhood of each point of M.

The first three axioms for neighborhoods have a clear meaning. The fourth axiom has a very important use in the structure of the theory, that of linking together the neighborhoods of different points of X.

A standard example of such a system of neighborhoods is for the real line $\mathbb{R}$, where a subset N of $\mathbb{R}$ is defined to be a neighborhood of a real number x if it includes an open interval containing x.

Given such a structure, a subset U of X is defined to be open if U is a neighborhood of all points in U. The open sets then satisfy the axioms given below. Conversely, when given the open sets of a topological space, the neighborhoods satisfying the above axioms can be recovered by defining N to be a neighborhood of x if N includes an open set U such that $x \in U$

1.3.1.2 Definition via open sets

A topological space is an ordered pair (X, τ), where X is a set and τ is a collection of subsets of X, satisfying the following axioms [51]:

1. The empty set and X itself belong to τ.
2. Any (finite or infinite) union of members of τ still belongs to τ.
3. The intersection of any finite number of members of τ still belongs to τ.

The elements of τ are called open sets and the collection τ is called a topology on X.

1.3.1.3 Definition via closed sets

Using de Morgan's laws, the above axioms defining open sets become axioms defining closed sets:

1. The empty set and X are closed.
2. The intersection of any collection of closed sets is also closed.
3. The union of any finite number of closed sets is also closed.

Using these axioms, another way to define a topological space is as a set X together with a collection τ of closed subsets of X. Thus the sets in the topology τ are the closed sets, and their complements in X are the open sets.

1.3.1.4 Definition via Kuratowski closure axioms

A topological space can be defined in terms of the Kuratowski closure axioms [105]. A topological space is an ordered pair (X, cl), where X is a set and $\mathrm{cl} : \mathcal{P}(X) \to \mathcal{P}(X)$ is a Kuratowski Closure Operator, satisfying the following axioms:

(K1) $\mathrm{cl}(\emptyset) = \emptyset$ (Preservation of Nullary Union)

(K2) $A \subseteq \mathrm{cl}(A)$ for every subset $A \subseteq X$ (Extensivity)

(K3) $\mathrm{cl}(A \cup B) = \mathrm{cl}(A) \cup \mathrm{cl}(B)$ for any subsets $A, B \subseteq X$ (Additive)

(K4) $\mathrm{cl}(\mathrm{cl}(A)) = \mathrm{cl}(A)$ for every subset $A \subseteq X$ (Idempotence)

A subset $C \subseteq X$ is called closed if and only if $\mathrm{cl}(C) = C$.
By using closed set defined from closure opoerator, we can show a closure operator naturally induces a topology as follows:
Empty Set and Entire Space are closed:
By extensitivity, $X \subseteq \mathrm{cl}(X)$ and since closure maps the power set of X into itself (that is, the image of any subset is a subset of X), $\mathrm{cl}(X) \subseteq X$ we have $X = \mathrm{cl}(X)$. Thus X is closed. The preservation of nullary unions states that $\mathrm{cl}(\varnothing) = \varnothing$. Thus $\varnothing$ is closed.
Arbitrary intersections of closed sets are closed:
Let $\mathcal{I}$ be an arbitrary set of indices and C_i closed for every $i \in \mathcal{I}$.
By extensitivity, $\bigcap_{i\in\mathcal{I}} C_i \subseteq \mathrm{cl}(\bigcap_{i\in\mathcal{I}} C_i)$.
Also, by preservation of inclusions,

$$\bigcap_{i\in\mathcal{I}} C_i \subseteq C_i \forall i \in \mathcal{I} \Rightarrow \mathrm{cl}(\bigcap_{i\in\mathcal{I}} C_i) \subseteq \mathrm{cl}(C_i) = C_i \forall i \in \mathcal{I} \Rightarrow \mathrm{cl}(\bigcap_{i\in\mathcal{I}} C_i) \subseteq \bigcap_{i\in\mathcal{I}} C_i.$$

Therefore, $\bigcap_{i\in\mathcal{I}} C_i = \mathrm{cl}(\bigcap_{i\in\mathcal{I}} C_i)$. Thus $\bigcap_{i\in\mathcal{I}} C_i$ is closed.
Finite unions of closed sets are closed:
Let $\mathcal{I}$ be a finite set of indices and let C_i be closed for every $i \in \mathcal{I}$. From the preservation of binary unions and using induction we have $\bigcup_{i\in\mathcal{I}} C_i = \mathrm{cl}(\bigcup_{i\in\mathcal{I}} C_i)$. Thus $\bigcup_{i\in\mathcal{I}} C_i$ is closed.

1.3.2 From Topology to Pretopology

In fact, in many practical situations such as for example that of social sciences, the axiomatic of topology was often incompatible with the requirements of the ground. Hence the idea of considering the construction of a theory with an axiomatic less restrictive than that of topology to obtain the "weakened structures": this is what pretopology proposes.

Based on the previous works of Fréchet spaces [72], Kuratowski's closure axioms [105] and Čech closure operator [51], Brissaud [36], "father of pretopology", gave the first definition of a pretopological space in 1975 using pseudo-closure function satisfied only two axioms (K1), (K2) from four axioms of Kuratowski's closure axioms.

Definition 1.1. *A pretopological space is an ordered pair* (X, a)*, where* X *is a set and* $\mathrm{a} : \mathcal{P}(X) \to \mathcal{P}(X)$ *is a* ***pseudo-closure*** *operator, satisfying the two following axioms:*

(P1): $a(\emptyset) = \emptyset$; *(Preservation of Nullary Union)*

(P2): $A \subset a(A) \quad \forall A, A \subset X$ *(Extensivity)*

We can note that pseudo-closure fulfills two of properties of a topological closure. In order to simplify the notation in the following we write $-A$ instead of $X \backslash A$ for the complement of A in X. Similarity to topological space, the c-duality of the pseudo-closure function is the *interior function* $\mathrm{i} : \mathcal{P}(X) \to \mathcal{P}(X)$ defined by:

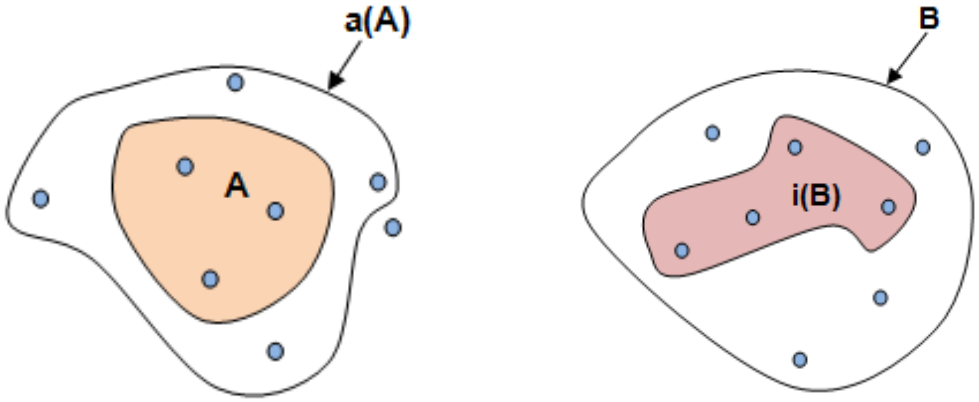
$$\mathrm{i}(A) = -\,\mathrm{a}(-A) \tag{1.1}$$

Given the interior function i, we obviously recover the pseudo-closure as $a(A) = -(i(-A))$. So, we can denote pretopology space as (X, a, i) or simply (X, a). We can also characterize a pretopological space via the proposition in the following:

Proposition 1.1. (X, a, i) *is a pretopological space if and only if*

i. the functions i *and* a *are c-dualites*

ii. $i(X) = X$



(a) Pseudo-closure of A (b) Interrior of B

Figure 1.1: Pseudo-closure and interior function

iii. $i(A) \subset A \quad \forall A, A \subset X$

If (X, a, i) is a pretopological space, we say that space X is equipped a pretopological structure (or a pretopology) defined by pseudo-closure function a(.) (see Fig.1.1a) or interior function i(.) (see Fig.1.1b) .
It is important to note that, by defining $a(.)$ we do not suppose that it is an idempotent transform. Then, conversely as it happens in topology, we can compute:
$a(A), a(a(A)), a(a(a(A))), \ldots, a^k(A)$. So, *pseudo-closure* allows, for each of its applications, to add elements to a set departure according to defined characteristics. The starting set gets bigger but never reduces. Subset $a(A)$ is called the pseudo-closure of A. As $a(a(A))$ is not necessarily equal to $a(A)$, a sequential appliance of pseudo-closure on A can be used to model expansions: $A \subset a(A) \subset a(a(A)) = a^2(A) \subset \ldots \subset a^k(A)$ (see Fig.1.2).

1.3.3 Closed and open sets, Closure and Openning

Definition 1.2. *Let (X, a, i) a pretopological space, $\forall A, A \subset X$. A is a closed subset if and only if $a(A) = A$.*

Definition 1.3. *Let (X, a, i) a pretopological space, $\forall A, A \subset X$. A is an open subset of X if and only if $A = i(A)$.*

If a subset A of a pretopological space (X, a, i) is both a closed and an open subset, we call it is an *oc* of the pretopological space. In the same way as in topology, we obviously obtain the following result.

Proposition 1.2. *Given a pretopological space (X, a, i) where $a(.)$ and $i(.)$ are defined by c-duality, then, for any $A, A \subset X$, we have:*
A is a closed subset of $X \Leftrightarrow A^c$ is an open subset of X.

Then, it is possible to generalize the two concepts of closure and opening of any subset of a pretopological space.

Definition 1.4. *Given a pretopological space (X, a, i), call the closure of A, when it exists, the smallest closed subset of X which contains A. The closure of A is denoted by $F(A)$.*

Definition 1.5. *Given a pretopological space (X, a, i) call opening of any subset A of E, when it exists, the biggest open subset of (X, a, i) which is included in A. The opening of A is denoted by $O(A)$.*

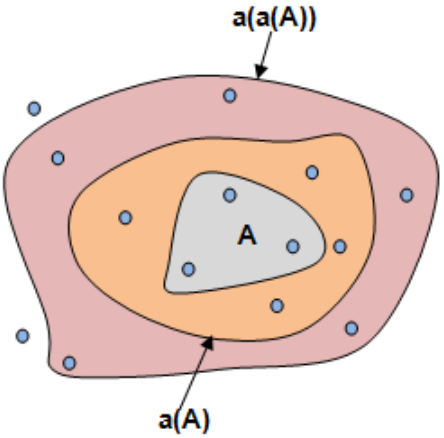
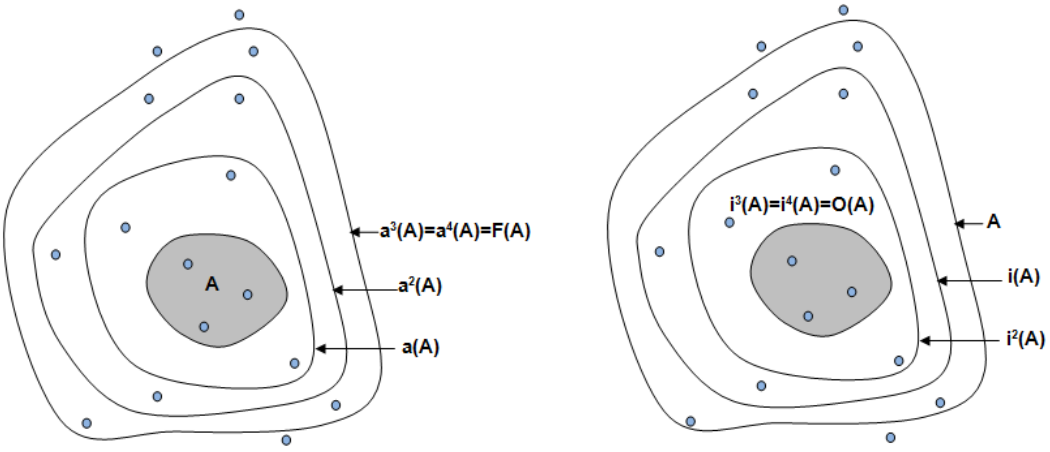


Figure 1.2: Successive computations of $a(A)$



(a) Closure of A (b) Opening of A

Figure 1.3: Iterated application of the pseudo-closure and interior map leading to the closure and opening.

Here, we can note a fundamental difference between topology and pretopology: as in topology, the family of closed subsets characterizes the topological structure, we, however, do not get the same result in pretopology. Two different pseudo-closure $a(.)$ and $a^{'}(.)$ can give the same family of closed subsets on a set X. Please refer [22, 23] for an example illustrates that result.

1.3.4 Neighborhood in pretopological space

A neighborhood of a point is a set of points containing that point where one can move some amount away from that point without leaving the set. In topology and related areas of mathematics, a neighborhood (or neighborhood) is one of the basic concepts in a topological space. It is closely related to the concepts of open set and interior. If X is a topological space and x is a point in X, a neighborhood of x is a subset V of X that includes an open set U containing x. This is also equivalent to $x \in X$ being in the interior of V. Similarly, we can define the concept of neighborhood in pretopological space.

1.3.4.1 Neighborhood of a point

Definition 1.6. *Let (X, a, i) be pretopological space and x is a point in X, a neighborhood (pretopological neighborhood) of x is a subset N of X such that $x \in i(N)$.*

Then, we can define neighborhood function and convergent function such that:

Definition 1.7. *Let (X, a, i) be pretopological space. Then the neighborhood function $\mathcal{N} : X \to \mathcal{P}(\mathcal{P}(X))$ and the convergent function $\mathcal{N}^* : X \to \mathcal{P}(\mathcal{P}(X))$ assign to each $x \in X$ the collections*

$$\mathcal{N}(x) = \{N \in \mathcal{P}(X) | x \in i(N)\} \tag{1.2}$$

$$\mathcal{N}^*(x) = \{Q \in \mathcal{P}(X) | x \in a(Q)\} \tag{1.3}$$

of its neighborhoods and convergents, respectively.

It is not hard to see that neighborhoods and convergents are equivalent:

Theorem 1.3 ([175], Thm. 1)**.** $Q \in \mathcal{N}^*(x) \Leftrightarrow Q^c \notin \mathcal{N}(x)$

The next result shows that pseudo-closure function and neighborhood function are equivalent. Hence given one of pseudo-closure function a(.), interior function i(.), neighborhood function $\mathcal{N}$ or covergent function $\mathcal{N}^*$, the other three functions are unambiguously defined.

Theorem 1.4 ([175], Thm.2)**.** *Let $\mathcal{N}$, $\mathcal{N}^*$ be the neighborhood function defined in equation (1.2) and the convergent function defined in equation (1.3). Then*

$$x \in a(A) \Leftrightarrow A^c \notin \mathcal{N}(x) \quad \text{and} \quad x \in i(A) \Leftrightarrow A^c \notin \mathcal{N}^*(x) \tag{1.4}$$

1.3.4.2 Neighborhood of Sets

The notation of a neighborhood for an individual point can be extended naturally to sets.

Definition 1.8. *Let (X, a, i) be pretopological space and $A \subset X$. A set V is a neighborhood of A, in symbols $V \in \mathcal{N}(A)$, if $V \in \mathcal{N}(x)$ for all $x \in A$.*

1.3.5 Pretopological Spaces

As we show in the previous, given a pretopological space (X, a, i), we can define the neighborhood function $\mathcal{N}$ defined in equation (1.2). Conversely, given a neighborhood function $\mathcal{N}$ on space X, we can also define pretopological structure with pseudo-closure function $a(.)$. Thus, as in topology, we get a concept of neighborhoods. However, up to now, pretopological neighborhoods do not verify the same properties than topological neighborhoods. For example, it is very easy to see that if U is a neighborhood of a given x in X and if U is included in a subset V of X, that does not mean that V is a neighborhood of x. So, we are led to define different types of pretopological spaces which are less general than the basic ones but for which, good properties are fulfilled by neighborhoods. In this section, we propose the different types of pretopological spaces which have been defined:

1. $\mathcal{V}$-type space or Isotone spaces
2. $\mathcal{V}_D$-type space or Distributive spaces
3. $\mathcal{V}_S$-type space
4. and (at last) Topological spaces

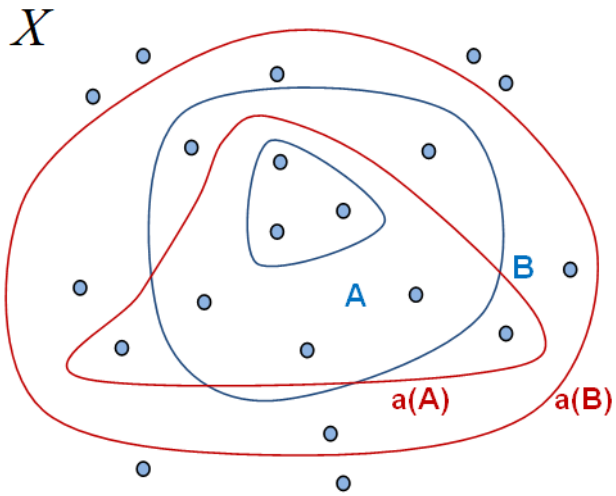


Figure 1.4: Example of $\mathcal{V}$-type pretopological space

1.3.5.1 $\mathcal{V}$-type space

Definition 1.9. *A Pretopology space (X, a, i) is called $\mathcal{V}$-type space if and only if*

$$(P3) \quad (A \subseteq B) \Rightarrow (a(A) \subseteq a(B)) \quad \forall A, B \in \mathcal{P}(X) \quad (Isotonic) \tag{1.5}$$

A pseudo-closure function satisfying (P3) is called *isotonic*. Then $\mathcal{V}$-type pretopological space is also called *Isotonic Pretopological Space*. Almost all approaches to extend the framework of topology at least assume that the pseudo-closure functions are isotonic, or, equivalently, that the neighborhoods of a point form a prefilter. The importance of isotonic property is emphasized by a large number of equivalent conditions.

Lemma 1.5 ([175], Lemma 2)**.** *The following conditions are equivalent for arbitrary pseudo-closure functions* a : $\mathcal{P}(X) \to \mathcal{P}(X)$.

(P3) $A \subseteq B \Rightarrow a(A) \subseteq a(B) \quad \forall A, B \in \mathcal{P}(X)$.

(P3^I) $a(A) \cup a(A) \subseteq a(A \cup B) \quad \forall A, B \in \mathcal{P}(X)$.

(P3^{II}) $a(A \cap B) \subseteq a(A) \cap a(A) \quad \forall A, B \in \mathcal{P}(X)$.

It is easy to derive equivalent conditions for the associated interior function by repeated applications of $i(A) = -a(-A)$ and $a(A) = -i(-A)$. One obtains

$(P3^{III})$ $A \subseteq B \Rightarrow i(A) \subseteq i(B) \quad \forall A, B \in \mathcal{P}(X)$.

$(P3^{IV})$ $i(A) \cup i(A) \subseteq i(A \cup B) \quad \forall A, B \in \mathcal{P}(X)$.

$(P3^{V})$ $i(A \cap B) \subseteq i(A) \cap i(A) \quad \forall A, B \in \mathcal{P}(X)$.

Most interestingly, we can use the concept of the neighborhood to characterize a $\mathcal{V}$-type pretopological space (X, a, i) as we show in the following. For that, we need to introduce the following definition.

Definition 1.10. *Given a family $\mathcal{B}$ of subsets of a set X, we say that $\mathcal{B}$ is a prefilter of subsets of X if and only if:*

i. $\emptyset \notin \mathcal{B}$

ii. $\forall A \in \mathcal{B}, (A \subseteq B \Rightarrow B \in \mathcal{B})$

Proposition 1.6 ([23], Prop.1.2.1)**.** *Given a $\mathcal{V}$-type space (X, a, i), for any $x \in X$, the family $\mathcal{N}(x) = \{N \in \mathcal{P}(X) | x \in i(N)\}$ of neighborhoods of x is a prefilter of subsets of X.*

Proposition 1.7 ([23], Prop.1.2.2)**.** *Let $\mathcal{N}(x)$ be a prefilter of subsets of X for any x in X.*
Let $i(.)$ and $a(.)$ be the functions from $\mathcal{P}(X)$ into itself defined as:

(i) $\forall A \in \mathcal{P}(X), \quad i(A) = \{x \in X | \exists N \in \mathcal{N}(x), V \subset A\}$

(ii) $\forall A \in \mathcal{P}(X), \quad a(A) = \{x \in X | \forall N \in \mathcal{N}(x), V \cap A \neq \emptyset\}$

Then (X, a, i) is a $\mathcal{V}$-type space. We say it is generated by the family $\{\mathcal{N}(x), x \in X\}$.

At this point, given a $\mathcal{V}$-type space (X, a, i), we are able to determine the family of neighborhoods $\mathcal{N}(x)$ for any $x \in X$. And if for any $x \in X$, we have a prefilter of subsets of X, we are able to determine a pseudo-closure $a(.)$ (and so an interior function $i(.)$) such as we get a $\mathcal{V}$-type space (X, a, i). The problem then is to answer the following question: "given the initial $\mathcal{V}$-type space (X, a, i), from the family $\mathcal{N}(x)$; $\forall x \in X$, we define a new pseudo-closure function and a new interior function, are they the same as the functions of the initial space?". The following proposition gives this answer.

Proposition 1.8 ([23], Prop.1.2.3)**.** *The $\mathcal{V}$-type pretopological space (X, a, i) is generated by an unique family of prefilters $\mathcal{N}(x)$ and conversely any family of prefilters $\mathcal{N}(x)$ generates an unique pretopological structure on X.*

We have defined open and closed subsets in a previous paragraph, which had led to closure and opening of a subset A. In the most general case, opening and closure do not necessarily exist. In the case of $\mathcal{V}$-type pretopological spaces, we get the following results which lead us to a specific result of opening and closure.

Proposition 1.9 ([23], Prop.1.2.5)**.** *Given a $\mathcal{V}$-type pretopological space (X, a, i)*

(i) $A \subset X$ is open if and only if A is a neighborhood for each of its elements

(ii) Let $x \in X$ and $V \subset X$, if there exists an open subset A such as $\{x\} \subset A \subset V$, then V is a neighborhood of x. The converse is generally not true.

(iii) Any union of open sets is an open set.

(iv) Any intersect of closed sets is a closed set.

This last result leads to the following which establishes the existence of opening and closure of any subset in a $\mathcal{V}$-type space.

Proposition 1.10 ([23], Prop.1.2.6)**.** *In any pretopological space of type V, given a subset A of X, the closure and opening of A always exists.*

So, in the $\mathcal{V}$-type space, given a finite set X, the closure $F(A)$ always exits and can be calculated by using the following property that is used for computing the distance between elements.
$\exists k < |X|, F(A) = a^k(A) = a(a^{k-1}(A))$.

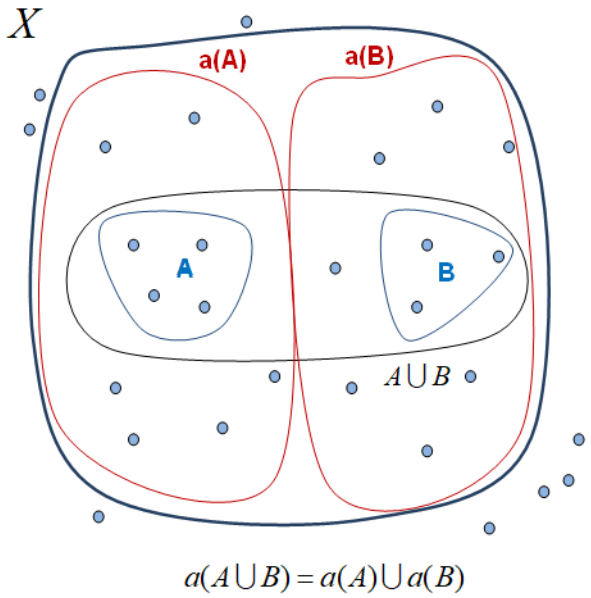


Figure 1.5: Example of $\mathcal{V}_D$-type pretopological space

1.3.5.2 $\mathcal{V}_D$-type space

In this section, we present a new type of pretopological space determined by adding a new property to the pseudo-closure function and by c-duality to the interior function.

Definition 1.11. *A Pretopology space (X, a, i) is called $\mathcal{V}_D$-type space if and only if*

$$(P4) \quad a(A \cup B) = a(A) \cup a(B) \quad \forall A \subset X, \quad \forall B \subset X \quad (Additive) \tag{1.6}$$

A pseudo-closure function satisfying (P4) is called *additive*. Then $\mathcal{V}_D$-type pretopological space is also called *Distributive Pretopological Space*.
It is easy to derive equivalent conditions for the associated interior function by repeated applications of $i(A) = -a(-A)$ and $a(A) = -i(-A)$. One obtains
(P4') $\quad i(A \cap B) = i(A) \cap i(B) \quad \forall A \subset X, \forall B \subset X$.

Proposition 1.11 ([23], Prop.1.2.7)**.** *Any $\mathcal{V}_D$-type space is a $\mathcal{V}$-type space.*

Thus a $\mathcal{V}_D$-type space is a particular case of $\mathcal{V}$-type space. Then, we can wonder if the family of neighborhoods fulfills more properties in the case of $\mathcal{V}_D$-type space as in the case of $\mathcal{V}$-type space. For that, we need to introduce the following definition.

Definition 1.12. *Let $\mathcal{F}$ a family of subsets of a set X, we say that $\mathcal{F}$ is a filter if and only if $\mathcal{F}$ is a prefilter which is stable for intersection: $\forall F \in \mathcal{F}, \forall G \in \mathcal{F}, F \cap G \in \mathcal{F}$*

Then, we have :

Proposition 1.12 ([23], Prop.1.2.8)**.** *A pretopological space (X, a, i) is of $\mathcal{V}_D$-type if and only if it is a $\mathcal{V}$-type for which, for any $x \in X$, the family $\mathcal{N}(x) = \{N \in \mathcal{P}(X) | x \in i(N)\}$ of neighborhoods of x is a filter of subsets of X.*

Proposition 1.13 ([23], Prop.1.2.9)**.** *Given a $\mathcal{V}_D$-type pretopological space (X, a, i)*

(i) $\emptyset$ and X are open sets

(ii) Any union of open sets is an open set.

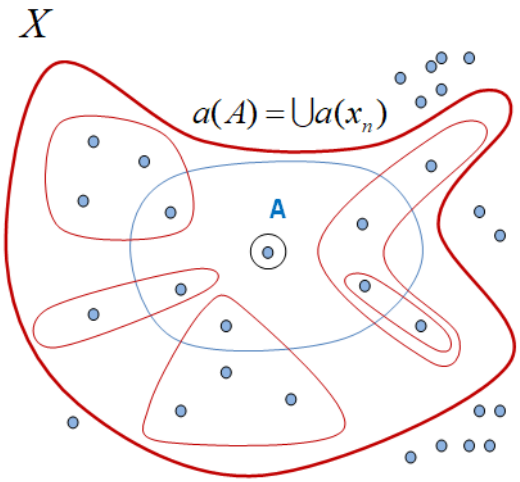


Figure 1.6: Example of $\mathcal{V}_S$-type pretopological space

(iii) Any finite intersect of open sets is an open set.

(iv) $\forall A, B \in \mathcal{P}(X), O(A \cap B) = O(A) \cap O(B)$

Proposition 1.14 ([23], Prop.1.2.10)**.** *Given a $\mathcal{V}_D$-type pretopological space (X, a, i)*

(i) $\emptyset$ and X are closed sets

(ii) Any finite union of closed sets is a closed set.

(iii) Any intersect of closed sets is an closed set.

(iv) $\forall A, B \in \mathcal{P}(X), F(A \cup B) = F(A) \cup F(B)$

1.3.5.3 $\mathcal{V}_S$-type space

Definition 1.13. *A Pretopology space (X, a, i) is called $\mathcal{V}_S$-type space if and only if*

$$(P5) \quad a(A) = \bigcup_{x \in A} a(\{x\}) \quad \forall A \subset E \tag{1.7}$$

Proposition 1.15 ([23], Prop 1.2.11)**.** *Any $\mathcal{V}_S$-type space is a $\mathcal{V}_D$-type space.*

The following proposition gives a specific property fulfilled by the family of neighborhoods.

Proposition 1.16 ([23], Prop 1.2.12)**.** *A necessary and sufficient condition for (X, a, i) being a $\mathcal{V}_S$-type space is, for any element x of X, the intersect of all neighborhoods of x is also a neighborhood of x.*

1.3.5.4 Topological space

The last level in pretopological spaces is the level of topological spaces.

Definition 1.14. *Given a pretopological space (X, a, i), we say (X, a, i) is a topological space if and only if (X, a, i) is of $\mathcal{V}_D$-type and*

$$(P6) \quad a(a(A)) = a(A) \quad \forall A \subset X \quad (Idempotence) \tag{1.8}$$

This definition is equivalent to the definition of Kuratowski [105]. It shows that any topological space is a particular pretopological space.

1.3.6 Comparison of Pretopological Spaces

Let (X, a, i) and (X, a', i') two pretopological spaces.

Definition 1.15. *We say that the pretopological space (X, a', i') is finer than the pretopological space (X, a, i) if and only if for any subset A of X we have $A \subset a'(A) \subset a(A)$, or equivalently $i(A) \subset i'(A) \subset A$.*

When the pretopological space (X, a', i') is finer than the pretopological space (X, a, i), we also say that the pretopological structure defined by (a', i') is finer than that defined by (a, i). And conversely, we also say that (a, i) is less fine or coarser than (a', i').

In $\mathcal{V}$-type pretopological space, comparison of pretopological structures can be represented by the prefilter of neighborhoods.

Definition 1.16. *Let $\mathcal{F}$ and $\mathcal{G}$ be two prefilters defined on X. $\mathcal{G}$ is said finer than $\mathcal{F}$ if and only if $\mathcal{F} \subset \mathcal{G}$, i.e $\forall F \in \mathcal{F}, F \in \mathcal{G}$*

Proposition 1.17 ([23], Prop.1.3.1)**.** *Let (X, a, i) and (X, a', i') be two $\mathcal{V}$-type pretopological spaces. a' (or i') is finer than a (or i) if and only if that for any x in X, the prefilter of neighborhoods of x for the pretopology defined by a' (or i') is finer than the prefilter of neighborhoods of x defined by a (or i).*

1.3.7 Continuity in pretopological spaces

1.3.7.1 Continuity in Topological space

In mathematics, a continuous function is a function for which sufficiently small changes in the input result in arbitrarily small changes in the output. Continuity of functions is one of the core concepts of topology, which is treated in full generality below. A function $f : X \to Y$ between topological spaces is called continuous if for every $x \in X$ and every neighbourhood N of $f(x)$ there is a neighbourhood M of x such that $f(M) \subseteq N$. This relates easily to the usual definition in analysis. Equivalently, f is continuous if the inverse image of every open set is open.

By using Kuratowski closure axioms [105], the topology can also be determined by a closure operator (denoted cl) which assigns to any subset $A \subseteq X$ its closure. In this term, a function $f\colon (X, \mathrm{cl}) \to (X', \mathrm{cl}')$ between topological spaces is continuous if and only if for all subsets A of X $f(\mathrm{cl}(A)) \subseteq \mathrm{cl}'(f(A))$.
That is to say, given any element x of X that is in the closure of any subset A, $f(x)$ belongs to the closure of $f(A)$. This is equivalent to the requirement that for all subsets A' of X': $f^{-1}(\mathrm{cl}'(A')) \supseteq \mathrm{cl}(f^{-1}(A'))$.

1.3.7.2 Continuity in pretopological spaces

Definition 1.17 ([175], Def.4)**.** *Given two pretopological spaces (X, a_X, i_X), (Y, a_Y, i_Y). A function $f : X \to Y$ is*

closure preserving *if for all $A \in \mathcal{P}(X)$ holds $f(a_X(A)) \subseteq a_Y(f(A))$;*

continuous *if for all $B \in \mathcal{P}(Y)$ holds $a_X(f^{-1}(B)) \subseteq f^{-1}(a_Y(B))$*

Theorem 1.18 ([175], Thm.4)**.** *Let (X, a_X, i_X), (Y, a_Y, i_Y) be two pretopological spaces and let $f : X \to Y$. Then the following conditions (for continuity) are equivalent:*

(i) $a_X(f^{-1}(B)) \subseteq f^{-1}(a_Y(B)) \quad \forall B \in \mathcal{P}(Y)$.

(ii) $f^{-1}(i_Y(B)) \subseteq i_X(f^{-1}(B)) \quad \forall B \in \mathcal{P}(Y)$.

(iii) $B \in \mathcal{N}(f(x)) \Rightarrow f^{-1}(B) \in \mathcal{N}(x) \quad \forall B \in \mathcal{P}(Y), \forall x \in X$

(iv) $f^{-1}(B) \in \mathcal{N}^*(x) \Rightarrow B \in \mathcal{N}^*(f(x)) \quad \forall B \in \mathcal{P}(Y), \forall x \in X.$

Conditions (iii) and (iv) are equivalent for each individual $x \in X$ as well.

Definition 1.18. *Let (X, a_X, i_X), (Y, a_Y, i_Y) be two pretopological spaces. Then $f : X \to Y$ is continuous in $x \in X$ if $\forall B \in \mathcal{P}(Y), B \in \mathcal{N}(f(x)) \Rightarrow f^{-1}(B) \in \mathcal{N}(x)$ (or, $f^{-1}(B) \in \mathcal{N}^*(x) \Rightarrow B \in \mathcal{N}^*(f(x))$).*

An immediate consequence of theorem 1.18 is the following familiar relationship between local and global continuity:

Corollary 1.19. *Let (X, a_X, i_X), (Y, a_Y, i_Y) be two pretopological spaces. Then $f : X \to Y$ is continuous if and only if it is continuous in x for all $x \in X$.*

Remark: In contrast to Topological space, in general pretopological spaces, a function f is *closure preserving* not mean it is *continuous* and vice versa. For $\mathcal{V}$-type space, we have a interesting result as following:

Theorem 1.20 ([175], Thm.10)**.** *Let (X, a_X, i_X), (Y, a_Y, i_Y) be two $\mathcal{V}$-type pretopological spaces. Then the following properties are equivalent:*

(i) $f : X \to Y$ is continuous

(ii) $f : X \to Y$ is closure preserving.

(iii) $f(A) \subseteq B \Rightarrow f(a_X(A)) \subseteq a_Y(B) \quad \forall A \in \mathcal{P}(X), \forall B \in \mathcal{P}(Y).$

1.3.8 Connectedness

In any pretopological space of type $\mathcal{V}$, given a subset A of X, the closure and opening of A always exists. So, in $\mathcal{V}$-type space, we can consider the concept of connectedness.

Definition 1.19. *Let (X, a) be a $\mathcal{V}$-type pretopological space.*

(i) (X, a) is strongly connected (SC) if and only if $\forall C \in X, C \neq \emptyset, F(C) = X$.

(ii) (X, a) is one-sidely connected (OC) if and only if $\forall C \in X, C \neq \emptyset, F(C) = X$ or $\forall B \in X, B \neq \emptyset$ if $B \subset X - F(C)$ then $C \in F(B)$.

(iii) (X, a) is hyperconnected (HC) if and only if $\forall C \in X, C \neq \emptyset, F(C) = X$ or $\exists B \in X, B \neq \emptyset$ if $B \subset X - F(C)$ then $C \in F(B)$.

(iv) (X, a) is apo-connected (AC) if and only if $\forall C \in X, C \neq \emptyset, F(C) = X$ or $\forall B \in X, B \neq \emptyset$ if $B \subset X - F(C)$ then $F(C) \cap F(B) \neq \emptyset$.

(v) (X, a) is connected (C) if and only if $\forall C \in X, C \neq \emptyset, F(C) = X$ or $F(X - F(C)) \cap F(C) \neq \emptyset$.

We shall note χ-connected for talking about these five connectedness.

Proposition 1.21 ([22], Prop.12.1.2)**.** *Let (X, a) be a $\mathcal{V}$-type pretopological space.*

(i) If (X, a) is strongly connected then (X, a) is one-sidedly connected.

(ii) If (X, a) is one-sidedly connected then (X, a) is hyper connected and (X, a) is apo-connected.

(iii) If (X, a) is hyperconnected then (X, a) is connected.

(iv) If (X, a) is apo-connected then (X, a) is connected.

Remark: These definitions come down to those given classically in graphs theory if we choose the pretopology of the descendants.

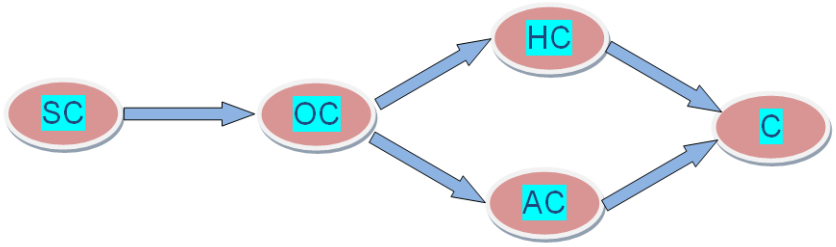


Figure 1.7: Relation between five types of pretopological connectivity

1.3.8.1 Connectedness of a set

Let (X, a) be a $\mathcal{V}$-type pretopological space and let $A \subset X$. We note F_A the closing obtained by restriction of closing F on A. F_A is such as $\forall C \subset A, F_A(C) = F(C) \cap A$.

Definition 1.20. *Let (X, a) be a $\mathcal{V}$-type pretopological space. Let $A \subset X$ with A non empty. A is a χ-connected subset of (X, a) if and only if A endowed with F_A is χ-connected. By misuse of language, we shall say that A is χ-connected to mean that A is a χ-connected subset of (X, a).*

The definitions of the various connectivities concerning the subsets are thus the following ones:

Proposition 1.22 ([22])**.**

(i) A is strongly connected if and only if $\forall C \in A, C \neq \emptyset, F(C) \cap A = A$.

(ii) A is one-sidely connected if and only if $\forall C \in A, C \neq \emptyset, F(C) \cap A = A$ or $\forall B \in A, B \neq \emptyset$ if $B \subset A - (F(C) \cap A)$ then $C \in F(B) \cap A$.

(iii) A is hyper connected if and only if $\forall C \in E, C \neq \emptyset, F(C) \cap A = A$ or $\exists B \in A, B \neq \emptyset$ if $B \subset A - (F(C) \cap A)$ then $C \in F(B) \cap A$.

(iv) A is apo-connected if and only if $\forall C \in A, C \neq \emptyset, F(C) \cap A = A$ or $\forall B \in A, B \neq \emptyset$ if $B \subset A - (F(C) \cap A)$ then $F(C) \cap F(B) \cap A \neq \emptyset$.

(v) A is connected if and only if $\forall C \in A, C \neq \emptyset, F(C) \cap A = A$ or $F(A - (F(C) \cap A)) \cap F(C) \cap A \neq \emptyset$.

1.4 Pratical Ways to build pretopological space

As its flexibility, pretopology can be considered as a good tool for modeling concept of proximity in discrete spaces. In general, ones can build any arbitrary pseudo-closure function satisfies only the two first properties (P1), (P2). However, as we showed in the previous, the general pretopological space lacks some essential properties that ones need for building applications such as the existence of closure or opening, preserve some interesting properties related to neighborhood, etc. It is the reason why we need to build some pretopological spaces that is less general than arbitrary pretopological space but larger enough for building applications. $\mathcal{V}$-type pretopological space (or isotonic pretopological space) is one of the most interesting cases. Two advances of $\mathcal{V}$-type space are that: firstly, it preserves almost essential properties for extended topological spaces and secondly, it can be characterized by the family of neighborhoods of elements since it is a prefilter. This gives us a practical way to build such spaces. In this section, we will present some essential theoretical concepts and propose different ways to build $\mathcal{V}$-type pretopological spaces.

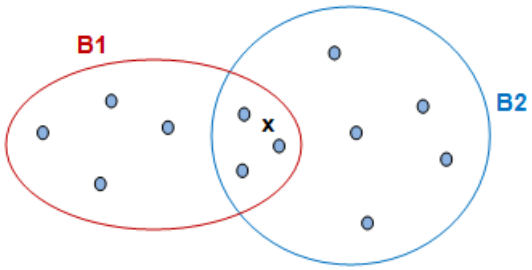


Figure 1.8: Two basic of neighborhoods of elements x

1.4.1 Pretopological space built from a basis of neighborhoods

Let (X, a, i) be pretopological space, $\forall x \in X$, the family of neighborhoods of x is defined in the equation (1.2):

$$\mathcal{N}(x) = \{N \in \mathcal{P}(X) | x \in i(N)\}$$

From the proposition 1.8, we can see that in $\mathcal{V}$-type spaces, the family of neighborhoods of elements is a prefilter which characterizes the space. This gives a practical way to build spaces. However, it can be difficult to specify this family of neighborhoods. A quite practical concept is the following.

Definition 1.21. *Given a $\mathcal{V}$-type pretopological space (X, a, i) defined by the family $\mathcal{N}(x), \forall x \in X$, the family $\mathcal{B}(x)$ is called a basis of neighborhoods of x if and only if*

$$\forall x \in X, \forall N \in \mathcal{N}(x), \exists B \in \mathcal{B}(x) \text{ such as } B \subset N \tag{1.9}$$

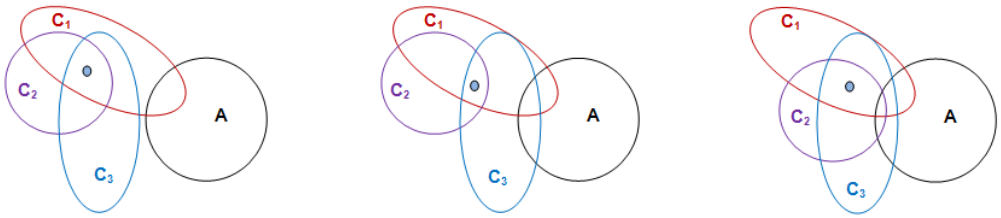
The question then is to know how $\mathcal{B}(x)$ works in the definition of the pseudo-closure $a(,)$ and the interior $i(,)$.

Proposition 1.23 ([23], Prop.1.2.4)**.** *Given a $\mathcal{V}$-type pretopological space (X, a, i) defined by the family $\mathcal{N}(x), \forall x \in X$ and the basis of neighborhoods $\mathcal{B}(x)$, then:*

$$(i) \quad \forall A \in \mathcal{P}(X), \quad i(A) = \{x \in X | \exists B \in \mathcal{B}(x), B \subset A\} \tag{1.10}$$

$$(ii) \quad \forall A \in \mathcal{P}(X), \quad a(A) = \{x \in X | \forall B \in \mathcal{B}(x), B \cap A \neq \emptyset\} \tag{1.11}$$

Remark: Every times the prefilter of neighbourhoods $\mathcal{N}(x)$ is generated by a basis $\mathcal{B}(x)$ **reduced to one element**, the space (X, a, i) is of $\mathcal{V}_S$-type pretopological space. Clearly, the proposition 1.23 gives us a good way to build pseudo-closure functions.



(a) $x \notin a(A)$ (b) $x \notin a(A)$ (c) $x \in a(A)$

Figure 1.9: Pseudo-closure built from the base of neighborhoods $\mathcal{B}(x) = \{C_1(x), C_2(x), C_3(x)\}$

Ones just define a basis of neighborhoods of each element x in the space X: $\mathcal{B}(x) = \{B_1(x), B_2(x), \ldots, B_n(x)\}$ (see fig.1.8 for an example) and then define pseudo-closure function from equation (1.11) (see fig.1.9 for an example) and iterior function from equation (1.10).
We present in the following the way we build pseudo-closure function in many situations.

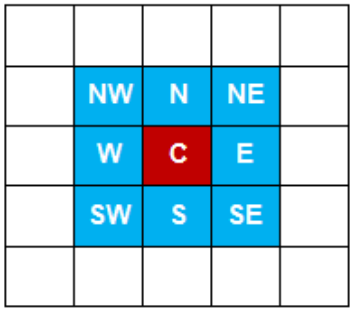


Figure 1.10: The Moore neighborhood is composed of nine cells: a central cell and the eight cells which surround it.

1.4.2 Pretopology in $\mathcal{Z}^2$

In this subsection, we present some ways to build pretopological spaces in $\mathcal{Z}^2$ based on basis of neighborhoods. For any $(x, y) \in \mathcal{Z}^2$, basis of neighborhoods $\mathcal{B}(x, y)$ of (x, y) is composed of (x, y) and its neighborhood points following some rules. We will start with the two most used kinds of the neighborhood: *Moore neighborhood*, *Von Neumann neighborhood* and will propose some more complex neighborhoods that are useful for building some more complex pretopological spaces.

1.4.2.1 Based on Moore neighborhood

In cellular automata, the *Moore neighborhood* (Fig.1.10) is defined on a two-dimensional square lattice and is composed of a central cell and the eight cells which surround it. The neighborhood is named after *Edward F. Moore*, a pioneer of cellular automata theory. It is one of the two most commonly used neighborhood types, the other one being the von Neumann neighborhood. The well known Conway's Game of Life, for example, uses the Moore neighborhood. It is similar to the notion of 8-connected pixels in computer graphics.
For all (x, y) in $\mathcal{Z}^2$, the *Moore neighborhood* of (x, y), denoted $B_8(x, y)$, is defined such as:
$B_8(x, y) = \{(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1), (x, y), (x + 1, y - 1), (x + 1, y + 1), (x - 1, y - 1), (x - 1, y + 1)\}\}$
Clearly, the families $(B_8(x, y))_{(x,y)\in\mathcal{Z}^2}$ is a basis of neighborhoods. We then define pseudo-closure function and interior function based on *Moore neighborhood* $B_8(x, y)$ such as:

$$\forall A \subset \mathcal{Z}^2, \quad a_8(A) = \{(x, y) \in \mathcal{Z}^2 | B_8(x, y) \cap A \neq \emptyset\}$$

$$\forall A \subset \mathcal{Z}^2, \quad i_8(A) = \{(x, y) \in \mathcal{Z}^2 | B_8(x, y) \subset A\}$$

So, (Z^2, a_8, i_8) is a $\mathcal{V}_S$-type pretopological space. Let we recall the *Chebyshev distance* in $\mathcal{Z}^2$

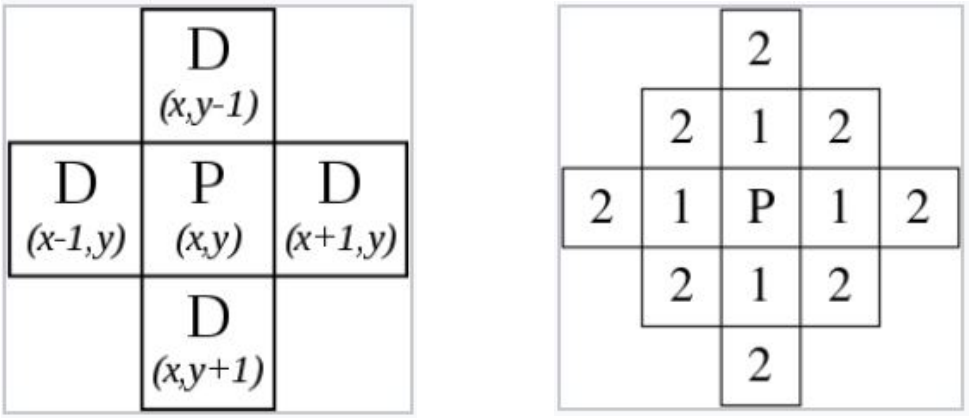
$$D_{\text{Cheby}}((x_1, y_1), (x_2, y_2)) = \max\left(|x_2 - x_1|, |y_2 - y_1|\right).$$

The Moore neighborhood of a cell is the cell itself and the cells at a Chebyshev distance of 1. So, we can define an extended Moore neighborhood with range r, $B_{8r}(x, y)$, such as:

$$B_{8r}(x, y) = \{(x_1, y_1) \in \mathcal{Z}^2 | d_{Cheby}((x, y), (x_1, y_1)) \leq r\}; r \geq 1$$

The number of cells in an extended Moore neighborhood, given its range r is: $(2r + 1)^2$. When $r = 1$, we have the Moore neighborhood. This neighborhood can be used to define the notion of 4-connected pixels in computer graphics.

1.4.2.2 Based on Von Neumann neighborhood



(a) Von Neumann neighborhood (b) Manhattan distance $r = 2$

Figure 1.11: Von Neumann neighborhood of range r (a) r=1; (b) r=2

In cellular automata, the *von Neumann neighborhood* (Fig.1.11a) is classically defined on a two-dimensional square lattice and is composed of a central cell and its four adjacent cells. The neighborhood is named after *John von Neumann*, who used it to define the von Neumann cellular automaton and the von Neumann universal constructor within it. For all (x, y) in $\mathcal{Z}^2$, the *von Neumann neighborhood* of (x, y), denoted $B_4(x, y)$, is defined such as:

$$B_4(x, y) = \{(x + 1, y), (x - 1, y), (x, y), (x, y + 1), (x, y - 1)\}$$

Similarity to Moore neighborhood, (Z^2, a_4, i_4) generated from $B_4(x, y)$ is also a $\mathcal{V}_S$-type pretopological space.
Let we recall the *Manhattan distance* in $\mathcal{Z}^2$

$$D_{\text{Manha}}((x_1, y_1), (x_2, y_2)) = |x_2 - x_1| + |y_2 - y_1| .$$

The Von Neumann neighbourhood of a cell is the cell itself and the cells at a Manhattan distance of 1. So, we can defined an extended Moore neighbourhood with range r (see Fig.1.11b for an example with $r = 2$), $B_{4r}(x, y)$, such as:

$$B_{4r}(x, y) = \{(x_1, y_1) \in \mathcal{Z}^2 | d_{Manha}((x, y), (x_1, y_1)) \leq r\}; r \geq 1$$

The number of cells in a 2-dimensional von Neumann neighborhood of range r can be expressed as $1 + 2r(r + 1)$.
An advantage of Pretopology is that we can build some complex interactions by using intersect operator for building pseudo-closure function. For example, we can build basis of neighborhoods $B_{4VR}(x, y)$ (see Fig.1.12) using both *Von Neumann* $B_4(x, y)$ and *Rotated Von Neumann* $B_{4R}(x, y)$ neighborhood such as:

$$B_{4VR}(x, y) = [B_4(x, y), B_{4R}(x, y)]$$

with $B_{4R}(x, y) = \{(x + 1, y + 1), (x + 1, y - 1), (x, y), (x - 1, y + 1), (x - 1, y - 1)\}$

Then we build pseudo-closure function $a_4(.)$ and interior $i_4(.)$ for any $A \subset \mathcal{Z}^2$ such as:

$$a_{4VR}(A) = \{(x, y) \in \mathcal{Z}^2 | B_4(x, y) \cap A \neq \emptyset \text{ and } B_{4R}(x, y) \cap A \neq \emptyset\}$$

$$i_{4VR}(A) = \{(x, y) \in \mathcal{Z}^2 | B_4(x, y) \subset A \text{ or } B_{4R}(x, y) \subset A\}$$

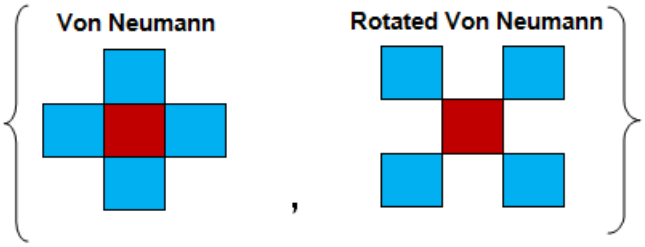


Figure 1.12: Basis of neighborhoods with Von Neumann neighbor and Rotated Von Neumann neighbor

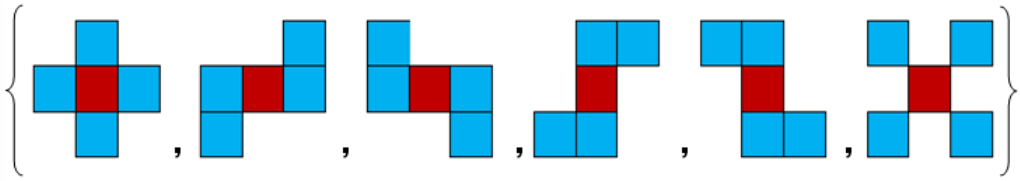
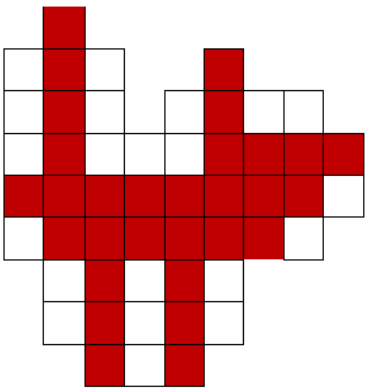
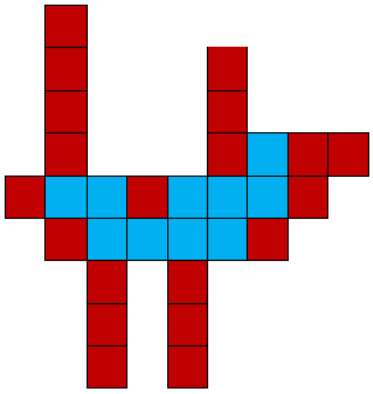


Figure 1.13: Basis of neighborhoods with 6 elements $B_{4V6eles}$

So, (Z^2, a_{4VR}, i_{4VR}) is a $\mathcal{V}$-type pretopological space.
We can also build other $\mathcal{V}$-type pretopological spaces built from a basis of neighborhoods with more than two elements. For example, figure 1.13 shows an example of basis $B_{4V6eles}$ with 6 elements and figure 1.14a shows the pseudo-closure built from this base of neighborhoods while figure 1.14b shows the interior.



(a) A in red and a(A) box



(b) A in red and i(A) in blue

Figure 1.14: Pseudo-closure and interior build from the basis $B_{4V6eles}$.

1.4.2.3 Based on 2-connected neighborhood

For 2-connected neighborhood, we can build four type of basis of neighborhoods (Fig.1.15) such that:

$$B_{2h}(x,y) = \{(x+1,y),(x-1,y),(x,y)\}$$
$$B_{2v}(x,y) = \{(x,y+1),(x,y-1),(x,y)\}$$
$$B_{2d1}(x,y) = \{(x+1,y+1),(x-1,y-1),(x,y)\}$$
$$B_{2d2}(x,y) = \{(x-1,y+1),(x+1,y-1),(x,y)\}$$

The pretopology space built from one of these neighborhoods is also of $\mathcal{V}_S$-type space. Similar to 4-connected neighborhood case, we can also build a basis of neighborhoods by using intersect operator. Firstly, we build a basis of neighborhoods with intersecting

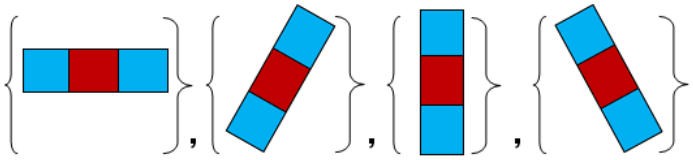
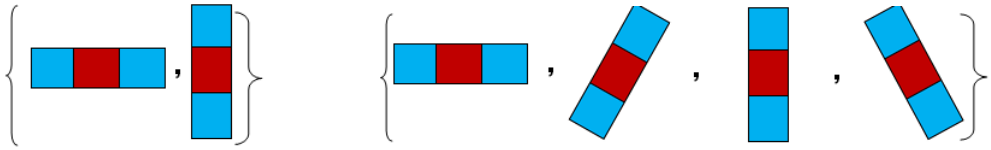


Figure 1.15: Four types of basic of neighborhoods with 2 neighbors



(a) Two directions

(b) Four directions

Figure 1.16: Basic of neighborhoods using intersect of two directions (a) or four directions (b)

of the horizontal direction and vertical direction (Fig.1.16a):

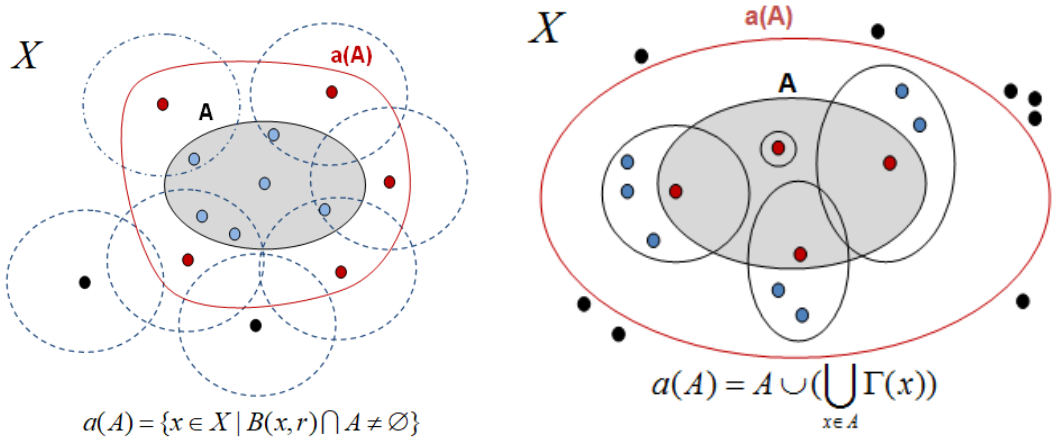
$$B_{2hv}(x,y) = [B_{2h}(x,y), B_{2h}(x,y)].$$

Secondly, we build basic of neighborhoods with intersecting of four directions: horizontal, vertical and two diagonal directions (Fig.1.16b) such as:

$$B_{2hvd}(x,y) = [B_{2h}(x,y), B_{2v}(x,y), B_{2d1}(x,y), B_{2d2}(x,y)]$$

Pseudo-closure a_{2hv} (respectively a_{2hvd}) built from $B_{2hv}(x,y)$ (respectively B_{2hvd}) generates a $\mathcal{V}$-type pretopological space.
Remark: These pseudo-closure functions can be applied for gray-level image analysis. Please refer [22, 106] for more details.

1.4.3 Pretopology in metric space



(a) a(A) in a metric space

(b) a(A) in a space equipped with a neighbor function

Figure 1.17: Pseudo-closure function in metric space and a space equipped with a neighbor function.

Let us consider space X is endowed with a metric defined by a distance d. Let r be a positive real. For each element x of X, $B(x,r)$ is a ball with the center x and a radius r defined by

$$B(x,r) = \{y \in V | d(x,y) \leq r\}$$

Clearly, $\mathcal{B}(x) = (B(x,r)_{x \in E})$ is basis of neighborhoods of X. We then can build a *pseudo-closure* function **a(.)** on X with $B(x,r)$ such as:

$$\forall A \in \mathcal{P}(X), \quad a(A) = \{x \in X | B(x,r) \cap A \neq \emptyset\} \tag{1.12}$$

Pretopological space built in the previous is of $\mathcal{V}_D$-type pretopological space ($\mathcal{V}_S$-type if X is finite). The *pseudo-closure* $a(A)$ is a set of all elements $x \in X$ such that x is within a distance of at most radius r from at least one element of A (see Fig.1.17a for an example).

1.4.4 Pretopology in a space equipped with a neighborhood function

Let us consider a multivalued function $\Gamma : X \to \mathcal{P}(X)$ as a neighborhood function. $\Gamma(x)$ is a set of neighborhoods of element x. We define a *pseudo-closure* **a(.)** (see Fig.1.17b for an example) as follows:

$$\forall A \in \mathcal{P}(X), \quad a(A) = A \cup (\bigcup_{x \in A} \Gamma(x)) \tag{1.13}$$

Clearly, pretopology space built from pseudo-closure $a(.)$ function defined in previous is of $\mathcal{V}_S$ type. Graph is defined by *Claude Berge sense* [24] is a special case of this kind of pretopology space.

1.4.5 Pretopology and binary relationships

1.4.5.1 Pretopoly built from one relationship

Suppose we have a binary relationship R on a finite set X. Let us consider neighborhood of x, $R(x)$, defined by:

$$R(x) = \{y \in X | x \, R \, y\} \tag{1.14}$$

Then, the pseudo-closure $a_d(.)$ is defined by:

$$a_d(A) = \{x \in X | R(x) \cap A \neq \emptyset\} \cup A \quad \forall A \subset X \tag{1.15}$$

and interior functon $i(.)$ is defined by:

$$i_d(A) = \{x \in X | R(x) \subseteq A\} \quad \forall A \subset X \tag{1.16}$$

The pretopological space build from pseudo-closure function $a_d(.)$ is called pseudo-closure of descendants. Similarity, we can build the pseudo-closure of ascendants, noted $a_a(.)$, is defined by:

$$a_a(A) = \{x \in X | R^{-1}(x) \cap A \neq \emptyset\} \cup A \quad \forall A \subset X$$

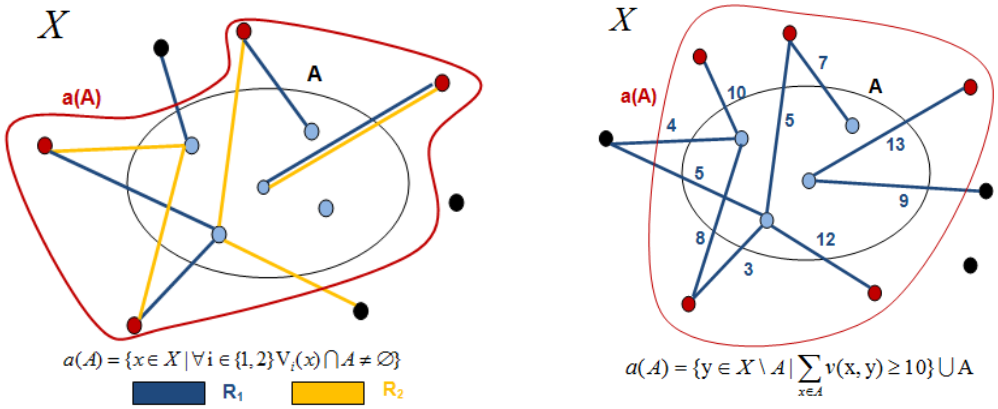
with $R^{-1}(x) = \{y \in X | y \, R \, x\}$
The pseudo-closure of ascendant-descendants, noted $a_{ad}(.)$, is defined by:

$$\forall A \subset X, a_{ad}(A) = \{x \in X | R^{-1}(x) \cap A \neq \emptyset \text{ and } R(x) \cap A \neq \emptyset\} \cup A$$

Proposition 1.24. *The pretopological space built from pseudo-closure of descendants $a_d(.)$ and pretopological space built from pseudo-closure of ascendants $a_a(.)$ are V_S-type ones. The pretopological space built from pseudo-closure of ascendant-descendants $a_{ad}(.)$ is V-type one.*

Remark: Pretopology of descendants enables us to model all applications relevant to the graph theory. In particular, the concepts of path, chain, connectivity are extended in pretopology while remaining compatible with the graph theory, see [22].

1.4.5.2 Pretopology built from a family of relationships



(a) a(A) in a binary space

(b) a(A) in a valued space

Figure 1.18: Pseudo-closure function in binary and valued spaces

Suppose we have a family $(R_i)_{i=1,\ldots,n}$ of binary relationships on a finite set X. Let us consider $\forall i = 1, 2, \ldots, n, \forall x \in X, V_i(x)$ defined by:

$$R_i(x) = \{y \in X | x \, R_i \, y\} \tag{1.17}$$

Then, the pseudo-closure $a_s(.)$ is defined by:

$$a_s(A) = \{x \in X | \forall i = 1, 2, \ldots, n, R_i(x) \cap A \neq \emptyset\} \quad \forall A \subset X \tag{1.18}$$

Pretopology defined on X by $a_s(.)$ using the intersection operator is called the strong pretopology. Figure 1.18a gives an example for strong pretopology built from two relationships.

Proposition 1.25. *$a_s(.)$ determines on X a pretopological structure and the spaces (X, a_s) is of V-type pretopological space.*

Similarly, we can define weak Pretopology from $a_w(.)$ by using the union operator:

$$a_w(A) = \{x \in X | \exists i = 1, 2, \ldots, n, R_i(x) \cap A \neq \emptyset\} \quad \forall A \subset X \tag{1.19}$$

Proposition 1.26. *$a_w(.)$ determines on X a pretopological structure and the space (X, a_s) is of $\mathcal{V}_D$-type.*

1.4.6 Pretopology and valued relationships

In order to model certain problems such as model in weighted graph, we often need the space X are bound by a valued relation. For instance, we can define an real value ν on relations as a function from $X \times X \to \mathbb{R}$ as: $(x, y) \to \nu(x, y)$. The *pseudo-closure* **a(.)** can build such as:

$$\forall A \in \mathcal{P}(X), \quad a(A) = \{y \in X - A | \sum_{x \in A} \nu(x, y) \geq s\} \cup A; s \in \mathbb{R} \tag{1.20}$$

The pseudo-closure $a(.)$ is composed of A and of all elements y where the sum of valued edges between some elements of A and y is greater than the threshold s. Pretopological space built from this kind of pseudo-closure is $\mathcal{V}$-type space. Fig.1.18b gives an illustration of this space with $s = 10$. This kind of modeling can be used in social networks

where weighted relations are necessary and illustrates the interest of the pretopology modeling. Indeed, this example shows that group behavior is different than the "sum" of individuals composing it. In Fig.1.18b, the person at the top is absorbed because he knows two persons in group A with the weights are 7 and 5 respectively, so he can be considered as a friend of group A. If we take each individual of A saying this external individual, the person, at the top, is a friend of A; if the value of a link is superior to $s = 12$, he will not be taken into account.

1.5 Pretopology as a tool for Data Analysis

As mentioned in the previous section, Pretopology is a mathematical tool for modeling the concept of proximity which allows us to follow structural transformation processes as they evolve. Therefore, it can be a good tool for data analysis. In this section, we recall two applications of pretopology for data analysis. Firstly, we present a pretopology approach for structure analysis using minimal closed subsets from the work of [111]. Secondly, we will present a pretopological approach for clustering data via Method of Classification using Pretopology with Reallocation (MCPR) [114] based on the k-means algorithms with the initial parameters is given from structure process and pseudo-closure distance is used to measure the distance between two elements which can be worked with both numeric and categorical data.

1.5.1 A pretopological approach for structural analysis

The data of a structural analysis problem are represented by a finite set X, composed of elements which are related to each other by some form of connection. The goal of the structural analysis is to highlight groups of "interdependent" elements. Depending on the concept retained to formulate the connection between the elements of a population, structural analysis problems can be approached in various manners: from a metric point of view if distance has been retained, in terms of topological space if neighborhoods have been chosen, or by graph theory.

In this subsection, we introduce the concepts of minimal closed subsets that can be applied for analyzing a structure of objects. The advantage of this approach is that it enables us to formulate and treat structural analysis problems in a unified manner, even if the connections between elements are diverse. It is enough to select a pseudo-closure adapted to the application. Obviously, according to certain definitions of pseudo-closures, this approach provides results which can also be obtained using usual methods like graph algorithms or single linkage method. Please refer the work of [111] for more details.

1.5.1.1 Minimal closed subsets

Let (X, a, i) be $\mathcal{V}$-type pretopological space.

Definition 1.22. *An elementary closed subset, noted F_x , is the closure of one element set $\{x\}$ of X.*

We denote $\mathcal{F}_e$ as the family of elementary closed subsets. So in a $\mathcal{V}$-type pretopological space, we get:

- $\forall x \in E, \exists F_x$: closure of $\{x\}$.

- $\mathcal{F}_e = \{F_x | x \in E\}$

Proposition 1.27. *Two distinct elementary closed subsets F_x and F_y are either disjoin ($F_x \cap F_y = \emptyset$) or contain a nonempty intersection such that for all $z \in F_x \cap F_y$, we have $F_z \subset F_x \cap F_y$.*

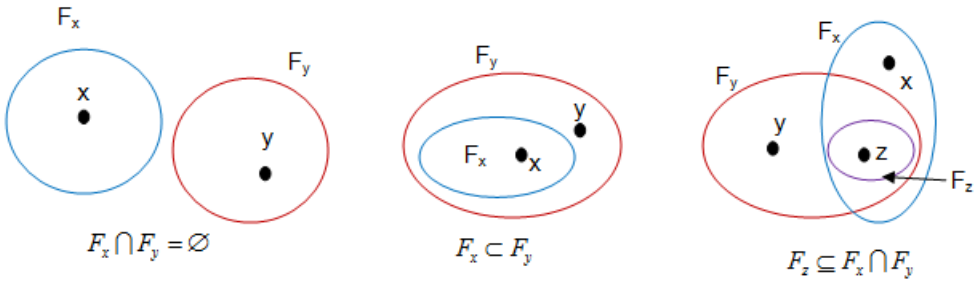


Figure 1.19: Properties of two elementary closed subsets F_x and F_y.

From the proposition 1.27, we can define a minimal closed subset of $\mathcal{F}_e$ in terms of inclusion such as:

Definition 1.23. *F_{min} is called a minimal closed subset if and only if F_{min} is a minimal element in terms of inclusion in $\mathcal{F}_e$.*

We denote $\mathcal{F}_m = \{F_{m_j}, j = 1, 2, \ldots, k\}$, the family of minimal closed subsets, the set of minimal closed subsets in $\mathcal{F}_e$. The following proposition proves that minimal closed subsets of $\mathcal{F}_m$ can be discovered in the elementary closed subsets $\mathcal{F}_e$.

Proposition 1.28. *$F \in \mathcal{F}_m \Leftrightarrow F \in \mathcal{F}_e$ and F is minimal by inclusion in $\mathcal{F}_e$.*

Algorithm 1.1 Elementary closed subsets algorithm.

Require: population X, pseudo-closure $a(.)$
1: **procedure** ElementaryClosedSubsets(X, a)
2: $\mathcal{F}_e = \emptyset$
3: **for** $x \in X$ **do**
4: **Begin**
5: $F = a(\{x\})$
6: **while** $(a(F) \neq F$ **do**$)$ $F = a(F)$
7: **End**
8: $\mathcal{F}_e$.append(F)
9: **end for**
10: **return** $\mathcal{F}_e$ ▷ Ouput
11: **end procedure**

Algorithm 1.2 presents the way we build *minimal closed subsets* based on the result of *elementary closed subsets* (see algorithm 1.1).

1.5.1.2 Structure analysis process

The underlying idea of the structural analysis method is to first highlight homogenous groups (minimal closed subsets), then those containing them (non-minimal elementary closed subsets) until the structural analysis of the entire population has been completed. Nevertheless, according to proposition 1.28, it is enough to define the inclusion relation on the set of elementary closed subsets: $(\mathcal{F}_e, \subset)$. It is necessary to proceed in three stages:

- The first step consists in determining the set of elementary closed subsets $\mathcal{F}_e$ by associating a closure F_x to all elements x of X by means of the function *ElementaryClosedSubsets* (see algorithm 1.1).

Algorithm 1.2 Minimal closed subsets algorithm.

Require: Elementary Closed Subsets $\mathcal{F}_e$
procedure MinimalClosedSubsets($\mathcal{F}_e$)
  $\mathcal{F}_m = \emptyset$
  loop until $\mathcal{F}_e = \emptyset$
    Begin
      Choose $F \subset \mathcal{F}_e$
      $\mathcal{F}_e = \mathcal{F}_e - F$
      minimal = True
      $\mathcal{F} = \mathcal{F}_e$
      loop until $\mathcal{F} = \emptyset$ and not minimal
        Begin
          Choose $G \in \mathcal{F}$
          If $G \subset F$ **then**
            minimal=False
          Else
          If $F \subset G$ **then**
            $\mathcal{F}_e = \mathcal{F}_e - \{G\}$
          $\mathcal{F} = \mathcal{F} - G$
        End
    End
  If (minimal =True) &&($F \notin \mathcal{F}_m$) **then**
    $\mathcal{F}_m = \mathcal{F}_m.append(F)$
  return $\mathcal{F}_m$ ▷ Ouput
end procedure

- The second step aims at searching for minimal closed subsets ($\mathcal{F}_m$ by means of the function *MinimalClosedSubsets* (see algorithm 1.2). In line with the previous statement, this means enumerating the set of elementary minimal closed subsets by inclusion in $\mathcal{F}_e$ (see proposition 1.28).

- The third step is the structural analysis phase. The aim of this step is to picture the inclusion relation between elements of $\mathcal{F}_e$. This process enables us to generate the structure from each elementary closed subset by means of successive enlargements (see algorithm 1.3).

The structural analysis which is named *StructuralAnalysis*, is presented in algorithm 1.4 [111] as follows:
The inputs of *StructuralAnalysis* procedure are:

- the population X,

- the pseudo-closure $a(.)$ defined on X.

The outputs are:

- the family of the elementary closed subsets $\mathcal{F}_e$,

- the family of the minimal closed subsets $\mathcal{F}_m$,

- the structure characterized by relations of inclusion between minimal closed subsets and elementary ones and relations of inclusion between elementary closed subsets with each other.

By performing the minimal closed subset algorithm, we get the family of minimal closed subsets. This family, by definition, characterizes the structure underlying the dataset X. So, the number of minimal closed subsets is a quite important parameter: it gives us the number of clusters for clustering a set. It can be used as an input for *k-means* algorithm.

Algorithm 1.3 Extract Structure.

Require: Elementary Closed Subsets $\mathcal{F}_e$, Minimal Closed Subsets $\mathcal{F}_m$
procedure ExtractStructure($\mathcal{F}_e, \mathcal{F}_m$)
  $Q = \emptyset$
  for $F \in \mathcal{F}_m$ **do** enqueue(Q,F)
  end for
  while ($Q \neq \emptyset$ **do**)
    Begin
      F =dequeue(Q)
      $\mathcal{F} = \{G \in \mathcal{F}_e, F \subset G \text{ and } F \neq G\}$ ▷ supersets to F
      for $G \in MinimalClosedSubsets(\mathcal{F})$ **do**
        Begin
          If $G \neq Q$ **then**
            enqueue(Q,G)
          G is a descendant of F
        End
      end for
    End
  end while
  return Q ▷ Ouput
end procedure

Algorithm 1.4 Structure Analysis algorithm.

Require: population X, pseudo-closure $a(.)$
procedure StructureAnalysis(X, a)
  $\mathcal{F}_e(X, a)$= ElementaryClosedSubsets(X, a) ▷ Compute family of elementary closed subsets
  $\mathcal{F}_m(X, a)$= MinimalClosedSubsets($\mathcal{F}_e(X, a)$) ▷ Compute $\mathcal{F}_m(X, a)$ by finding in $\mathcal{F}_e(X, a)$
  Structure=ExtractStructure($\mathcal{F}_e(X, a), \mathcal{F}_m(X, a)$) ▷ Extraction of the structure $\mathcal{F}_e(X, a)$
  return $\mathcal{F}_e(X, a), \mathcal{F}_m(X, a), Structure$ ▷ Ouput
end procedure

Table 1.1: Relationship and pseudo-closure data

x	$\mathcal{R}(x)$	$a(x)$	$a^2(x)$	$a^k(x)$	F_x
1	1,2,3	1	1	...	1 ⋆
2	2,3	1,2,3	1,2,3	...	1,2,3
3	2,3	1,2,3	1,2,3	...	1,2,3
4	4,5,7	4,5	4,5	...	4,5 ⋆
5	4,5,6	4,5	4,5	...	4,5 ⋆
6	6	5,6	4,5,6	...	4,5,6
7	7	6,7	5,6,7	...	4,5,6,7
8	8,9	8,9	8,9,10	...	8,9,10 ⋆
9	8,9,10	8,9,10	8,9,10	...	8,9,10 ⋆
10	9,10,11	8,9,10	8,9,10	...	8,9,10 ⋆
11	11,12	10,11	10,11	...	8,9,10,11
12	12,13,15	11,12,13	10,11,12,13,14	...	8,9,10,11,12,13,14
13	12,13,16	12,13,14	11,12,13,14	...	8,9,10,11,12,13,14
14	13,14	14	14	...	14 ⋆
15	15	12,15	11,12,13,15	...	8,9,10,11,12,13,14,15
16	15	13,16	12,13,14,16	...	8,9,10,11,12,13,14,16

1.5.1.3 Toy Example

We recall in this subsection the toy example from the work of [114]. In this work, we study about the toxic diffusion between 16 geographical areas $X = \{x_1, \ldots, x_{16}\}$. The family of reflexive binary relationships reduce to only one relationship: $x_i R x_j$ (x_j pollutes x_i) if the distance between x_i and x_j is less than a positive given threshold r and x_j is higher in 3D space equal than x_i.

We thus get the results presented in the table 1.1 which gives us three information:

- The set of neighborhood of x_i, $\mathcal{R}(x_i) = \{x_j | x_j R x_i\}$, $i = 1, \ldots, 16$.
- The successive pseudo-closures $a^k(x_i)$ of x_i, $i = 1, \ldots, 16$.
- The closure F_{x_i} of x_i, $i = 1, \ldots, 16$.

The final structure shown in Fig.1.20 is obtained as follows: in the first step, we get the minimal closed subsets {{1}, {4, 5}, {8, 9, 10}, {14}} (the red areas in the Fig.1.20). Afterwards, we get the smallest elementary closed subsets which contain the minimal ones: {{1,2,3}, {4, 5, 6}, {4, 5, 7}, {8, 9, 10} } . And so on: {{8, 9, 10, 11}, {9, 10, 11, 12, 13, 14}, {8, 9, 10, 11, 12, 13, 14, 15, 16} }.

In the above table and image, we note that the sets marked by stars (*) are minimal element forming homogeneous groups of the population. They cannot transfer the poison to the other elements but they are influenced by the ones of the group which contains them. The advantage of this method is to help us to analyze the connection between the elements in discrete space.

However, this method only provides a clustering of X in the case in which the relationship between elements of X is a symmetric one. In many practical situations, it is not the case, so we propose using this minimal closed subsets algorithm as a pre-treatment for classical clustering methods, in particular for the k-means method. Two possible cases can thus occur at the end of the minimal closed subsets algorithm:

- $\mathcal{F}_m$ provides a partition of X. The clustering is obtained.
- $\mathcal{F}_m$ does not provide a partition of X. In this case, we must perform the second step that we present in the following in order to build a clustering based on the result obtained by the previous stage.

1.5.2 Method of Classification using Pretopology with Reallocation

Algorithm 1.5 Method of Classification using Pretopology with Reallocation (MCPR) algorithm.

Require: population X, pseudo-closure $a(.)$
procedure MCPR(X, a)
  $\mathcal{F}_e$= ElementaryClosedSubsets(X, a)
  $\mathcal{F}_m$= MinimalClosedSubsets($\mathcal{F}_e$)
  $k = |\mathcal{F}_m|$ ▷ number of clusters
  $M = \{m_i\}_{,i=1,\ldots,k}\, , m_i = Centroid(F_{m_i})$ ▷ Initial centroids
  while clusters centroids changed **do**
    for each x $\in E - M$ **do**
      compute $\delta(x, m_i), i = 1, \ldots, k$
      find m_0 with $\delta(x, m_0) = min\delta(x, m_i)_{i=1,\ldots,k}$
      $F_{m_0} = F_{m_0} \cup \{x\}$
    end for
    Recompute clusters centroids M.
  end while
  return $Clusters = \{F_1, F_2, \ldots, F_k\}$ ▷ Output
end procedure

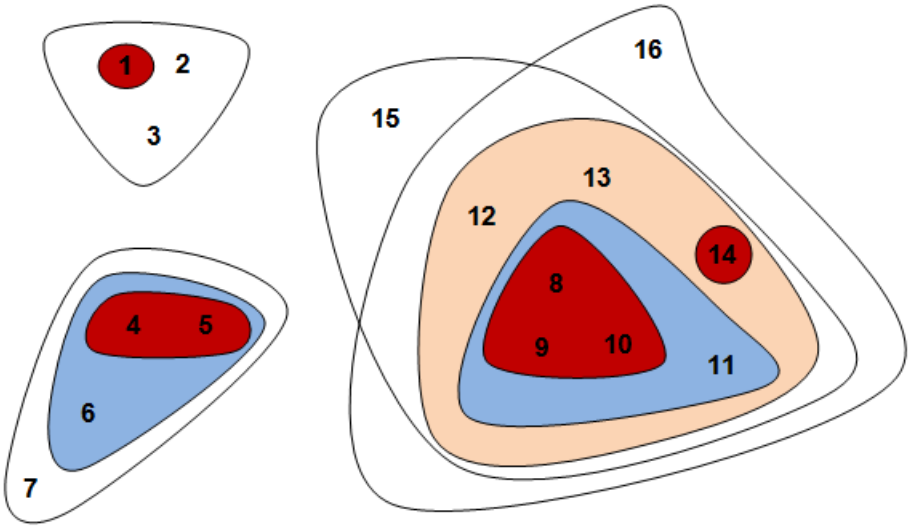


Figure 1.20: Result of Structure Process

Method of Classification using Pretopology with Reallocation) (MCPR) [113, 114] is based on the k-means algorithm. The initial parameters are found from the result of minimal closed subsets and pseudo-closure distance is used to measure the similarity between two elements. MCPR method works in two steps:

- The first step is a structuration process: this process consists in structuring the population by the minimal closed subsets method.
- The second step is the partitioning process: from the previous step, the number k of clusters is directly determined as well as the germs of the clusters. We use the k-means algorithm for partitioning process with pseudo-closure distance for measuring the distance between two elements and interior-pseudo-closure distance to re-compute centers.

Please refer the work of [114, 113] for more details.

1.5.2.1 Pseudo-closure distance

In standard *k-means*, the centroid of a cluster is the average point in the multidimensional space. Its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster which are not effective with categorical data analysis. On the other hand, the pseudo-closure distance is used to examine the similarity using both numeric and categorical data. Therefore, it can contribute to improving the classification with k-means.

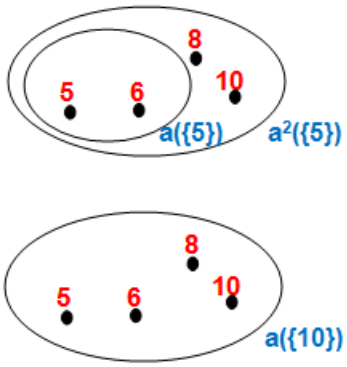
Definition 1.24. *We define $\delta(A, B)$ pseudo-closure distance between two subsets A and B of a finite set E:*

$$k_0 = \min(\min\{k | A \subset a^k(B)\}, \infty)$$

$$k_1 = \min(\min\{k | B \subset a^k(A)\}, \infty)$$

$$\delta(A, B) = \min(k_0, k_1)$$

where $a^k(.) = a^{k-1}(a(.))$



$$k_0 = \min(\min\{k \mid \{5\} \in a^k(\{10\})\}, \infty) = \min(1, \infty) = 1$$
$$k_1 = \min(\min\{k \mid \{10\} \in a^k(\{5\})\}, \infty) = \min(2, \infty) = 2$$
$$\delta(5,10) = \min(1, 2) = 1$$

Figure 1.21: Example of pseudo-closure distance

Table 1.2: Interior-distance for each elements in Group {8,9,10}

$\delta(x,y)$	8	9	10	D(x)
8	0	1	2	1
9	1	0	1	2/3
10	2	1	0	1

Definition 1.25. *We call $D_A(x)$ interior-pseudo-distance of a point x in a set A:*

$$D_A(x) = \frac{1}{|A|} \sum_{y \in A} \delta(x, y).$$

In case where A and B are reduced to one element x and y, we get the distance $\delta(x, y)$ (see Fig.1.21 for an example). For clustering with k-means algorithm, we use the pseudo-closure distance $\delta(x, y)$ to compute distance between two elements and the interior-pseudo-distance $D_A(x)$ to compute centroid of A (x_0 is chosen as centroid of A if $D_A(x_0) = min_{x \in A} D_A(x)$).

1.5.2.2 MCPR algorithms

Method of Classification using Pretopology with Reallocation (MCPR) [114] algorithm is presented in algorithm 1.5

1.5.2.3 Toy Example (cont)

In order to have a better understanding of this algorithm, let us return to the previous toy example in subsectiton 1.5.1.3. From the result of structure process, we get the minimal closed subsets: $\mathcal{F}_m = \{\{1\}, \{4, 5\}, \{8, 9, 10\}, \{14\}\}$. Based on this information, we determine the initial parameters:
Firstly, we choose number of clusters $k = |\mathcal{F}_m| = 4$.
Secondly, we determine the initial centroids: we use inter-pseudo-closure distance to define the center of each cluster. For example, we computer the inter-pseudo-closure distance to decide the centroid of cluster {8, 9, 10}. From the result presented in the table 1.2, 9 is designated centroid of group *{8, 9, 10}*. Similarly, we can obtain the initial set of centroids is M = {{1}, {4}, {9}, {14}}.
It is then possible to use the classical k-means algorithm in conjunction with the pseudo-closure distance. The distance table between the elements of X and the initial centroids M = {{1}, {4}, {9}, {14}} is given in the table 1.3. From the result presented in the table 1.3, we get the new clusters (see Fig.1.22a):

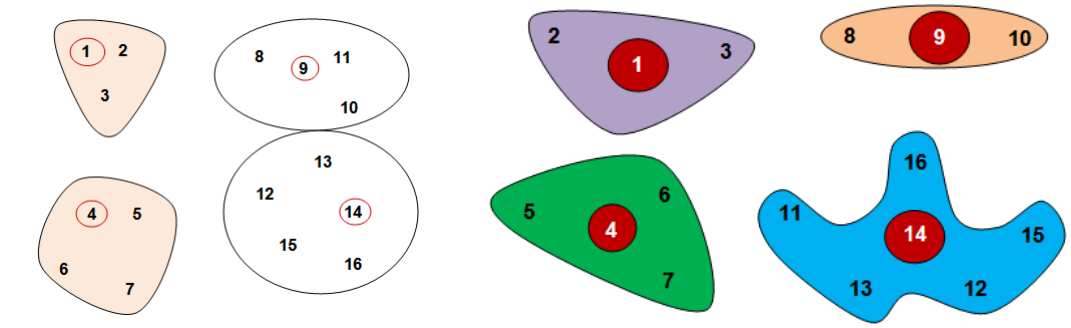
Table 1.3: Distance between data points and centroids

δ	2	3	5	6	7	8	10	11	12	13	15	16
1	**1**	**1**	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
4	∞	∞	**1**	**2**	**3**	∞	∞	∞	∞	∞	∞	∞
9	∞	∞	∞	∞	∞	**1**	**1**	**2**	3	4	4	5
14	∞	∞	∞	∞	∞	∞	∞	∞	**2**	**1**	**3**	**2**

$$Clusters = \{\{1,2,3\},\{4,5,6,7\},\{9,8,10,11\},\{14,12,13,15,16\}\}.$$

By repeating this process, the data of the example lead us to the final partition of X (see Fig.1.22b):

$$\{\ \{1,\ 2,\ 3\},\ \{4,\ 5,\ 6,\ 7\},\ \{8,\ 9,\ 10\},\ \{11,\ 12,\ 13,\ 14,\ 15,\ 16\}\}.$$



(a) Calculation of centers and groups

(b) Result of clustering using MCPR

Figure 1.22: Example of clustering using MCPR

1.6 Pretopology as a tool for network modeling

1.6.1 Pretopology as an extension of Graph theory

1.6.1.1 Graphs in the Claude Berge sense

By using the knowledge from multivalued function, *Claude Berge* [24] defined a graph such as:

Definition 1.26. *A graph, which is denoted by $G = (V, \Gamma)$, is a pair consisting of a set V of vertices or nodes and a multivalued function Γ mapping V into $\mathcal{P}(V)$.*

The pair (x, y), with $y \in \Gamma(x)$ is called an arc or edge of the graph. We therefore can also denote a graph by a pair $G = (V, E)$, which V is a set of nodes and E is a set of edges. Conversely, if we denote a graph as $G = (V, E)$, we can define the Γ function as: $\Gamma(x) = \{y \in V | (x, y) \in E\}$. $\Gamma(x)$ is a set of neighbors of node x.

1.6.1.2 Pretopology as an extension of Graph theory

In this part, we show reflexive graph (V, Γ) which is a special case of pretopology. More specifically, as it is known, a finite reflexive graph (V, Γ) complies the property: $\forall A \subset V, a(A) = \cup_{x \in A} a(\{x\})$ where pseudo-closure function defined as $a(A) = \cup_{x \in A} \Gamma(x)$. For this reason, graph may be represented by a $\mathcal{V}_D$-type pretopological space. Conversely, we can build a pretopology space (V, a) presented a graph such as: $a(A) = \{x \in V | \Gamma(x) \cap$

$A \neq \emptyset\}$ where $\Gamma(x) = \{y \in V | x\, R\, y\}$ built from a binary relation R on V. Therefore, a graph (V, Γ) is a pretopological space (V, a) in which the pseudo-closure function built from a binary relation or built from a neighborhood function in equation (1.13).

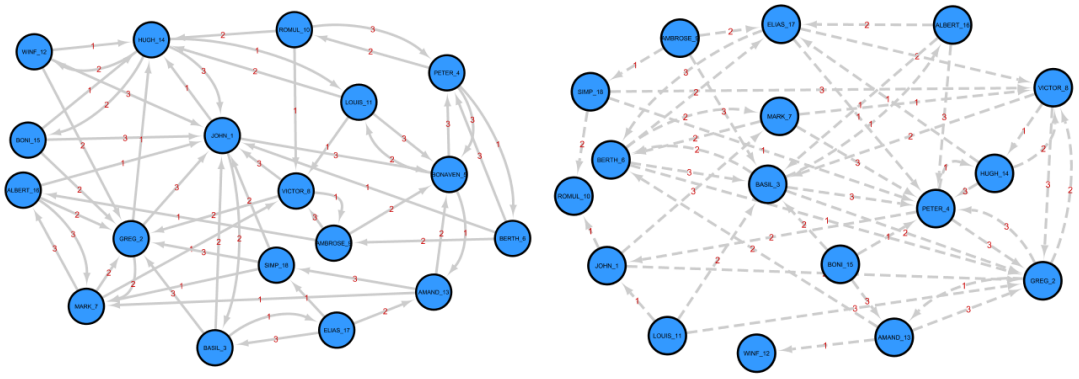
By using a graph, a network is represented by only one binary relation. In the real world, however, a network is a structure made of nodes that are tied by one or more specific types of binary or value relations. As we show in the previous, by using pretopology theory, we can generalize the definition of complex network such as:

Definition 1.27. *(Pretopology network) A pretopology network, which is denoted by* $G^{(Pretopo)} = (V, a)$*, is a pair consisting of a set* V *of vertices and a pseudo-closure function* $\boldsymbol{a(.)}$ *mapping* $\mathcal{P}(V)$ *into* $\mathcal{P}(V)$*.*

1.6.2 Example of a Complex Group Interactions Model

In this section, we show an example in which we can deal with complex interactions in social networks modeling. We use the *Sampson Monastery* dataset [3] from the work of Samuel F. Sampson [167] which consists of social relations among a set of 18 monk-novitiates preparing to enter a monastery and has been often used in sociology studies. The data include a variety of relations, such as *esteem, liking, dislike, influence, praise* and so on. A lot of relations are coded, however, in this example, we concentre on two of them: *like* (see Fig.1.23a) and *dislike* (see Fig.1.23b). Since, each member ranked only his top three choices on that tie, relations are nonsymmetric and weighted with three integer values, from 1 to 3 (3 indicates the highest or first choice and 1 the last choice).

In our model, for group coalitions, we consider the question: "who wants to join the group?". Based on the two relations *like* and *dislike*, we can see that: people who have the greatest like for x should join his group more than the others but this person will not accept in his group people he does not like, even if these people have like for him. The question here is how we can model this kind of social networks.



(a) Sampson *Like* Network

(b) Sampson *DisLike* Network

Figure 1.23: Sampson Networks with two relations *like* and *dislike*.

To overcome this aim, we firstly set two valued relations as fuctions $like(x, y)$, $dislike(x, y)$ from $X \times X$ to $\mathbb{N}$ with X, *like*, *dislike* being the set containing individuals, *like* relation, *dislike* relation respectively and then propose the way a new member x joins to a group $A \in \mathcal{P}(X)$ with two steps:

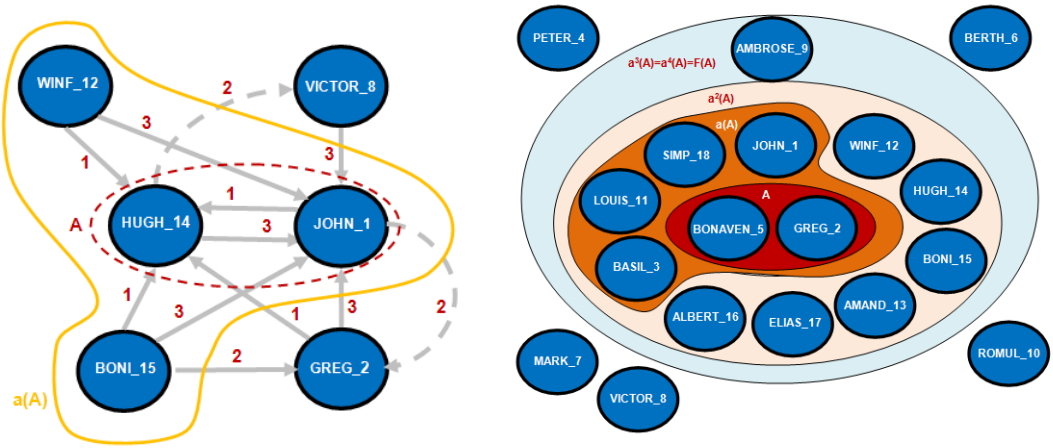
ST1: if person x has *like* relation with some members in group A and the sum of all *like* weights is greater than a chosen threshold θ_1 then he will send a message **"Want to join"** to this group.

[3] http://www.casos.cs.cmu.edu/computational_tools/datasets/sets/sampson

ST2: when group A receive a "Want to join" message from person x, the group will see if the current members of the group have any "problem" with x by considering the *dislike* relation. If sum of all *dislike* weights from all member of the group to x is less than a chosen threshold θ_2, they will accept this member and send a message **"Accept for joining"**.

We therefore can build the group coalitions model of $A \in \mathcal{P}(X)$ by defining pseudo-closure $a(A)$ as:

$$a(A) = \{y \in X | \sum_{x \in A} like(y, x) > \theta_1 \wedge \sum_{x \in A} dislike(x, y) < \theta_2\}; \theta_1, \theta_2 \in \mathbb{N} \tag{1.21}$$



(a) Pseudo-closure of A.

(b) Largest group started with two members found according to the neighborhood definition.

Figure 1.24: Example of the complex group interactions model with $\theta_1 = 3, \theta_2 = 2$. *Like* relation is plain and *dislike* relation is dashed.

By setting up the values of two parameters θ_1, θ_2, we can capture strong or weak restriction for growing a group. If we apply this pseudo-closure on a set A, people would be in $a(A)$ if they have *like* relation with one or several persons of A (according to θ_1) and if no people or a few people of A (according to θ_2) have *dislike* relation with them. Figure 1.24 shows an example with $\theta_1 = 3, \theta_2 = 2$. Figure 1.24a shows an illustration of pseudo-closure function of a set A=[HUGH_14, JOHN_1]. We can see WINF_12 will join to group A since he has *like* relations with JOHN_1 (weight =3) and HUGH_14 (weight =1) and nobody in this group has any "problem" with him. The same situation occurs with BONI_15 but it is different with GREG_2 since JOHN_1 *dislikes* to him with weight = 2 although GREG_2 would like to join this group. VICTOR_8 does not want to joint the group since he just likes only JOHN_1 with weight=3 and even if he wants to joint the group in the future, the group does not accept since HUGH_14 dislikes to him with weight=2. The pretopology space built in this example is neither V_D-type nor V_S-type since $a([1]) \cup a([14]) = [1] \cup [14]$ while $a([1, 14]) = [1, 12, 14, 15]$ with 1,12,14,15 denote JOHN_1, WINF_12, HUGH_14, BONI_15 respectively.

In this model, the question of finding the largest group started with 2 members following the rules of our neighborhood can be treated by building closure function. When the closure function is applied to all subsets of X which contains two elements, it reveals that [GREG_2, BONAVEN _5] is the group that can form the largest amount of people in the network (Fig.1.24b).

1.6.3 Path-connectedness in pretopological spaces

As we show in the previous, pretopology can be considered as an extension of Graph. In this subsection, we will generalize some basic concepts on graph theory for pretopology such as path, chain, path-connectedness.

Definition 1.28. *Let (X, a) be a $\mathcal{V}$-type pretopological space and $A, B \in \mathcal{P}(X)$, $A \neq \emptyset, B \neq \emptyset$. There exists a path in (X, a) from A to B if and only if $B \subseteq F(A)$.*

We note that in $\mathcal{V}$-type pretopological space, the closure of any set A, denoted $F(A)$, always exists. $F^{'}$, the inverse of the closure generated by a, is defined by:

$$\forall A \in E, F^{'}(A) = \{y \in E | F(\{y\}) \cap A \neq \emptyset\} \tag{1.22}$$

We note $a = F^{'}F$ (a is the composed of the mapping $F^{'}$ and F) and $F^{''}$ is the closure according to a.

Definition 1.29. *Let (X, a) be a $\mathcal{V}$-type pretopological space and $A, B \in \mathcal{P}(X), A \neq \emptyset, B \neq \emptyset$. There exists a chain in (X, a) from A to B if and only if $B \subseteq F^{''}(A)$.*

Remark: If a is of $\mathcal{V}$-type then $a^n, F, a, F^{''}$ also are of $\mathcal{V}$-type and $F^{'}$ is of $\mathcal{V}_S$-type. If a is of $\mathcal{V}_S$-type then $a^n, F, a, F^{''}, F^{'}$ are also $\mathcal{V}_S$-type.

Proposition 1.29 ([60], Property 2)**.** *Let (X, a) be a $\mathcal{V}$-type pretopological space. Let $x \in X$ and $y \in X$.*

*i. If there exists a sequence $x_0, x_1, \ldots, x_n$ of elements of X such as $x_0 = x, x_n = y$ with $\forall j = 0, 1. \ldots, n-1, x_{j+1} \in a(x_j)$ or $x_{j1} \in a(x_{j+1})$ then there exists a **chain** in (X, a) from x to y.*

*ii. If there exists a sequence $x_0, x_1, \ldots, x_n$ of elements of X such as $x_0 = x, x_n = y$ with $\forall j = 0, 1. \ldots, n-1, x_{j+1} \in a(x_j)$ then there exists a **path** in (X, a) from x to y.*

Proposition 1.30 ([60], Property 3)**.** *Let (X, a) be a $\mathcal{V}_S$-type pretopological space. Let $x \in X$ and $y \in X$.*

*i. There exists a **chain** in (X, a) from x to $y \Leftrightarrow$ it exists a sequence $x_0, x_1, \ldots, x_n$ of elements of X such as $x_0 = x, x_n = y$ with $\forall j = 0, 1. \ldots, n-1, x_{j+1} \in a(x_j)$ or $x_{j1} \in a(x_{j+1})$.*

*ii. There exists a **path** in (X, a) from x to $y \Leftrightarrow$ it exists a sequence $x_0, x_1, \ldots, x_n$ of elements of X such as $x_0 = x, x_n = y$ with $\forall j = 0, 1. \ldots, n-1, x_{j+1} \in a(x_j)$ then there exists a **path** in (X, a) from x to y.*

Proposition 1.31 ([60], Proposition 2)**.** *Let (X, a) be a $\mathcal{V}$-type pretopological space.*

i. If $\forall x, y \in X$, it exists a chain in (X, a) from x to y then (X, a) is connected.

ii. (X, a) is strongly connected $\Leftrightarrow \forall x, y \in X$, it exists a path in (X, a) from x to y.

Proposition 1.32 ([60], Proposition3)**.** *Let (X, a) be a $\mathcal{V}_S$-type pretopological space. (X, a) is strongly connected $\Leftrightarrow \forall x, y \in X$, it exists a chain in (X, a) from x to y.*

1.7 Conclusion

In this chapter, we presented basic concepts of pretopology as a generalized topology space for modeling the concept of proximity that can be applied for solving some problems in complex systems. From the point of view the computer science, we also proposed the practical way to build pretopological spaces for application and illustrated this method in many situations. For applications of pretopology, we recalled two applications that we will apply for our works in the two next parts. The first related to data analysis with structure analysis via the concept of *minimal closed subset* and clustering via *the method of classification using pretopology with Reallocation* (MCPR). The second related to complex networks by proposing pretopology as a general framework for complex network modeling.

Part II

Pretopology and Topic Modeling for Text mining

Chapter 2

Latent Dirichlet Allocation and Its Application

2.1 Introduction

In this age of information technology, a significant portion of unstructured data in textual format is being collected, digitized and stored in many forms such as e-mail, news, blogs, research articles, web pages, social media platforms and diverse documents type. This data is used to train different models that are built to process information retrieval queries from diverse systems like modern libraries or search engines. To have a better way of managing this kind of data, it requires using new techniques or tools that deal with automatically organizing, searching, indexing, and browsing large collections. On the basis of today's research of machine learning and statistics, it has developed new techniques for finding patterns of words in document collections using hierarchical probabilistic models. These models are called "topic models".

Technically speaking, Topic modeling refers to a group of machine learning algorithms that infer the latent structure behind a collection of documents. The main importance of topic modeling is to discover patterns of word-use and how to connect documents that share similar patterns. So, the idea of topic models is that term which can be working with documents and these documents are mixtures of topics, where a topic is a probability distribution over words. In other words, topic model is a generative model for documents. It specifies a simple probabilistic procedure by which documents can be generated.

The three main topic models are:

- Latent Semantic Analysis (LSA) [109] or Latent Semantic Indexing (LSI) uses the Singular Value Decomposition (SVD) methods to decompose high-dimensional sparse matrix to three matrices: one matrix that relates words to topics, another one that relates topics to documents and a diagonal matrix of singular value.

- Probabilistic Latent Semantic Analysis (pLSA) [93] is a probabilistic model that treats the data as a set of observations coming from a generative model. The generative model includes hidden variables representing the probability distribution of the words in the topics and the probability distribution of the topics in the words.

- Latent Dirichlet Allocation (LDA) [26] is a Bayesian extension of probabilistic Latent Semantic Analysis. It defines a complete generative model with a uniform prior and full Bayesian estimator.

LSA [109] can reduce the dimension of the sparse matrix but it is not a generative model. The pLSA approach models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of "topics". While Hofmann's work is a useful step toward probabilistic modeling of text, pLSA model does not provide any probabilistic model at the document level which makes it difficult to generalize it to model new unseen documents. Blei et al. [26] extended this model by introducing a Dirichlet prior to mixture weights of topics per documents and called the model LDA. In this chapter, we describe the LDA method.

In LDA [26] model, a document is a mixture of topics and a topic is a mixture of words. The key problem in LDA is posterior inference. This refers to reversing the defined generative process and learning the posterior distributions of the latent variables in the model given the observed data. Learning the different parameters of the latent mixtures is a problem of Bayesian inference. Two of the approximate inference algorithms are used to overcome this problem: the variational inference (as used in the original LDA paper [26]) and Gibbs Sampling (as proposed by [82]). The former is faster and the latter is slower but more accurate.

In information retrieval systems we need both speed and accuracy. However, for massive corpora of text, the iterations of Gibbs sampling are extremely slow and can require days or even months of execution time [148]. Clusters of computers can resolve this problem. To this aim, we propose a distributed version of Gibbs sampling built on Spark. Spark is a cluster computing and data-parallel processing platform for applications that focus on data-intensive computations. The main idea of the proposed algorithm is to make local copies of the parameters across the processors and synchronize the global counts matrices that represent the coefficients of LDA mixtures.

LDA is a popular algorithm for topic modeling that has been widely used in the NLP area. However, this methods is unsupervised and therefore does not allow to include knowledge to guide the learning process. Therefore, we present a semi-supervised version of LDA (ss-LDA) based on the works of [161, 132]. The supervision of the process is within two levels: word level and document level. By connecting the ss-LDA and Random Forest classifier, we propose a new methodology for a multilayer soft classification for web pages.

This chapter is organized as follows. Firstly, we describe the LDA and the collapsed version of Gibbs sampling in section 2. We then present the AD-LDA and our implementation with Spark in section 3. Section 4 presents the ss-LDA and its application on multilayer classification of web pages. Then it is followed by conclusion in section 5.

2.2 Latent Dirichlet Allocation

Topic Modeling is a method for analyzing large quantities of unlabeled data. For our purposes, a topic is a probability distribution over a collection of words and a topic model is a formal statistical relationship between a group of observed and latent (unknown) random variables that specifies a probabilistic procedure to generate the topics [26, 93, 82, 176]. In many cases, there exists a semantic relationship between terms that have high probability within the same topic – a phenomenon that is rooted in the word co-occurrence patterns in the text and that can be used for information retrieval and knowledge discovery in databases.

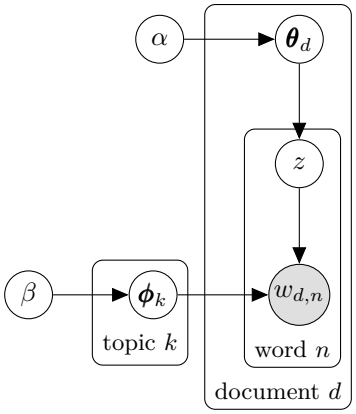


Figure 2.1: Bayesian Network (BN) of Latent Dirichlet Allocation.

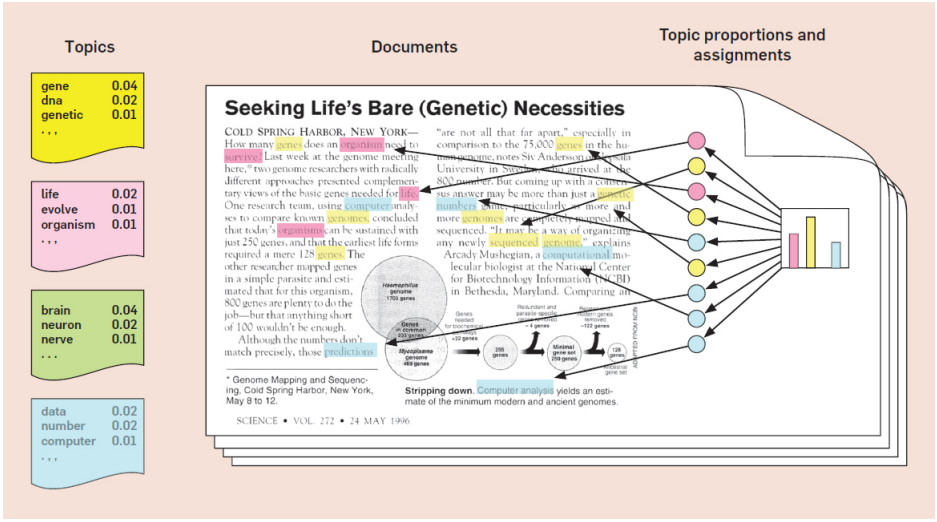


Figure 2.2: LDA Example.

2.2.1 Latent Dirichlet Allocation

LDA by Blei et al. [26] is a generative probabilistic model for collections of grouped discrete data. Each group is described as a random mixture over a set of latent topics where each topic is a discrete distribution over the vocabulary collection. LDA is applicable to any corpus of grouped discrete data. In our work, we refer to the standard Natural Language Processing (NLP) use case where a corpus is a collection of documents, and the discrete data are represented by the occurrence of words.

LDA is a probabilistic model for unsupervised learning, it can be seen as a Bayesian extension of the pLSA [93]. More precisely, LDA defines a complete generative model which is a full Bayesian estimator with a uniform prior while pLSA provides a Maximum Likelihood (ML) or Maximum a Posterior (MAP) estimator. For more technical details we refer to the work of Gregor Heinrich [87]. The generative model of LDA is described with the probabilistic graphical model [103] in Fig.2.1 and an example [25] shows in Fig.2.2.

In this LDA model, different documents d have different topic proportions θ_d. In each position in the document, a topic z is then selected from the topic proportion θ_d. Finally, a word is picked from all vocabularies based on their probabilities ϕ_k in that topic z. θ_d and ϕ_k are two Dirichlet distributions with α and β as hyperparameters. We assume symmetric Dirichlet priors with α and β having a single value.

The hyperparameters specify the nature of the priors on θ_d and ϕ_k. The hyperparameter α can be interpreted as a prior observation count of the number of times a topic z is sampled in document d [176]. The hyper hyperparameter β can be interpreted as a prior observation count on the number of times words w are sampled from a topic z [176].

The generative process for a document collection D under the LDA model is as follows:

1. For $k = 1, \ldots, K$:
   (a) $\phi^{(k)} \sim Dirichlet(\beta)$
2. For each document $d \in D$:
   (a) $\theta_d \sim Dirichlet(\alpha)$
   (b) For each word $w_i \in d$:
      i. $z_i \sim Multinomial(\theta_d)$
      ii. $w_i \sim Multinomial(\phi^{(z_i)})$

The advantage of the LDA model is that interpreting at the topic level instead of the word level allows us to gain more insights into the meaningful structure of documents since noise can be suppressed by the clustering process of words into topics. Consequently, we can use the topic proportion in order to organize, search, and classify a collection of documents more effectively.

2.2.2 Inference with Gibbs sampling

In this subsection, we specify a topic model procedure based on the LDA and Gibbs Sampling.

The key problem in Topic Modeling is posterior inference. This refers to reverse the defined generative process and learning the posterior distributions of the latent variables in the model given the observed data. In LDA, this amounts solving the following equation:

$$p(\theta, \phi, z|w, \alpha, \beta) = \frac{p(\theta, \phi, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{2.1}$$

Unfortunately, this distribution is intractable to compute [87]. The normalization factor in particular, $p(w|\alpha, \beta)$, cannot be computed exactly. However, there are a number of approximate inference techniques available that we can apply to the problem including variational inference (as used in the original LDA paper [26]) and Gibbs Sampling (as proposed by [82]) that we shall use.

Gibbs Sampling is one member of a family of algorithms from the Markov Chain Monte Carlo (MCMC). The MCMC algorithms aim to construct a Markov chain that has the target posterior distribution as its stationary distribution. In other words, after a number of iterations of stepping through the chain, the samples from the distribution should converge to desired close sample from the posterior. Gibbs Sampling is based on sampling from conditional distributions of the variables of the posterior.

For example, to sample x from the joint distribution $p(x) = p(x_1, \ldots, x_m)$, where there is no closed form solution for $p(x)$, but a representation for the conditional distributions is available, using Gibbs Sampling one would perform the following:

1. Randomly initialize each x_i
2. For $t = 1, \ldots, T$:
   2.1. $x_1^{t+1} \sim p(x_1 \mid x_2^{(t)}, x_3^{(t)} \ldots, x_m^{(t)})$

2.2. $x_2^{t+1} \sim p(x_2 \mid x_1^{(t+1)}, x_3^{(t)} \ldots, x_m^{(t)})$

2.m. $x_m^{t+1} \sim p(x_m \mid x_1^{(t+1)}, x_2^{(t+1)} \ldots, x_{m-1}^{(t+1)})$

This procedure is repeated a number of times until the samples begin to converge to what would be sampled from the true distribution. While convergence is theoretically guaranteed with Gibbs Sampling, there is no way of knowing how many iterations are required to reach the stationary distribution. Therefore, diagnosing convergence is a real problem with the Gibbs Sampling approximate inference method. However, in practice, it is quite powerful and has fairly good performance. Typically, an acceptable estimation of convergence can be obtained by calculating the log-likelihood or even, in some situations, by inspection of the posteriors.

For LDA, we are interested in the proportions of the topic in a document represented by the latent variable θ_d, the topic-word distributions $\phi^{(z)}$, and the topic index assignments for each word z_i. While conditional distributions - and therefore an LDA Gibbs Sampling algorithm - can be derived for each of these latent variables, we note that both θ_d and $\phi^{(z)}$ can be calculated using just the topic index assignments z_i (i.e. z is a sufficient statistic for both these distributions). Therefore, a simpler algorithm can be used if we integrate out the multinomial parameters and simply sample z_i. This is called a collapsed Gibbs sampler [82, 176].

The collapsed Gibbs sampler for LDA needs to compute the probability of a topic z being assigned to a word w_i, Section all other topic assignments to all other words. Somewhat more formally, we are interested in computing the following posterior up to a constant:

$$p(z_i \mid z_{-i}, \alpha, \beta, w) \tag{2.2}$$

where z_{-i} means all topic allocations except for z_i.

Algorithm 2.1 The LDA Gibbs sampling algorithm.

Require: corpus $\mathcal{D} = (d_1, d_2, \ldots, d_M)$
procedure LDA-Gibbs($\mathcal{D}$, α, β, T)
    randomly initialize z and increment counters
    loop for each iteration
        loop for each word w in corpus $\mathcal{D}$
            Begin
                word $\leftarrow w[i]$
                tp $\leftarrow z[i]$
                $n_{d,tp} -= 1; n_{word,tp} -= 1; n_{tp} -= 1$
                loop for each topic j $\in \{0, \ldots, K-1\}$
                    compute $P(z_i = j | z_{-i}, w)$
                $tp \leftarrow sample \quad from \quad p(z|.)$
                $z[i] \leftarrow tp$
                $n_{d,tp} += 1; n_{word,tp} += 1; n_{tp} += 1$
            End
    Compute $\phi^{(z)}$
    Compute θ_d
    return $z, \phi^{(z)}, \theta_{\mathcal{D}}$ ▷ Output
end procedure

Equation (2.3) shows how to compute the posterior distribution for topic assignment.

$$P(z_i = j | z_{-i}, w) \propto \frac{n_{-i,j}^{w_i} + \beta}{n_{-i,j}^{(\cdot)} + V\beta} \frac{n_{-i,j}^{d_i} + \alpha}{n_{-i,\cdot}^{d_i} + K\alpha} \tag{2.3}$$

where $n_{-i,j}^{w_i}$ is the number of times word w_i was related to topic j. $n_{-i,j}^{(\cdot)}$ is the number of times all other words were related with topic j. $n_{-i,j}^{d_i}$ is the number of times topic j was related with document d_i. The number of times all other topics were related with document d_i is annotated with $n_{-i,\cdot}^{d_i}$. Those notations were taken from the work of Thomas Griffiths and Mark Steyvers [82].

$$\hat{\phi}_j^{(w)} = \frac{n_j^{(w)} + \beta}{n_j^{(\cdot)} + V\beta} \tag{2.4}$$

$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_\cdot^{(d)} + K\alpha} \tag{2.5}$$

Equation (2.4) is the Bayesian estimation of the distribution of the words in a topic. Equation (2.5) is the estimation of the distribution of topics in a document. Algorithm 2.1 presents the proceduce for inference LDA with Gibbs sampling.

2.3 Approximate Distributed LDA and its implementation in Spark

The LDA is one of the most used topic models to discover complex semantic structure. However, for massive corpora of text LDA can be very slow and can require days or even months. This problem created a particular interest in parallel solutions, like the Approximate Distributed LDA (AD-LDA) [148], where clusters of computers are used to approximates the popular Gibbs sampling used by LDA. Nevertheless, this solution has two main issues: first, requiring local copies on each partition of the cluster (this can be inconvenient for large datasets). Second, it is common to have read/write memory conflicts. In this section, we propose a new implementation of the AD-LDA algorithm where we provide computation in memory and a good communication between the processors. The implementation was made possible with the syntax of Spark. We show empirically with a set of experimentations that our parallel implementation with Spark has the same predictive power as the sequential version and has a considerable speedup. We finally document an analysis of the scalability of our implementation and the super-linearity that we obtained. We provide an open source version of our Spark LDA.

The result of this section has been published in the *Proceedings of the Seventh Symposium on Information and Communication Technology, SoICT 2016* [169].

2.3.1 Approximate Distributed LDA

In the AD-LDA model first proposed by Newman et al. [148], a corpus is divided into P processors, with approximately $\frac{D}{P}$ documents on each processor. Then, LDA is implemented on each processor, and Gibbs sampling is simultaneously executed on each $\frac{D}{P}$ documents to approximate a new z_i from the equation (2.3) for every word i in every document j in the collection of documents.

In each iteration each processor has a local copy of the counts matrix word by topic n_p^{wk} and the counts matrix document by topic n_p^{dk} in parallel. A global synchronization described by the equation (2.6) and (2.7) is executed to have global counts n^{wk} and n^{dk}.

$$n_{\text{new}}^{wk} = n_{old}^{wk} + \sum_p (n_{\text{p}}^{wk} - n_{old}^{wk}) = \sum_p n_p^{wk} - (p-1)n_{old}^{wk} \tag{2.6}$$

$$n_{\text{new}}^{dk} = n_{old}^{dk} + \sum_p (n_{\text{p}}^{dk} - n_{old}^{dk}) = \sum_p n_p^{dk} - (p-1)n_{old}^{dk} \tag{2.7}$$

There are two main issues with AD-LDA: first, it needs to store P copies of the global counts for all the processors in parallel, this can be inconvenient for large datasets. Second, we can have read/write memory conflicts on the global counts n^{wk} and n^{dk} which can lower the prediction accuracy.

Spark provides computation in memory and therefore we don't need to store the global counts in parallel. We can avoid read/write memory conflicts with broadcasting temporary copies of the global counts n^{wk} and n^{dk}. In the next section, we explain why we chose to work with Spark instead of Hadoop/MapReduce and then present the algorithm of our implementation.

2.3.2 Implemement AD-LDA with Spark

2.3.2.1 Spark

Spark [187] is a cluster computing and data-parallel processing platform for applications that focus on data-intensive computations. The main component and primary abstraction in Spark is the Resilient Distributed Dataset (RDD). An RDD is a distributed collection of elements in memory that is both fault-tolerant (i.e. Resilient) and efficient (i.e the operation performed on them are parallelized).

Spark automatically distributes the data contained in the different RDDs and applies in parallel different operations (i.e functions) defined within the so-called driver program. The driver program contains the application (e.g Spark LDA) main functions (e.g MCMC methods) and applies them on the cluster.

The prominent difference between MapReduce/Hadoop and Spark is that the former creates an acyclic data flow graph [15] and the latter a lineage graph [187]. As soon as the Hadoop implementation became popular, users wanted to implement more complex applications; iterative algorithms (e.g machine learning algorithms), interactive data mining tools (e.g Python, R) that can not be expressed efficiently as acyclic data flows.

In our work, we used the Gibbs sampling algorithm to approximate the inference. This Monte Carlo algorithm depends on randomness. Hadoop/MapReduce does not allow us to have this randomness because it considers each step of computation in the implemented application as the same no matter where or when it runs. However, this problem can be fixed by seeding a random number generator [142], it adds another layer of complexity to the implementation and can slow it down or affect the complete synchronization of the counts of different parameters distributed on the cluster.

We chose Spark, because it is faster than Hadoop (i.e. computation in memory), allows randomness, iterative jobs, general programming tasks (i.e Machine Learning algorithms are not usually built for Map Reduce tasks). We will use three simple data abstractions provided by the Spark syntax to program the clusters: the RDDs, broadcast variables for the global counts of the different parameters in the Gibbs sampling and the Map and Reduce implementation.

2.3.2.2 The Algorithm

Algorithm 2.2 Algorithm: for distributing the LDA

```
1: procedure SparkLDA
2:     RddData = sc.textFile("textexample.txt")
3:     If FirstIteration then
4:         initialize global counts at Random
5:     loop For each iteration
6:         Begin
7:             rdd = RddData.Map ()
8:             globalcounts = rdd.ReduceAndUpdate()
9:             rddGlobal = globalcounts.parallelize()
10:            rdd = rddGlobal.Map()
11:            φ, θ = rdd.ReduceAndUpdate()
12:        End
13:    return φ, θ
14: end procedure
```

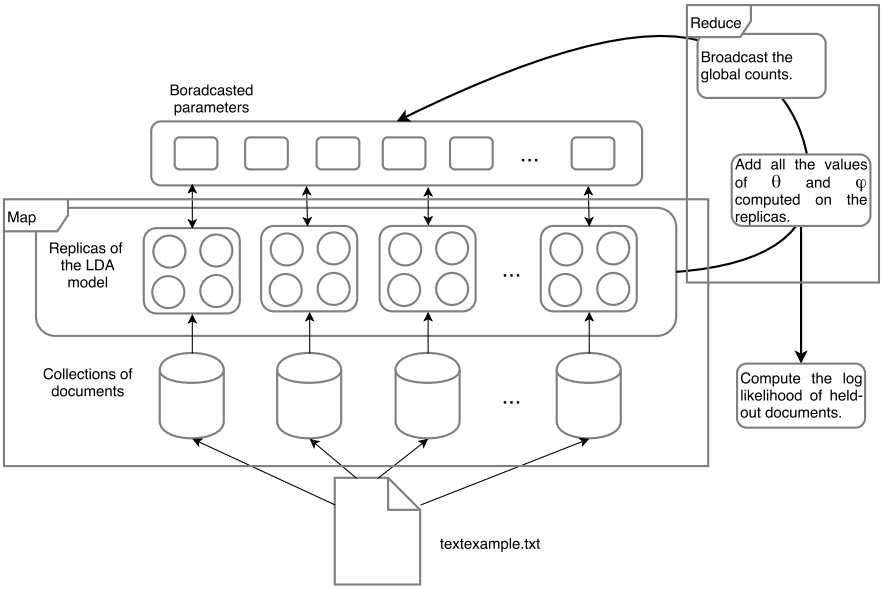


Figure 2.3: Workflow of the Spark implementation. Each iteration is executed within the Map and Reduce framework. After the reduce application the global counts parameters n^{wk} and n^{dk} are broadcasted to the different replicas of the LDA model. The log-likelihood is computed on the held-out documents after each iteration.

Our implementation on Spark of the LDA with collapsed Gibbs sampling (CGS) can be considered as an extension of the AD-LDA algorithm where we aim to provide computation in memory and a good communication between the processors to avoid read/write memory conflicts when we synchronize the global counts n^{wk} and n^{dk}. Spark generalizes the MapReduce model, therefore before presenting the overall algorithm 2.2, we will present what is executed on each processor in the algorithm 2.3 and what is synchronized through all the processors in the algorithm 2.4.

Algorithm 2.3 Algorithm : Mapper

procedure MAP
  $D_1, \cdots, D_p$ = partition (Corpus(D))
  loop For each partition do in parallel
    Begin
      $n_p^{wk} = n^{wk}$
      $n_p^{dk} = n^{dk}$
      loop For each document j and for each word i in V
        Begin
          Sample z_{ij} from n_p^{wk} and n_p^{dk} using CGS.
          Get $n_{p_{new}}^{wk}$ and $n_{p_{new}}^{dk}$ from z_{ij}.
        End
      Get the broadcasted n^{wk} and n^{dk} and compute the ϕ and θ equation (2.4) and (2.5).
    End

end procedure

The first procedure called *Map* described the algorithm 2.3, represents the instructions executed on P processors to sample topics locally. These instructions are the parts of the algorithm that benefit from the improvement of the system (i.e the increase of the number of processors or memory). First, we initialize the global counts with the synchronized versions then we sample z_{ij} and produce new global counts that we broadcast again to compute the distribution of topics per document θ and the distribution of words per topic ϕ.

The second procedure called *ReduceAndUpdate*, in the algorithm 2.4, is executed at

Algorithm 2.4 Algorithm : Reducer

1: **procedure** ReduceAndUpdate
2: n_{new}^{wk} = equation (2.6)
3: n_{new}^{dk} = equation (2.7)
4: broadcast(n_{new}^{wk},n_{new}^{dk})
5: θ = add($\theta_{1...p}$)
6: ϕ = add($\phi_{1...p}$)
7: **end procedure**

the end of each Gibbs sampling iteration to synchronize the word-topic counts and to compute the words per topic count matrix ϕ and the topic per document count matrix θ from all the local ones in different partitions P. The Map and Reduce procedures are executed until convergence of the Gibbs algorithm.

Finally, the overall algorithm in Spark uses the defined procedures Map and Reduces, shown in the algorithms 2.3 and 2.4, to perform the computations on the different partitions where we divided our collections of documents (e.g. an input text file "textexample.txt"). The interconnections between the different procedures of the Spark implementation are depicted in Fig.2.3. In the next section, we will evaluate the results of our implementation.

2.3.3 Experiments and results

In this subsection, we document the speed and the likelihood of held-out documents. The purpose of the experiments is to investigate how our distributed version of LDA on Spark performs when executed on small and big data.

We report three evaluations: first the perplexity of the sequential CGS of LDA and the distributed CGS on Spark. Second, after an investigation of the Spark job in the workload, we discuss the percentage of the execution time that benefits from the improvement of the resources and we compute the speedup. Finally, we discuss the scaling behavior of our proposition. The results are reported on three datasets retrieved from the UCI Machine Learning Repository[1].

2.3.3.1 Data Collection, Processing and Environment

Table 2.1: Description of the four datasets used in experiments

	Nips	**KOS**	**NYT**
D	1,500	3430	300,000
V	12,419	6906	102,660
N	2,166,058	467714	99,542,125

We used three datasets from UCI Machine Learning Repository. Table 2.1 summarizes the information about the dataset where D is the number of documents, V is the size of the vocabulary, and N is the number of tokens.

Nips and KOS are considered as small data, we used them to report the experimentation on the perplexity between the sequential CGS and the distributed CGS. The NYT dataset is considered as a big data set. We used the datasets to report the experimentation on the speedup and the scaling behavior.

The downloaded files from UCI are formatted as follows: each line in the file has an ID that corresponds to a document, an ID for a word in the vocabulary V and the

[1]https://archive.ics.uci.edu/ml/datasets/Bag+of+Words

number of occurrence of this work in the document. We needed to transform this format to the following: each line in the input file contains the ID of the document, followed by the ID of the different words in the vocabulary.

For set up the environment, we run all the experiments on a cluster with 10 nodes running Spark 1.5.2. Each node has an Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz (4 cores/8threads), 4 of them have 32GB of RAM and the rest have 16GB of RAM. In the following, we empirically validate our results by analyzing the computed perplexity and the scaling behavior of our sequential and distributed version of LDA.

2.3.3.2 Evaluation with the Perplexity

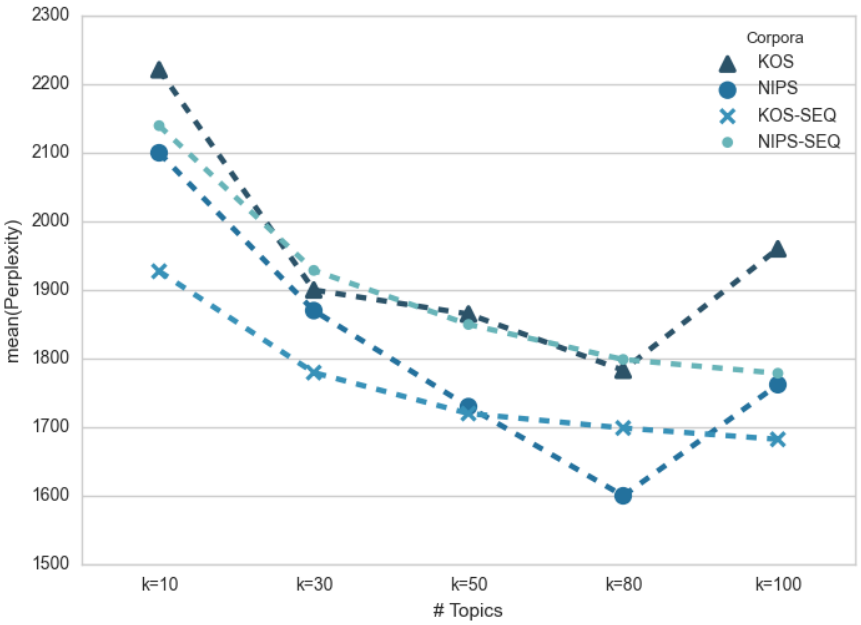


Figure 2.4: Perplexity values comparison between the Spark implementation and the sequential LDA version denoted by SEQ. The perplexity was computed with different number of topics for the KOS and NIPS datasets.

To evaluate the prediction quality of the trained Spark LDA we measure the log-likelihood of a held-out test set given an already trained model [183]. This evaluation is called the test set perplexity and it is defined as

$$exp(-\frac{1}{N_{test}} \log p(x^{test})) \tag{2.8}$$

For LDA, the test set is a set of unseen document W_d and the trained LDA is represented by the distribution of words ϕ. We compute the likelihood $p(x^{test})$ using $S = 10$ samples with 1000 iterations.

$$p(x^{test}) = \prod_{ij} \log \frac{1}{S} \sum_s \sum_k \hat{\theta}^s_{jk} \hat{\phi}^s_{x_{ijk}} \tag{2.9}$$

$$\hat{\theta}^s_{jk} = \frac{\alpha + n^s_{jk}}{K\alpha + \sum_k n^s_{jk}} \quad \hat{\phi}^s_{x_{ijk}} = \frac{\beta + n^s_{wk}}{W\beta + n^s_k}$$

Where $\alpha = 50/K$ and $\beta = 0.1$. We used two small datasets from UCI (i.e. KOS and NIPS) to show that the Spark cluster version of CGS has the same predictive power as

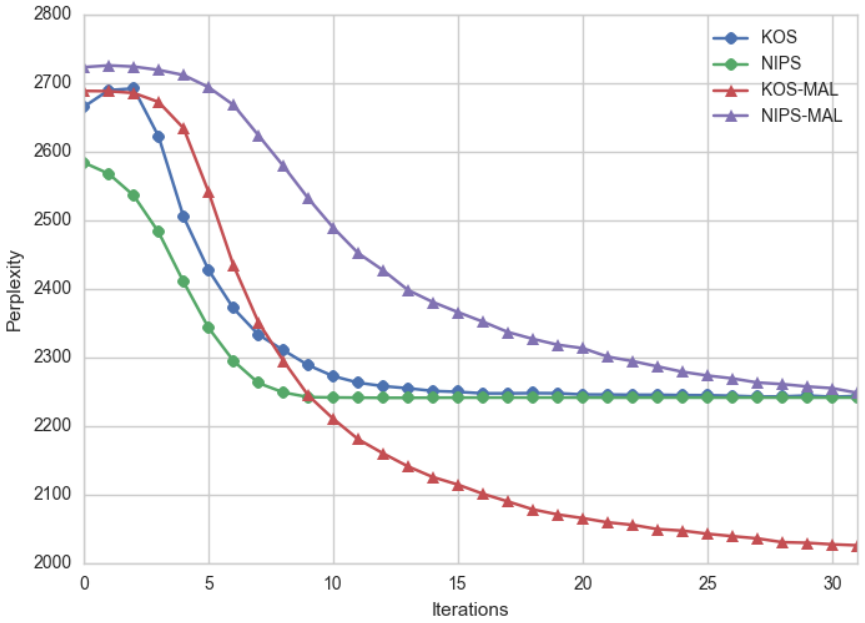


Figure 2.5: Convergence of the perplexity for the KOS and NIPS datasets compared to the convergence of the Mallet (MAL) multi-thread paralleling system.

the sequential version. We computed the test perplexity with different initializations of the parameters.

In this work, we set aside 25% of the documents in each corpus as a test set and train on the remaining 75% of documents. We then compute predictive rank and predictive likelihood. In Fig.2.4, we compare the perplexity values of the sequential LDA version and our distributed LDA version on Spark. For this plot, we set different numbers of topics. We observe that our implementation has the same predictive power as the sequential version. It has even better predictive power for the bigger dataset, in the case the NIPS dataset. For example, when $K = 80$ the perplexity is equal to 1600, compared to 1800 for the sequential version.

For a fixed number of K, we observe in Fig.2.5 that our implementation converges to models having the same predictive power as standard LDA. In this figure, we compare the accuracy of our implementation to the most used framework for topic modeling, called Mallet. Mallet [138] uses multi-thread to distribute the computation of the Gibbs sampling. For this experiment, we used for our implementation 12 cores and 12 threads for the Mallet instance and K = 10.

We note in Fig.2.5, that our implementation represented by the circle points converges before the Mallet instance represented by the triangle points. And this, for the two datasets NIPS and KOS. Whereas for the smallest dataset, i.e. KOS, the Mallet has a better accuracy, for the NIPS our implementation performs better with high convergence rate.

2.3.3.3 Speed Up of the Algorithm

We report in this part the speedup of the collapsed Gibbs sampling (CGS) with a metric which compares the improvement in speed of execution of a task on two similar architectures with different resources. We compute the speedup S based on Amdahl law

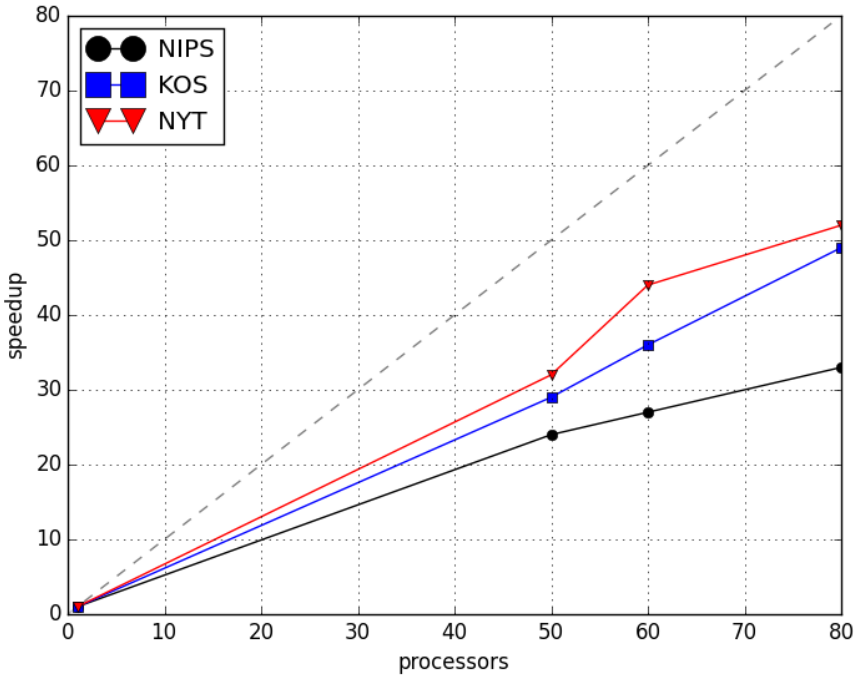


Figure 2.6: Speed up of the spark implementation compared to the number of processors.

[8] presented in the equation below

$$S = \frac{1}{1 - p + \frac{p}{s}} \tag{2.10}$$

where p is the percentage or the portion of the overall task that benefits from the changes in the resources of the architecture and s is the number of resources. Here s is equal to the number of processors. Our speed up experiments is conducted on the NIPS, KOS, and the large NYT dataset.

In our algorithm we have split the Gibbs sampling part into three consecutive parts as shown in the Algorithm 2.4: we broadcast the reduce global counts n^{wk} and n^{dk} then we compute the distribution for θ and ϕ. Finally, we compute the overall speedup by using this equation $s = \frac{1}{\frac{p_1}{s_1} + \frac{p_2}{s_2} + \frac{p_3}{s_3}}$ from Amdahl law.

Figure 2.6 shows a strong correlation between the speedup and the size of the dataset. The speedup approaches the linear case with the NIPS and NYT datasets. For the KOS dataset, we observe a stable speed up from 60 processors. This is due to the small number of words in each partition that have no more effect on the time of the sampling.

2.3.3.4 Scaling Behavior of the Spark Implementation

Once a program is developed with the Spark syntax and worked on a small number of cluster nodes, it can be scaled to an arbitrary number of nodes with no additional development effort. The scalability here is the ability to speed up a task with improving the resources of an architecture of a particular system.

The main point of this part of experimentation is to quantify the scalability of the overall distributed implementation on the Spark framework. To this end, we will use the Universal Scalability Law introduced by Dr. Gunther [84], to analyze the configuration of the system that matches a speedup result. The speedup analyzed in subsection 2.3.3.3

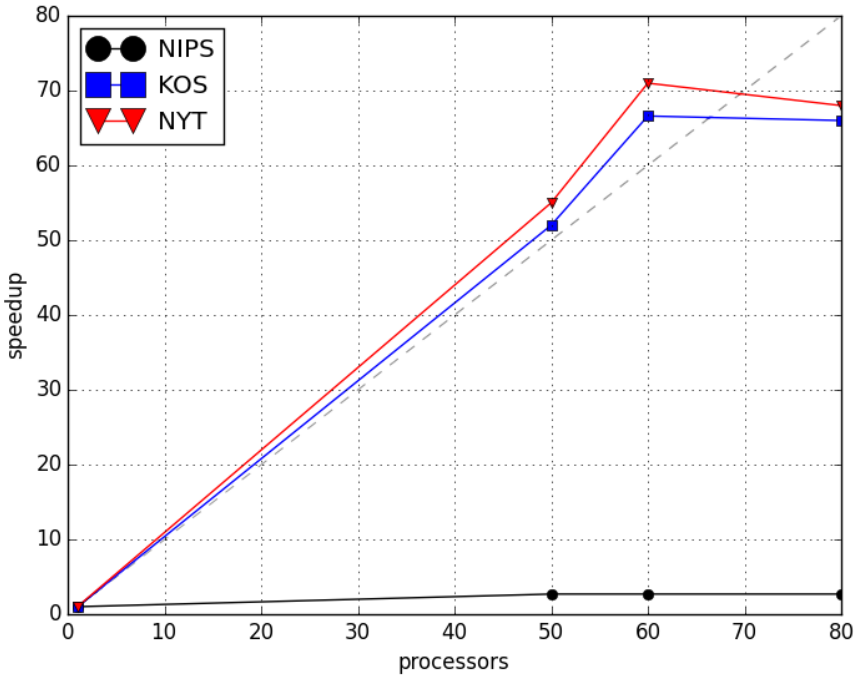


Figure 2.7: Scaling behavior of the Spark implementation computed on NIPS KOS and NYT datasets.

was about the sampling time and not the scalability of the overall implementation. First, we redefine the speedup factor, in equation 2.11.

$$S_p = \frac{T_1}{T_p} \tag{2.11}$$

Where T_1 is the measured time on 1 processor and T_p is the measured time on p processors. The equation 2.11 is a generalization of the Amdahl law and is obtained from the performance model represented in equation 2.12.

$$S_p = \frac{p}{1 + \sigma(p-1) + \kappa p(p-1)} \tag{2.12}$$

where σ represents the degree of contention between different parts of the system and κ represents the delay needed to keep the system coherent. A delay caused by the distribution of the data on the different partitions. Figure 2.7 that describes the scaling behavior of the algorithm, we observe that we have a sublinear speedup for the KOS data. For the NIPS and NYT datasets, we observe a super linear speed up and then from 65 processors the scalability curve cross the linear bound and enter what it is called a payback region. A detailed explanation of this phenomenon was studied on a Hadoop cluster and can be found in this paper [85].

2.3.4 Related works and discussion

To the best of our knowledge, there exist two works similar to our implementation. The first [159] showed good results with the same datasets that we used but the authors did not give any accounts about the global synchronization. For example, in their implementation, the authors did not synchronize C^{DK} alias the document-topic global counts matrix which important to decide when to stop the sampling algorithm (i.e. if

the C^{DK} does not change through the iterations). Moreover, the authors did not report the scaling behavior of their implementation, we found that the paper lacked details of implementation (e.g. the broadcasted counts, the instructions in the map and reduce method), and we could not find their code.

The second [1] gave more details about the code but the author did not show any evaluation of the results and the implementation does not work directly on any text file.

We mention also the work of the Spark community on the LDA. In their implementation, they use online variational inference as a technique for learning the LDA models. The community announced that they are currently working on a Gibbs sampling version to improve the accuracy of the LDA package that they propose.

We cite in the following two of the works that did not implement their proposition in Spark but offered interesting solutions for the communication and memory management.

Wang et al. [184] implemented LDA using Gibbs sampling. To overcome the issue of consistent counts, the authors used message passing between the different partitions on the cluster. This IO/Communication dominates the cost in time of their parallel algorithm which affects the performance of the implementation. In our work, we don't have this problem since we work in memory and we don't have any persistence on the disk.

Ahmed et al. [5] tackled the problem of synchronizing the latent variable of the modeling between different machines. The authors used a distributed memory system to achieve consistent counts and the dual decomposition methods to make local copies of the global variables to obtain consistency in the final counts. The ideas in this work are very close to the Spark framework.

2.3.5 Section conclusion

We proposed in this section a distributed version of the LDA that was implemented in Spark. We reduced the I/O communication and the memory conception by tackling the synchronization of the latent variables of the model. The next step of our work is to improve our implementation to handle the Medline dataset. We also intend to implement a streaming distributed version of LDA where the documents will be processed as they are crawled from the internet in general or social media in particular (e.g Twitter).

2.4 Semi-supervised Latent Dirichlet Allocation

The classification of web pages content is essential to many information retrieval tasks. In this section, we propose a new methodology for a multilayer soft classification. Our approach is based on the connection between the ss-LDA and the Random Forest classifier. We compute with LDA the distribution of topics in each document and use the results to train the Random Forest classifier. The trained classifier is then able to categorize each web document in different layers of the categories hierarchy. We have applied our methodology on a collected data set from *dmoz* [2] and have obtained satisfactory results.

The result of this section has been published in the *Proceeding of the 15th International Conference on Innovations for Community Services, I4CS 2015* [168].

2.4.1 Semi-supervised Latent Dirichlet Allocation

LDA is one of the most used topic models to discover complex semantic structure in the NLP area. However, this methods is unsupervised and therefore does not allow to

[2] http://dmoz-odp.org/

include knowledge to guide the learning process. In this subsection we present a semi-supervised version of LDA based on the works of [161, 132]. The supervision of the process is within two levels.

Algorithm 2.5 The semi-supervised Latent Dirichlet Allocation algorithm

Require: corpus $\mathcal{D} = [d_1, d_2, \ldots, d_M]$
Require: *SetKW*: Set of labels for key words; D_L: Set of labels for document.
procedure SS-LDA($\mathcal{D}$,*SetKW*, D_L)
  loop for each iteration
    loop for each document d
      loop for each word w
        if $w \in SetKW$ **then**
          apply equation (2.14)
          sample z_{ij} from $T_{i,j}$ based on equation (2.13)
        else if $d \in D_L$ **then**
          apply equation (2.15)
          sample z_{ij} form T_i based on equation (2.13)
        else
          sample z_{ij} form T based on equation (2.3)
        end if
  Compute $\phi^{(z)}$
  Compute θ_d
  return $z, \phi^{(z)}, \theta_{\mathcal{D}}$ ▷ Output
end procedure

First, we assign labels at a word level: for each keyword kw_i in set of chosen keywords $SetKW = \{kw_1, kw_2, \ldots, kw_{n_{kv}}\}$, we assign kw_i with a new target topic that is restricted to belong to a set of labels $T_{i,j} = \{T_{i,j_1}, T_{i,j_2}, \ldots, T_{i,j_k}\}$. This step gives us the W_t dictionary where we have for each keywords an associated array of topics $W_t[keyword]$.

Then, at a document level, we label with one or several topics a set of chosen documents. In this step, the new target topic for all words in the labeled document d_i in the set of labeled documents $D_L = \{d_1, d_2, \ldots, d_{M_L}\}$ is restricted to belong to the set of labels $T_i = \{T_{i_1}, T_{i_2}, \ldots, T_{i_k}\}$. This step gives us the D_t dictionary where we have for each labeled document an associated array of topics $D_t[Labeleddoc]$.

For all words in any unlabeled document and unlabeled words, the topic is sampled within the whole topics domain $T = \{T_1, T_2, \ldots, T_K\}$.

Both labeling actions present constraints for the new computed topics. In ss-LDA when we process a labeled document or a labeled word, it applies the sampling only on the topics subset of the labeled entities. Thus, the sampling equation (2.3) is modified and replaced by equation (2.13).

$$P(z_i = j | z_{-i}, w) \propto \frac{n^{w_i}_{-i,j} + \beta}{n^{(\cdot)}_{-i,j} + v\beta} \frac{n^{d_i}_{-i,j} + \alpha}{n^{d_i}_{-i,\cdot} + k\alpha} \tag{2.13}$$

Where v and k are calculated by the equations (2.14) for the labeled word process and the equations (2.15) for the labeled document process :

$$k = |T_{i,j}| = len(W_t[w]); v = |SetKV| \tag{2.14}$$

$$k = |T_i| = len(D_t[d]); v = V \tag{2.15}$$

We note that the sampling with a Gibbs processing has the same behavior applied on complete sets or subsets [103]. Our version is similar to the work of [132] where they used a vector-valued for the hyperparameters α and β. In our algorithm, we used a single value where $\beta = 0.1$ and $\alpha = 50/K$ as in [82] where they have conducted a series of experiments and have explained the chosen values. We implemented the ss-LDA, as described in algorithm 2.5, in python on top of our own implementation of a standard LDA and Gibbs sampling inference.

2.4.2 Application on Multilayer classification of web pages

2.4.2.1 Introduction

The task of text classification (also known as text categorization) is a standard problem addressed in machine learning and statistical NLP [170]. In this task, a text is assigned to one or more predefined class (i.e category) labels through a specific process in which a classifier is built, trained and then applied to label future incoming texts. Several Machine Learning algorithms have been applied to text classification, to name a few: Rocchio's Algorithm, N-Nearest Neighbors, Naive Bayes, Decision tree, Support Vector Machine (SVM), and Neural Network. These algorithms have showed good results [170, 2].

With the increasing popularity of the web, text classification was soon applied to web pages motivated by many information retrieval tasks [137] related to the web content. The following applications as described in [158] show this motivation.

First, web pages classification is essential to the development, expansion, and maintenance of web directories, such as those provided by *Yahoo!* [3] or the Directory Mozilla Dmoz ODP (*dmoz* [4] which used to require a considerable human effort to manage. In the aim of automatically maintain those directory services, this work [156] applied the Naive Bayes for an automatic classification based on the content of the home pages of different websites. The Naive Bayes approach is easy to implement and gives good results [2].

Second, web search engines usually present the search results in a ranked list. Web pages classification gives us the possibility to have different ranked results lists with different categories. This may help the user to get more insights on what he is looking for when he does not have a well-formulated query. Towards this effort, the authors of this work [188] used SVM to cluster the research results into different categories. Although their approach showed good results, their clustering was flat and did not take into account the hierarchy of the categories.

Third, the task of data extraction from the web, also known as crawling, is a critical problem due to the highly heterogeneous data sources. Web pages classification can help with building a focused web crawler rather than performing a full crawl which is usually inefficient. This article [147] presented a good architecture based on XML to extract data from websites. This data extraction requires a solid data validation and error recovery rules. For now, those rules are being manually edited and the authors emphasized the importance of using classification techniques to generate the rules.

In order to propose a new approach to the issues cited above, this section presents a methodology for a multilayer soft classification of web pages textual content. Soft classification refers to a set of probabilities distribution representing the features of each web page and multilayer refers to the different layers in the hierarchy of categories (see fig.2.8). In contrast to the related work, our approach takes into account all the semantic

[3] http://www.yahoo.com
[4] http://dmoz-odp.org/

structures of the text in the web page and classify it accordingly. Our methodology allowed us to obtain good accuracy results on a Data collection from *dmoz*.

2.4.2.2 Related work

Random Forests have been used in different areas of research [54] including image classification, network intrusion detection, fraud detection, biological activity categorization, etc. Nevertheless, few works have been dedicated to the categorization of web text content using random forests. We briefly review them in the following.

In [101] the authors employed the Random Forest to classify web documents into a hierarchy of directories. The keywords were extracted from the documents and were used as attributes to learn the random trees. The authors already used an implemented version of Random Forest in WEKA 3.5.6 software developed by the University of Waikato and showed that Random Forest performed better than other well-known learning methods such as Naive Bayes or the multinomial regression model.

This work [129] introduced a news article classification framework based on Random Forests that are trained on multimodal features (i.e. textual and visual features). To extract the textual feature the authors used an N-gram statistics [137]. Although their multimodal approach only gave a slight improvement in the results compared to the random forests trained on the textual feature, this article proved the capacity of Random Forest to classify texts based on different types of attributes.

Towards a common framework for text categorization based on the Random Forest, the authors of this work [186] presented an improvement in the used methods for building the random forest. A new feature extraction method and a tree selection were developed and synergistically served for making random forest framework well suited for categorizing text. The authors compared their framework with classical machine learning methods used for text classification and showed that their algorithm effectively reduces the upper bound of the generalization error and improve classification performance.

The main issue with learning the random forest classifier is the task of features extraction from the web documents. The work of [129] used the N-grams method, and the work of [101] used a bag of word approach with a simple document frequency selection. In our work, we chose to use LDA because it showed better information rates than the both approaches [26, 181]. It also allows us to perform a soft classification with a set of probability distributions over the possible classes. But, LDA is a complete unsupervised method that does not allow any control on the computed features. Thus, we decided to use a semi-supervised version based on the works of [161, 132].

2.4.2.3 Random Forests

Random Forests belong to the machine learning *ensemble methods* [162]. The term 'random forests' originally comes from [89] wherein 1995 Ho et al. proposed a model that aggregates a set of decision tree built upon a random subset. However, this approach encounter an issue in prediction due to *overfitting* (know also as *overlearning*). This problem was approached by Kleinberg in 1996 [102]. In the same year, Breiman [34] proposed the Bagging technique which tends to resolve the problem of *overlearning*. Bagging improves the estimate or prediction itself by averaging this prediction over a collection of bootstrap (random training set) samples. Combing the ideas of decision trees and *ensemble methods* [162], Breiman in 2001 [35], gave use to decision forests, that is, sets of randomly trained decision trees. Random Forest improve bagging by reducing the correlation between the sampled trees by different sources of randomness.

In the random forest, there are two different sources of randomness within the processing of building the trees. First, the bootstrap is a technique to build a random set that gives the best accuracy of a parameter estimate or prediction. We Draw a bootstrap (bagging: averaging estimators) sample $\mathbf{Z}^*$ of size N from the training data. Second, the random selection of the attributes to split each node where we draw uniformly at random with the replacement N data. (Each tree will have a random different set). The process of building a random forest is described in algorithm 2.6.

Algorithm 2.6 The random forest algorithm

Require: : Data D of N points.
1: **procedure** RANDOMFORESTTRAIN(D)
2: **loop** for each tree b ∈ B trees
3: Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
4: **repeat**
5: Select m variables at random form the p variables.
6: Pick the best variable/split-point among the m.
7: Split the node into two daughter nodes.
8: **until** the minimum node size n_{min} is reached.
9: **return** the set of trees $\{T_b\}_1^B$ ▷ Output
10: **end procedure**

2.4.2.4 Our Contribution

We propose a methodology of classifying web pages documents. The classification that we perform is a soft classification, where we take into account, within the process, the distribution of the words (topics) in each document and the distribution of the topics in each document. Thus, we do not classify according to one single word, or a set of keywords or a single topic.

Our methodology takes into account the whole hierarchy of the categories that the web documents belong to, allowing to have a multi-layer classification (see Fig.2.8). First, we guide the learning process of LDA by a ss-LDA to capture the low-level categories of each web documents (Layer 3 in Fig.2.8). This first step gives us a distribution of the different low-level categories for each web document (e.g. Storage, Open Source, Educational, etc ..). Then, we build a spreadsheet with the low-level categories as features and the higher level categories as values that we want to predict. Finally, we run the Random Forest Classifier on a train dataset (See the *MultiLayerClassify* procedure in algorithm 2.7). This algorithm describes our methodology for classifying web documents on a multiple layers begin from a low level categories and ascends to the high level categories. This classification takes into account the distribution of the topics in the web documents and the distribution of the words in each topic (i.e soft classification).

Algorithm 2.7 MultiLayer Classify algorithm

1: **procedure** MULTILAYERCLASSIFY(WebDocs)
2: DocsWordFile ← PREPROCESS(WebDocs)
3: θ_d ← ss-LDA(DocsWordFile)
4: csvFile ← buildSheet(θ_d)
5: clf ← RANDOMFORESTRAIN(csvFile)
6: predictionOfCategories ← predict(clf)
7: **end procedure**

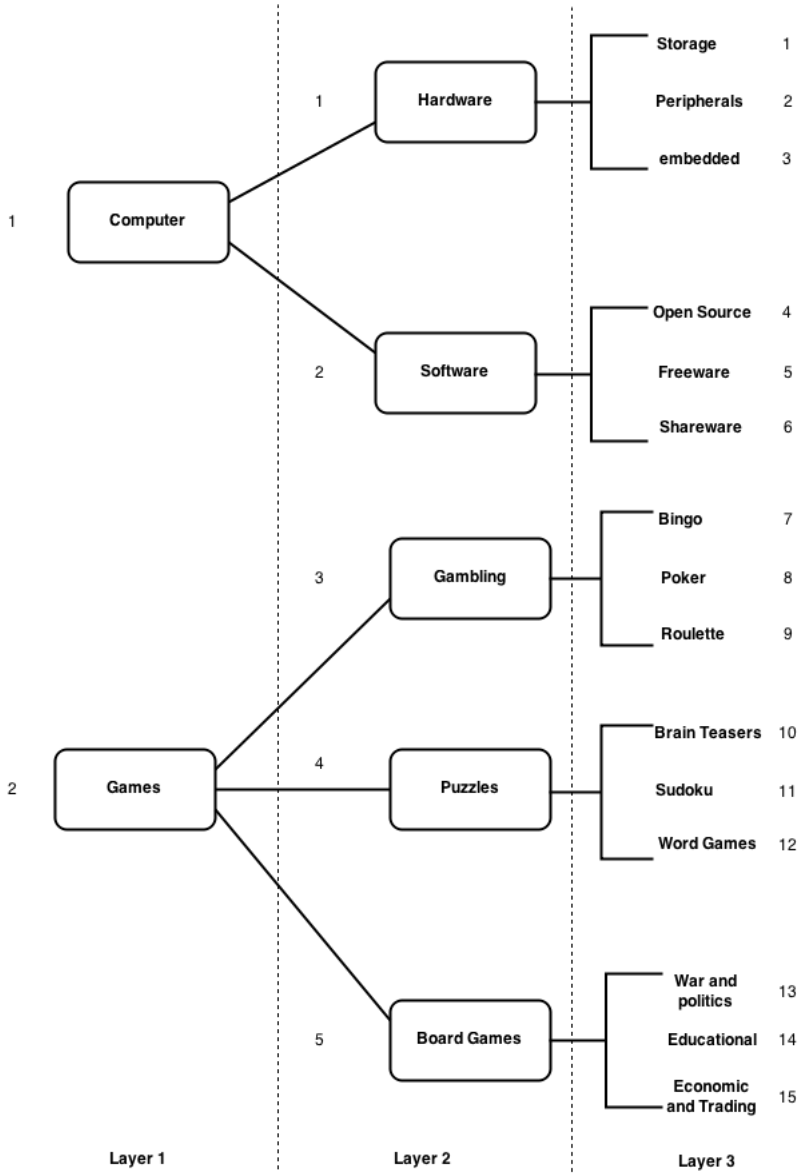


Figure 2.8: Three layers of categories: we have 3 layers, in the first layer we have 2 main category Computer and Games, as we go deep in the layers we have more categories.

2.4.2.5 Experiments and results

Data collection and preprocessing We have collected text content from the home pages of 60 websites listed in the *dmoz* [5] web directories. As described in Fig.2.8, we have 15 low-level categories (Layer 3), 5 mid-level categories (Layer 2) and 2 high-level categories (Layer 1). For each low-level category, we have 4 documents. Before running the *ss-LDA* and building the spreadsheet for Random Forest, we first took the text from the home page of each website and processed it to get good insights. The data processing comes in the following steps and it is described in algorithm 2.8. First, the home pages usually contain a lot of HTML entities (e.g < > &amp), the first step is to get rid of these entities by using the python library *HTMLParser* which convert these entities to standard HTML tags. Second, decoding the data in a standard encoding format e.g. UTF-8 encoding is widely accepted. To avoid any disambiguation in the text it is better to maintain a proper structure of the text with a free grammar context. Therefore, the

[5] http://dmoz-odp.org/

Algorithm 2.8 Data collection and pre-processing for the ss-LDA input

Require: $WebDocs$ ▷ List of the Web Documents

procedure PREPROCESS($WebDocs$)
  for $user \in user_{list}$ **do**
    $WebText \leftarrow$ Decode(UTF8)
    $WebText \leftarrow$ HtmlParse()
    $WebText \leftarrow$ UrlRemove()
    $WebText \leftarrow$ AppoReplace()
    $WebText \leftarrow$ StopwordRemove()
    $WebText \leftarrow$ HExpRemove()
    return CreateDocsWordsFile($WebText$)
  end for
end procedure

third step consist of converting all the apostrophes into standard lexicons e.g {('s: is); ('re : are); ('ll : will); ...}.

The *ss-LDA* analyzes the data at a word level. We then need to remove the commonly occurring words by creating a list of the called stop-words. In the remaining steps, we remove punctuations, common words for human expressions and replace the slang words with their actual meaning. We precise that we have not used any stemming techniques because we wanted to keep the different variations of a particular word that could be on different topics. As for the *ss-LDA* we have implemented our own version with *Python* as described in algorithm 2.8. In the following subsection, we present our obtained results.

Results We used the Random Forest implementation of the *Sklearn* package available within the *Scikit-learn* library[6]. The computed spread sheet with *ss-LDA* was divided into two files : 75% for training and 25% for testing. We have 15 attributes, corresponding to the 15 lower level categories (see Fig.2.8, Layer 3) and 2 target classes (Layer 1 and 2 in Fig.2.8).

In our analysis, we compare two strategies of web pages classification. The first is comparable to a keyword classification [101] where we don't take into account the results of the classification of the layer above (i.e Layer 1 in Table 2.2 and Fig.2.9.). We called this strategy A. The second, is a multi-layer classification where the results of the classification of the above layer is taken as an input for the classification of the second layer's web pages. We called this strategy B. We also evaluate in our analysis the effects of the change in the number of trees and put the results in Table 2.2 where we averaged the results of the 10 times executions of the Random Forest classifier.

In the Layer 1, Table 2.2, we obtained a classification rate of 93.93% with a null standard deviation. This classification of the web pages in the highest layer is obtained from the results of the classification in layer 3, where we computed the topics with the *ss-LDA*. For the strategy A, Layer 2 in Table 2.2, we obtained a lowest classification rate of 66.67% and a highest classification rate of 73.33%. For the strategy B, we obtained a lowest classification rate of 77.78% and a highest classification rate of 100%. Thus, as mentioned in the conclusion of the work of [101], the more topics there are, the less accuracy can be obtained. Our methodology offers the use of the strategy B that leads to a better accuracy within the high-level description categories (e.g Layer 1).

[6] http://scikit-learn.org/stable/

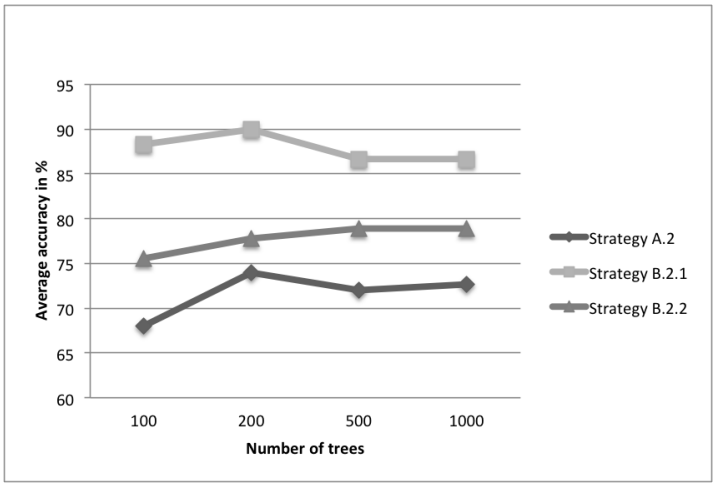


Figure 2.9: The average accuracy of the two strategies: this graph shows the average accuracy of the strategy A and B of our classification methodology

2.4.3 Section Conclusion

We presented in this section a novel approach for classifying the web pages content. The Multi-Layer classification is a connection between the ss-LDA and the Random Forest Classifier. We obtained a classification rate of 93,33% for the top layer and we improved the results of the classification rate for the lowest layers with a minimum variation.

2.5 Conclusion

We have presented in this chapter the LDA, a well-known probabilistic topic model and the two extended of this model. The first we proposed a distributed version of the LDA, AD-LDA, that was implemented in Spark. We reduced the I/O communication and the memory conception by tackling the synchronization of the latent variables of the model. The second, we presented the ss-LDA, an LDA version for the learning process, and apply this model in classifying web pages and obtained a good classification rate.

Table 2.2: The classification rates of our methodology with different parameters values.

Layer 1				
	100 trees	**200 trees**	**500 trees**	1000 trees
Min	93.3%	93.3%	93.3%	93.3%
Max	100%	100%	100%	93.3%
Average	95.9%	95.9%	95.9%	93.3%
Standard deviation	3.4	3.2	2.8	0

Layer 2 (strategy A)				
	100 trees	**200 trees**	**500 trees**	**1000 trees**
Min	53.3%	66.6%	66.6%	66.6%
Max	80%	80%	73,3%	73,3%
Average	68%	74%	71,9%	72,6%
Standard deviation	6,8	5,8	2,8	2,1

Layer 2.1 (strategy B)				
	100 trees	**200 trees**	**500 trees**	**1000 trees**
Min	50%	83.3%	83.3%	83.3%
Max	100%	100%	100%	100%
Average	88.3%	89.9%	86.6%	86.6%
Standard deviation	15.8	8.6	7	7

Layer 2.2 (strategy B)				
	100 trees	**200 trees**	**500 trees**	**1000 trees**
Min	66.6%	66.6%	66.6%	77.7%
Max	88.8 %	100%	88.8%	88.8%
Average	75.5%	77,7%	78,8%	78,8%
Standard deviation	7	10.4	8.1	3.5

Chapter 3

Pretopology and Topic Model for Document Clustering

3.1 Introduction

Classifying a set of documents is a standard problem addressed in machine learning and statistical natural language processing [137]. Text-based classification (also known as text categorization) examines the computer-readable ASCII text and investigates linguistic properties to categorize the text. When considered as a machine learning problem, it is also called statistical NLP [137]. In this task, a text is assigned to one or more predefined class labels (i.e category) through a specific process in which a classifier is built, trained on a set of features and then applied to label future incoming texts. Given the labels, the task is performed within the supervised learning framework. Several Machine Learning algorithms have been applied to text classification (see [2] for a survey): Rocchio's Algorithm, N-Nearest Neighbors, Naive Bayes, Decision tree, Support Vector Machine (SVM), etc.

In the absence of predefined labels, the task is referred as a clustering task and is performed within the unsupervised learning framework. Clustering is one of the most popular data mining algorithms and have extensively studied in the context of text. The clustering is the task of finding groups of similar documents in a collection of documents. The similarity is computed by using a similarity function. There are many clustering algorithms that can be used in the context of text data. Text document can be represented as a binary vector, i.e. considering the presence or absence of word in the document. Or we can use more refined representations which involves weighting methods such as Tf-idf (term-frequency (tf) and inverse document frequency (idf)) matrix. In Tf-idf, text-based features are typically extracted from the so-called word space model that uses distributional statistics to generate high-dimensional vector spaces. Each document is represented as a vector of word occurrences. This gives the model its name, the *Vector Space Model* (VSM) [166].

There are two typical categories of clustering algorithms, the partitioning and the agglomerative. K-means and the hierarchical clustering are the representatives of these two categories, respectively. There are many comparisons between k-means and hierarchical clustering [97]. In our work, we choose the k-means algorithm for document clustering since it is much faster.

For document clustering using *VSM* and k-means algorithm, we face up to four challenges:

Q3.1: How to reduce the high-dimensional matrix of document representation using VSM?

Q3.2: How to choose a "good" distance measure to get the most accurate clusters?

Q3.3: How to choose the number of cluster for input of k-means?

Q3.4: How to work with multi-criteria clustering?

For the first challenge Q3.1, one of the possible solutions is to represent the text as a set of topics and use the topics as an input for a clustering algorithm. This is the idea of topic modeling. Topic modeling is one of the most popular probabilistic clustering algorithms which has gained increasing attention recently. The main idea of topic modeling [26, 93, 82, 176] is to create a probabilistic generative model for the corpus of text documents. In topic models, documents are a mixture of topics, where a topic is a probability distribution over words. The three main topic models are Latent Semantic Analysis (LSA) [109] which uses the Singular-value decomposition methods to decompose high-dimensional sparse matrix; Probabilistic Latent Semantic Analysis (pLSA) [93], a probabilistic model that treats the data as a set of observations coming from a generative model; *Latent Dirichlet Allocation* (LDA) [26] is a Bayesian extension of pLSA.

Therefore, in our work, we first reduce the dimensionality by decomposing the document matrix into latent components using the LDA [26] method. Each document is represented by a probability distribution of topics and each topic is characterized by a probability distribution over a finite vocabulary of words. We use the probability distribution of topics as the input for k-means clustering. This approach called *LDA+k-means* was proposed by [185] and [46]. We note that [185] proposed LDA+k-means but only used Euclidean distance.

For the second challenge Q3.2, we investigate the efficiency of eight distance measures represented for eight distance families categorized by [52] for *LDA+k-means*. These measures are based on two approaches: i) Vector-based measurement (VBM) with Euclidean distance, Sørensen distance, Tanimoto distance, Cosine distance and ii) Probabilistic-based measurement (PBM) with Bhattacharyya distance, Probabilistic Symmetric χ^2 divergence, Jensen-Shannon divergence, Taneja divergence.

For the two last challenge Q3.3, Q3.4, we propose to use pretopology approach. The Pretopology theory [23] offers a framework to work with categorical data, to establish a multi-criteria distance for measuring the similarity between the documents and to build a process to structure the space [111] and infer the number of clusters for k-means. We can then tackle the problem of clustering a set of documents by defining a family of binary relationships on the topic-based contents of the documents. The documents are not only grouped together using a measure of similarity but also using the pseudo-closure function built from a family of binary relationships between the different hidden semantic contents (i.e topics). We present this approach in a method that we named the Method of Clustering Documents using Pretopology and Topic Modeling (MCPTM). MCPTM organizes a set of unstructured entities in a number of clusters based on multiple relationships between two entities. Our method discovers the topics expressed by the documents, tracks changes step by step over time, expresses similarity based on multiple criteria and provides both quantitative and qualitative measures for the analysis of the document.

This chapter is organized as follows. Firstly, we describe in section 2 the VSM for document clustering. Dimensionality reduction techniques will be presented in section 3. Section 4 presents the method for document clustering using Topic modeling with two approaches: *LDA+Naive*, and *LDA+k-means*. We then examine the role of probabilistic distance for *LDA+k-means* in section 5. Section 6 presents the pretopological approach for multi-criteria document clustering. Then it is followed by conclusions in section 7.

3.2 Vector Space Model

The Vector Space Model (VSM) is the basic model for document clustering, upon which many modified models are based. We briefly review a few essential topics to provide a sufficient background for understanding document clustering.

3.2.1 Tf-idf Matrix

In VSM, each document, d, is considered to be a vector in the term-space, represented in its simplest form by the *term-frequency* (TF) vector

$$d_{tf} = (tf_1, tf_2, \ldots, tf_N) \tag{3.1}$$

where tf_i is the frequency of the i^{th} word in the document. This gives the model its name, the vector space model (VSM) [166].

A widely used refinement to the vector space model is to weight each term based on its *inverse document frequency* (IDF) in the document collection. The motivation behind this weighting is that terms appearing frequently in many documents have limited discrimination power and thus need to be de-emphasized. This is commonly done [165] by multiplying the frequency of the ith term by $log(N/df_i)$, where df_i is the number of documents that contain the ith term (i.e., document frequency). This leads to the *tf-idf* representation of the document,

$$d_{tf-idf} = (tf - idf_1, tf - idf_2, \ldots, tf - idf_N) \tag{3.2}$$

Put these tf-idf vectors together, we get a *tf-idf* matrix

3.2.2 Similarity measures in vector space

Before clustering, a similarity/distance measure must be determined. The measure reflects the degree of closeness or separation of the target objects and should correspond to the characteristics that are believed to distinguish the clusters embedded in the data. In many cases, these characteristics are dependent on the data or the problem context at hand, and there is no measure that is universally best for all kinds of clustering problems. In the following, we present four distances between two vectors that are often used in document clustering: Euclidean distance, Cosine distance, Sørensen distance, Tanimoto distance. From now on, we denote $A = (a_1, a_2, \ldots, a_k)$ and $B = (b_1, b_2, \ldots, b_k)$ be two vectors with k dimensions.

3.2.2.1 Euclidean distance

Euclidean distance, also known as the L^2 norm in the *Minkowski family* of distance metrics [52], is a standard metric for geometrical problems. It is the ordinary distance between two points and can be easily measured with a ruler in two- or three-dimensional space. Euclidean distance is widely used in clustering problems, including clustering text. It is a true metric. It is also the default distance measure used with the k-means algorithm. The Euclidean distance between two points, A and B, with k dimensions is calculated as:

$$d_{Euc}(A, B) = \sqrt{\sum_{i=1}^{k} |a_i - b_i|^2} \tag{3.3}$$

3.2.2.2 Cosine distance

The cosine similarity represented for *Inner product family* [52] is one of the most used measures to compute similarity between two documents in the vector space. The inner product of two vectors $d_{IP}(A,B) = \langle A,B\rangle = \sum_{i=1}^{k} a_i b_i$ yields a scalar and is sometimes called the scalar product or dot product. The cosine similarity is the normalized inner product and called the *cosine* coefficient because it measures the angle between two vectors and thus often called the *angular metric*. The cosine distance between two points is one minus the *cosine* of the included angle between points (treated as vectors). Given vectors A and B, the cosine distance between A and B is defined as

$$d_{Cos}(A,B) = 1 - Sim_{Cos}(A,B) = 1 - \frac{\sum_{i=1}^{k} a_i b_i}{\sqrt{\sum_{i=1}^{k} {a_i}^2}\sqrt{\sum_{i=1}^{k} {b_i}^2}}. \tag{3.4}$$

3.2.2.3 Sørensen distance

Sørensen distance [177] is one of the distances in L_1 *family* [52], more precisely the absolute difference. It is widely used in ecology [130]. Given vectors A and B, the Sørensen distance between A and B is defined as:

$$d_{Sor}(A,B) = \frac{\sum_{i=1}^{k} |a_i - b_i|}{\sum_{i=1}^{k} (a_i + b_i)} \tag{3.5}$$

3.2.2.4 Tanimoto distance

Tanimoto distance [64] is a distance in *intersection family* distances [52]. Given vectors A and B, the Tanimoto distance between A and B is defined as:

$$d_{Tani}(A,B) = \frac{\sum_{i=1}^{k} (max(a_i,b_i) - min(a_i,b_i))}{\sum_{i=1}^{k} max(a_i,b_i)}. \tag{3.6}$$

3.2.3 K-means algorithm

K-means which proposed by Forgy [79] is one of the most popular clustering algorithms. It provides a simple and easy way to classify objects in k groups fixed a priori. The basic idea is to define k centroids and then assign objects to the nearest centroid. A loop has been generated. In each step, we need to re-calculate k new centroids and re-assign objects until no more changes are done. The algorithm works as follows:

1. Selecting k initial objects called centroids of the k clusters.
2. Assigning each object to the cluster that has the closest centroid.
3. Computing the new centroid of each cluster.
4. Repeat step 2 and 3 until the objects in any cluster do no longer change.

Please refer the figure 3.1 for a diagrammatic view of the K-Means algorithm and the figure 3.2 for an example of k-means algorithm.

3.2.4 Document clustering with Vector Space Model

After representing the documents by VSM matrix, we can use this matrix as an input for standard clustering algorithms. There are two typical categories of clustering algorithms, the partitioning and the agglomerative. K-means and the hierarchical clustering are

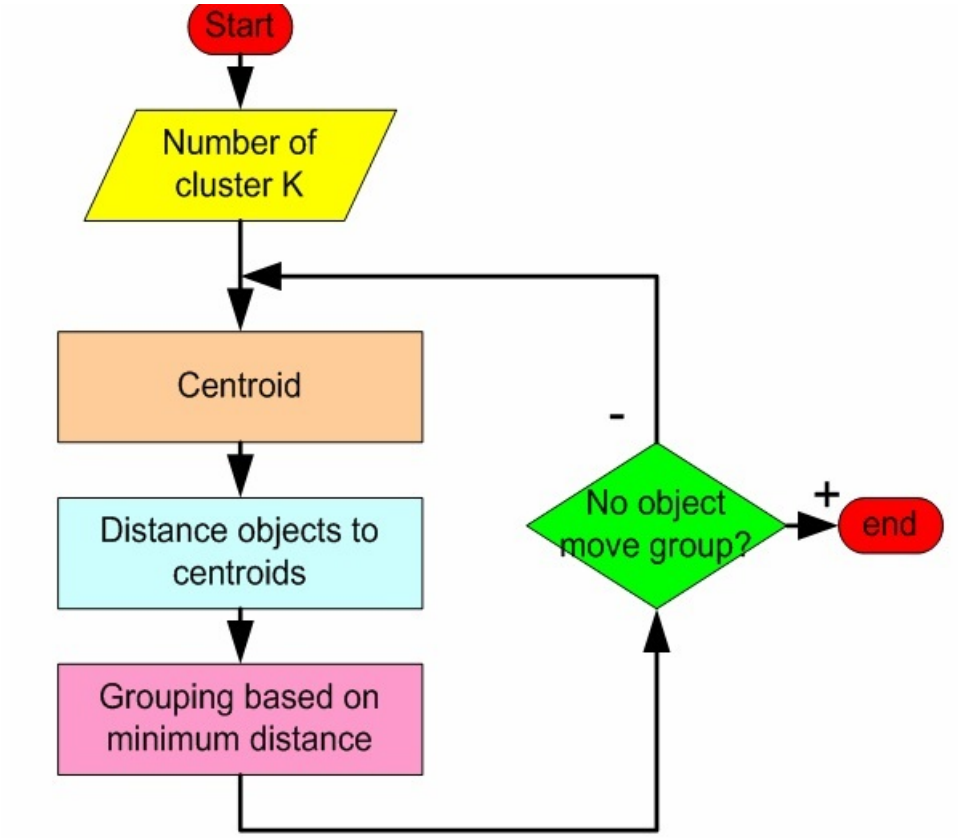


Figure 3.1: K-Means Algorithm.

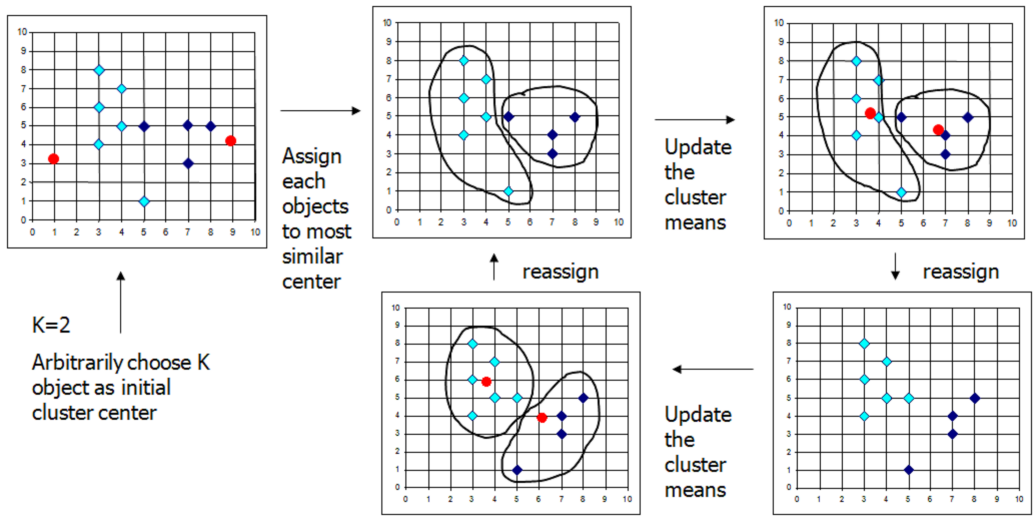


Figure 3.2: K-Means Algorithm Example.

the representatives of these two categories, respectively. There are many comparisons between k-means and hierarchical clustering. But our consideration is speed since we are going to apply clustering algorithms on big social network data, which is always of GB or TB size. And the hierarchical clustering is extremely computational expansive as the size of data increases, since it needs to compute the $D \times D$ similarity matrix, and merges small clusters each time using certain link functions. In contrast, k-means is much faster. It is an iterative algorithm, which updates the cluster centroids (with

normalization) each iteration and re-allocates each document to its nearest centroid. A comparison of k-means and hierarchical clustering algorithms can be found in [97].

For this reason, in our work, we choose the k-means algorithm for document clustering. Figure 3.3 shows a schema of document clustering with VSM using k-means algorithm.

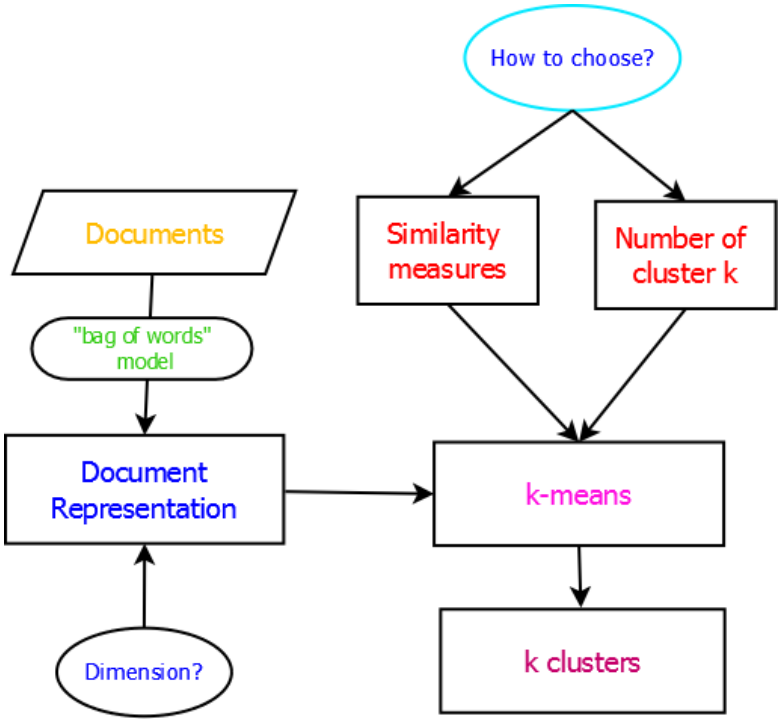


Figure 3.3: Vector Space Model Approach for Document Clustering using k-means.

In VSM, each document is represented as a vector, i.e., a point in the Euclidean space or a Cartesian coordinate system. Hence, numerous geometrical distances can be applied to compare two vectors. Subsection 3.2.2 presented four distances (Euclidean, Cosine, Sørensen, Tanimoto) that have been used in document clustering. For other distances, please refer the works of [63, 52]. For comparing the effectiveness of different distance measures in VSM, please refer the works of [95, 128, 160] (see Table 3.1).

For document clustering by using the k-means algorithm with the full space VSM, we often face up to the problem of high computational cost, which makes it less appealing when dealing with a large collection of documents. The problem is the curse of dimensionality: V is large. Generally, there are more than thousands of words in a vocabulary, which makes the term space high dimensional. Hence, various dimensionality reduction techniques have been developed to make improvements above this approach.

3.3 Dimensionality Reduction Techniques

As to our knowledge, there are two main categories of dimensionality reduction techniques. One is to first write down the full VSM matrix, and then try to reduce the dimension of the term space by numerical linear algebra methods. Another is to try using a different representation of a document (not as a vector in a space) from the very beginning.

3.3.1 Feature Selection Method - LSA

The Latent Semantic Analysis [109] method is based on the singular value decomposition (SVD) technique in numerical linear algebra. It can capture the most variance by combinations of original rows/columns, whose number is much less than the original matrix. In addition, the combinations of rows (term) always show certain semantic relations among these terms, and combinations of columns (document) indicate certain clusters. After SVD, the k-means algorithm is run on this reduced matrix, which is much faster than on the original full matrix. But the problem with this approach is that the complexity of SVD is $O(D^3)$, so as the number of documents increases, the computation of SVD will be very expansive, and therefore the LSA approach is not suitable for large datasets.

3.3.2 Alternative Document Representations

There are other document representations and similarity measures besides VSM and cosine similarity, such as Tensor Space Model (TSM) and a similarity based on shared nearest neighbors (SNN), etc. These alternatives may be effective in some special cases, but not in general.

One significant step in this area is the introduction of the concept of 'latent topics'. It is similar to the latent class label in the mixture of Gaussian models. This concept has led to a series of generative models, which specify the joint distribution of the data and the hidden parameters as well. Among these models are the Probabilistic Latent Semantic Analysis (pLSA) [93], the Mixture of Dirichlet Compound Multinomial (DCM), the LDA [26]. The pLSA has a severe overfitting problem, since the number of parameters estimated in the model is linear in D, the number of documents. On the other hand, LDA specifies three levels of parameters: corpus level, document level, and word level. And the total number of parameters is fixed: $K + K \times V$, where K is the number of latent topics regarded as given. This multilayer model seems more complicated than others, but it turns out to be very powerful when modeling multiple-topic documents.

Therefore, in our work, we choose LDA [26, 25] for dimensionality reduction.

3.4 Document clustering using Topic modeling

Generally, there are two ways of using topic models for document clustering.

3.4.1 LDA + naive

The first approach that we call *LDA+naive* uses topic models more directly. The basic assumption is that each topic corresponds to a cluster, so the number of topics in topic models matches the number of clusters. After estimating the parameters, the documents are clustered into the topic with the highest probability:

$$x = argmax_j \theta_j.$$

3.4.2 Combining LDA and k-means

The second approach uses a topic model such as LDA to map the original high-dimensional representation of documents (word features) to a low-dimensional representation (topic features) and then applies a standard clustering algorithm like k-means in the new feature space. In our work, we used LDA document-topic distributions θ extracted from LDA as the input for k-means clustering algorithms. We call this approach is *LDA+k-means*.

Here we focus on the second approach of clustering because it allows us to examine the effectiveness of dimension reduction using topic models for the document clustering. After reducing the dimension by LDA, the input for k-means is the probability distribution, therefore it is necessary to compare the efficiency of clustering using *Probability-based measurements* (PBM) and *Vector-based measurements* (VBM). The next section will examinate the role of Probabilistic distance for document clustering using *LDA+k-means*.

3.5 LDA and k-means: Role of Probabilistic distances

This work evaluates through an empirical study eight different distance measures used on the *LDA+k-means* model. We performed our analysis on two miscellaneous datasets that are commonly used. Our experimental results indicate that the probabilistic-based distance measures are better than the vector-based distance measures including Euclidian when it comes to cluster a set of documents in the topic space. Moreover, we investigate the implication of the number of topics and show that k-means combined to the results of the LDA model allows us to have better results than the *LDA+Naive* and VSM.

The result of this section has been published in the *Proceedings of ACIIDS 2017, Part I. Lecture Notes in Computer Science 10191*, 2017, ISBN 978-3-319-54471-7, pages: 248-257 [44].

3.5.1 Introduction

Clustering a set of documents is a standard problem addressed in data mining, machine learning, and statistical natural language processing. Document clustering is an important information processing method which can automatically organize a large number of documents into a small number of meaningful clusters and find latent structure in unlabeled document collections. Document clustering is often used in intelligence analysis field to resolve the issue of information overload.

K-means algorithm is one of the partitioned-based clustering algorithms, which has been popularly used in such areas as information retrieval [137] and personalized recommendation. To apply k-means for document clustering, documents are quantified as a vector in which each component indicates the value of its corresponding feature in the document. Documents are usually represented with a bag-of-words (BOW) model or vector space model (VSM) where each document corresponds to as high-dimensional and sparse vectors. In such situations, if we use k-means in document clustering, we must solve two problems: one is how to reduce the document dimensionality and capture more semantic of documents as efficient as possible; another is how to choose a "good" distance or similarity measure to quantify how different two given documents are.

For the first problem, we will use LDA [26] to model documents. In fact, document clustering and LDA are highly correlated and LDA can mutually benefit document clustering. As one of the basic topic models, LDA can discover latent semantic structure in a document corpus. The latent semantic structure is able to put words with similar semantics into the same group. The topics can capture more semantic information than raw term features. So, through LDA model, each document will be represented by a topic space to reduce the noise in similarity measure and also reduce the document dimensionality. Having the distribution of topics in documents from LDA, we can use it as the input for k-means clustering. This approach, denoted *LDA+k-means* proposed by [185, 46] is integrated LDA and k-means.

For the second problem, since the input for k-means is the probability distribution,

it is necessary to compare the efficiency of clustering using PBM and VBM. Although researchers have compared the effectiveness of a number of measures described in table 3.1, they just did it with vector space model (VSM). Our experiments extended their work by examining the *LDA+k-means* model with eight distances or similarity measures from eight distance families categorized by [52]. These measures are based on two approaches: i) Vector-based approach with Euclidean distance, Sørensen distance, Tanimoto distance, Cosine distance and ii) Probabilistic-based approach with Bhattacharyya distance, Probabilistic Symmetric χ^2 divergence, Jensen-Shannon divergence, Taneja divergence.

Table 3.1: Summary of Literature Survey

Cite/Year	Datatsets	Methods	Measures
[95]/2008	20NG, classic,hitech re0, tr41, wap, webkb	k-means with VSM	Euclidean, Cosine, Jaccard, Person correlation, KL
[128]/2014	WebKB, Reuters-8, RCV1	k-means with VSM	Euclidean, Cosine, SMTP
[134]/2016	WebKB, Reuters-8, RCV1	k-means with VSM	Euclidean, Cosine, SMTP
[160]/2013	20NG, Reuters, WebKB, Classic, OSHUMED	topic map	K-L Divergence
[185]/2013	20NG, Reuters	k-means+MGCTM	Euclidean

In order to come up with a sound conclusion, we have performed an empirical evaluation of the eight distance measures according to a labeled clustering. We compared the clusters with the two evaluation criteria: Adjusted Rand Index (ARI) [96] and Adjusted Mutual Information (AMI) [180]. We used two common datasets in the NLP community: the 20NewsGroup dataset contains newsgroup posts and the WebKB contains texts extracted from web pages.

Our experiments can be compared to the work of [95, 185, 134]. The key differences are the following: in comparison with the VBM we conducted our experiments with a PBM, we show that in the case of *LDA+k-means* where the input is a probability distribution the use of PBM leads to better results. Then, our results show that the Euclidean distance may not be suitable for this kind of application. Finally, by evaluating the results of the VBM and PBM with ARI and AMI criteria we have investigated the implication of the number of topics in the clustering processing.

This section is organized as follows. The two next subsections describe the similarity measures in probabilistic spaces and evaluation indexes used in the experiments. We then explain the experiment, discuss the results and also conclude our work.

3.5.2 Similarity measures in probabilistic spaces

Since LDA represent documents as probability distributions, We need to consider the "good" way to choose a distance or similarity measure for comparing two probability distribution. There are two approaches in probability density function (pdf) distance/similarity measures: vector and probabilistic. For probabilistic approach, computing the distance between two pdf's can be regarded as the same as computing the Bayes (or minimum misclassification) probability. This is equivalent to measuring the overlap between two pdfs as the distance. For this approach, f-divergence $D_f(P||Q)$ [55, 145, 7] is often used to measure the difference between two probability distributions P and Q. It helps the intuition to think of the divergence as an average, weighted by the function f, of the odds ratio given by P and Q.

Definition 3.1. *Let* $P = \{p_i|i = 1, 2, \ldots, d\}$ *and* $Q = \{q_i|i = 1, 2, \ldots, d\}$ *be two probability distributions over a space* Ω *such as* P *is absolutely continuous with respect*

to Q. Then, for a convex function f such that $f(1) = 0$, the f-divergence of Q from P is defined as:

$$D_f(P||Q) = \sum_{i=1}^{d} q_i f(\frac{p_i}{q_i}) \tag{3.7}$$

Many common divergences, such as Kullback-Leibler (KL) divergence ($f(t) = tlnt$, $t = \frac{p_i}{q_i}$) [104], Hellinger distance ($f(t) = (\sqrt{t} - 1)^2$), χ^2 divergence ($f(t) = (t-1)^2$) are special cases of f-divergence, coinciding with a particular choice of f.
For analyzing the effectiveness of probabilistic-based distance measures, we chose four distances or divergences described below:

3.5.2.1 Jensen-Shannon divergence

The Jensen-Shannon (JS) divergence [127] is a symmetrized and smoothed version of the KL divergence, relative to Shannon's concept of uncertainty or "entropy" $H(P) = \sum_{i=1}^{n} p_i lnp_i$ [174].

$$d_{JS}(P,Q) = \frac{1}{2}\sum_{i=1}^{d} p_i ln(\frac{2p_i}{p_i + q_i}) + \frac{1}{2}\sum_{i=1}^{d} q_i ln(\frac{2q_i}{p_i + q_i}) \tag{3.8}$$

3.5.2.2 Bhattacharyya distance

Bhattacharyya distance [63] given in the equation (3.9), which is a value between 0 and 1, provides bounds on the Bayes misclassification probability and closely related to Hellinger distance.

$$d_B(P,Q) = -ln\sum_{i=1}^{d} \sqrt{p_i q_i} \tag{3.9}$$

3.5.2.3 Probabilistic Symmetric χ^2 divergence

Probabilistic Symmetric χ^2 divergence [63] is a special case of χ^2 divergence. It is a combination of Pearson χ^2 divergence and Newman χ^2 divergence.

$$d_{PChi}(P,Q) = 2\sum_{i=1}^{d} \frac{(p_i - q_i)^2}{p_i + q_i} \tag{3.10}$$

3.5.2.4 Taneja divergence

Taneja divergence [178] is utilized multiple ideas. It is a combination between KL divergence and Bhattacharyya distance, using KL-divergence with $p_i = \frac{p_i + q_i}{2}, q_i = \sqrt{p_i q_i}$

$$d_{TJ}(P,Q) = \sum_{i=1}^{d} (\frac{p_i + q_i}{2}) ln(\frac{p_i + q_i}{2\sqrt{p_i q_i}}) \tag{3.11}$$

3.5.3 Evaluation Methods

For each dataset, we obtained a clustering result from the k-means algorithm. To measure the quality of the clustering results, we used two evaluation indexes: Adjusted Rand Index (ARI) [96] and Adjusted Mutual Information (AMI) [180], which are widely used to evaluate the performance of unsupervised learning algorithms.

$P \setminus Q$	Q_1	Q_2	$\cdots$	Q_l	Sums
P_1	n_{11}	n_{12}	$\cdots$	n_{1l}	$n_{1\circ}$
P_2	n_{21}	n_{22}	$\cdots$	n_{2l}	$n_{2\circ}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
P_k	n_{k1}	n_{k2}	$\cdots$	n_{kl}	$n_{k\circ}$
Sums	$n_{\circ 1}$	$n_{\circ 2}$	$\cdots$	$n_{\circ l}$	$\sum_{ij} n_{ij} = n$

Table 3.2: The Contingency Table, $n_{ij} = |P_i \cap Q_j|$

Dataset	**#Docs**	**#Classes**	**< Class**	**> Class**
News20	18821	20	628	999
WebKB	8230	4	504	1641

Table 3.3: Statistics of the datasets. Where #Docs refers to the number of documents in the dataset, #Classes refers to the number of classes in the dataset and < Class, > Class, refers to the minimum number of documents and the maximum number of documents in a class.

3.5.3.1 Adjusted Rand Index

Adjusted Rand Index (ARI) [96], an adjusted form of Rand Index (RI), is defined as:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{n_{i\circ}}{2}\sum_j \binom{n_{\circ j}}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_{i\circ}}{2} + \sum_j \binom{n_{\circ j}}{2}] - [\sum_i \binom{n_{i\circ}}{2}\sum_j \binom{n_{\circ j}}{2}]/\binom{n}{2}} \tag{3.12}$$

where $n_{ij}, n_{i\circ}, n_{\circ j}, n$ are values from the contingency Table 3.2.

3.5.3.2 Adjusted mutual information

The Adjusted Mutual Information (AMI) [180], an adjusted form of mutual information (MI), is defined:

$$AMI(P,Q) = \frac{MI(P,Q) - E\{MI(P,Q)\}}{\max\{H(P), H(Q)\} - E\{MI(P,Q)\}} \tag{3.13}$$

where

$$H(P) = -\sum_{i=1}^{k}\frac{n_{i\circ}}{n}\log\frac{n_{i\circ}}{n}; MI(P,Q) = \sum_{i=1}^{k}\sum_{j=1}^{l}\frac{n_{ij}}{n}\log\frac{n_{ij}/n}{n_{i\circ}n_{\circ j}/n^2}.$$

Both ARI and AMI have a boundary above by 1. Higher values of ARI or AMI indicate more agreement between the two partitions. Please refer to the work of [96], [180] for more details.

3.5.4 Datasets and Setup

3.5.4.1 Datasets

The proposed methodology is evaluated on 2 miscellaneous datasets that are commonly used for the NLP community regarding the task of document clustering. Table 3.3 describes some statistics about the used datasets. The *20Newsgroup* collect has 18821 documents distributed across 20 different news categories. Each document corresponds to one article with a header that contains the title, the subject, and quoted text. The *WebKB* dataset contains 8230 web pages from the computer science department of different universities (e.g. Texas, Wisconsin, Cornell, etc).

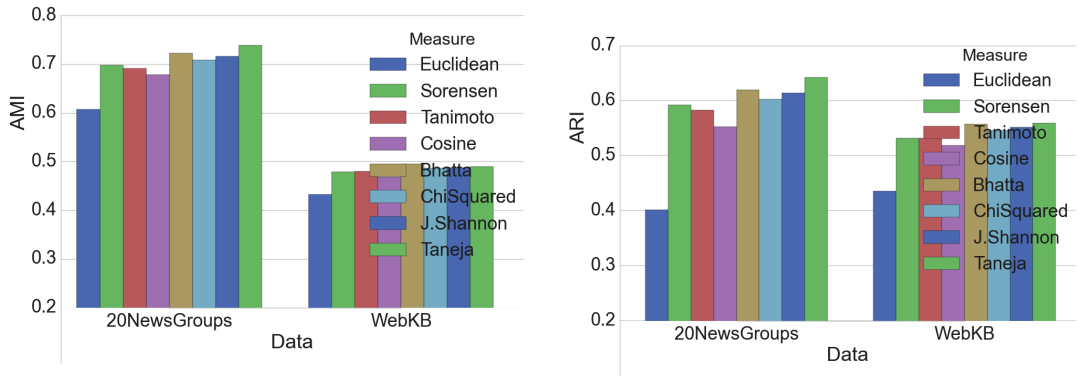
3.5.4.2 Setup

In our experiments, we compared eight distances used with *LDA+k-means* divided into the two categories: the PBM and VBM. We run LDA with Gibbs sampling method using the `topicmodels` R package[1]. The prior parameters α and β are respectively set to 0.1 and 0.01. These parameters were chosen according to the state-of-the-art standards [82]. The number of iterations of the Gibbs sampling is set to 5000. The input number of topics for the *20NewsGroups* dataset is set to 30 and for the *WebKB* dataset is set to 8. This number of topics will be confirmed in our experiments by testing different values. For each of the eight distances, we run the k-means 20 times with a maximum number of iterations equal to 1000. We compute the ARI and AMI on the results of each k-means iteration and report the average values.

3.5.5 Results

	20NewsGroups		WebKB	
Distances	ARI	AMI	ARI	AMI
Euclidean	0,402	0,608	0,436	0,432
Sorensen	0,592	0,698	0,531	0,479
Tanimoto	0,582	0,691	0,531	0,48
Cosine	0,552	0,678	0,519	0,468
Bhattacharyya	0,619	0,722	0,557	0,495
ChiSquared	0,602	0,708	0,545	0,487
JensenShannon	0,614	0,717	0,551	0,488
Taneja	0,642	0,739	0,559	0,489
VSM	0,128	0,372	0,268	0,335
LDA+Naive	0,434	0,590	0,171	0,197

Table 3.4: The average values of ARI, AMI for VSM, LDA-Naive, LDA+k-means with eight different distance measures for two datasets



(a) AMI

(b) ARI

Figure 3.4: The average values of ARI, AMI for LDA+k-means with eight different distance measures for two datasets

[1]https://cran.r-project.org/web/packages/topicmodels/index.html

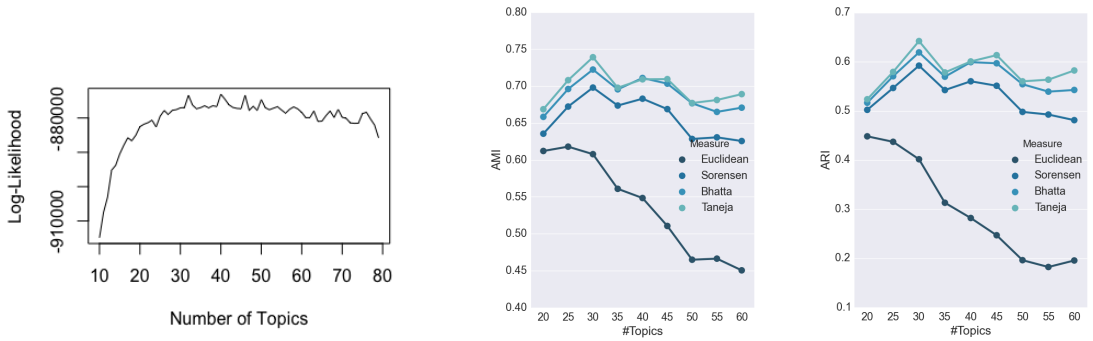
3.5.5.1 Comparing effectiveness of eight distance measures for *LDA +k-means*

The average values of the ARI and AMI are reported in Table 3.4. The average ARI and AMI values of the PBM group are better than the average values of the VBM group. We notice that the Euclidean distance has the worst results regarding the ARI and AMI criteria. In the PBM group, the best average values are obtained by the two distances Bhattacharyya and Taneja. Thus, we propose to work with Taneja or Bhattacharyya distance for *LDA+k-means*. For a better understanding of the results, we additionally provide a bar plot illustrated in Fig.3.4.

3.5.5.2 The role played by the number of topics for *LDA +k-means*

We chose the number of topics based on the Harmonic mean of Log-Likelihood (HLK) [47]. We notice in Fig.3.5a that the best number of topics are in the range of $[30, 50]$ of a maximum value of HLK. We run the *LDA+k-means* with a different number of topics and four distances: two from the PBM group, two from the VBM group including the Euclidean distance. We plot the evaluation with AMI and ARI in the Fig.3.5b and Fig.3.5c.

As the number of topics increases, the *LDA+k-means* with Euclidean distance decreases in performance. The Euclidean distance is clearly not suitable for the *LDA+k-means*. The other three used distances (i.e. Sorensen, Bhattacharyya, and Taneja) kept a steady behavior with a slight advantage for the Taneja distance. This is due to the fact that these distance were defined for probability distribution and thus are more suitable for the kind of input provided by LDA. We notice that after 50 topics the performance of the three distances decreases.

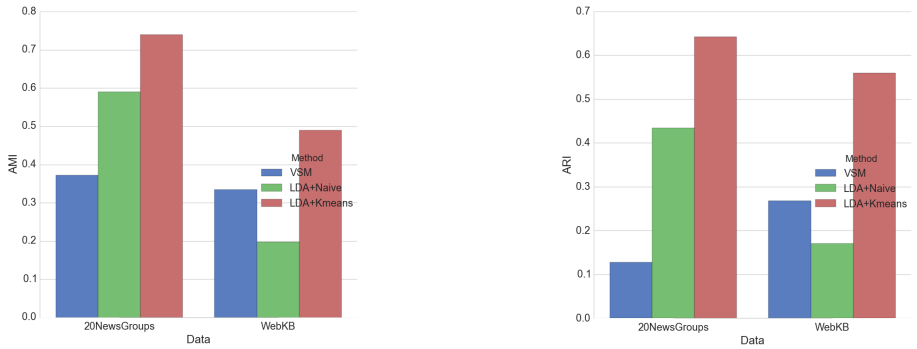


(a) Log-Likelihood (b) AMI (c) ARI

Figure 3.5: The Harmonic Mean of the Log-Likelihood and ARI, AMI values with four distances for 20NG dataset with different # of topics.

3.5.5.3 Comparing *LDA+k-means*, *LDA+Naive*, VSM

In order to study the role played by topic modeling, we compare three document clustering methods. The first is VSM that uses a word-frequency vector $d_{wf} = (wf_1, wf_2, \ldots, wf_n)$, where wf_i is the frequency of the ith word in the document as input for k-means [143]. The second is proposed in [131], which considers each topic as a cluster. In fact, document-topic distribution θ can be viewed as a mixture proportion vector over clusters and thus can be used for clustering as follows. Suppose that x is a cluster, a



(a) AMI (b) ARI

Figure 3.6: ARI, AMI values for three methods: VSM, *LDA+Naive*, *LDA+k-means* with Taneja distance computed on 20NGNewsGroups and WebKB datasets

document is assigned to x if $x = argmax_j \theta_j$. Note that this approach is a simple solution, usually referred to as a naive solution to combine topic modeling and document clustering. This approach is denoted in our experiments as *LDA+Naive*. The third one is the *LDA+k-means* with the probabilistic-based distance measure (eg. Bhattacharyya, Taneja). The results are plotted in Fig.3.6, we notice that the *LDA+k-means* used with Taneja distance obtains the best average results for both of the used datasets.

3.5.6 Related works

For the document clustering with k-means algorithm using Vector Space model (VSM), a lot of measures have been proposed for computing the similarity between two documents. Huang et al. (2008) [95] compared and analyzed the effectiveness of five distance or similarity measures (i.e. Euclidean, Cosine, Jaccard, Person correlation, Averaged Kullback-Leibler Divergence) with seven common text document datasets. Yung Shen Lin et al. (2014) [128] have presented a novel SMTP similarity measure between two documents by embedding several properties in this measure. Maher et al. (2016) [134] proposed SMTP (Similarity Measure for Text Processing) distance measures for document clustering and have shown that the performance obtained by the SMTP measure is much better than that achieved by Euclidean distance; Cosine similarity.

The drawback of k-means algorithm using VSM is the high-dimensional of input vectors. One of the possible solutions is to represent the text as a set of topics and use the topics as an input for a clustering algorithm. Through topic extraction, we can reduce the document feature dimensionality and measure the similarity between the documents. In fact, many researchers have applied LDA to document clustering. Wallach (2008) [182] proposed a cluster-based topic model (CTM) which introduces latent variables into LDA to model groups and each group owns a group-specific Dirichlet prior governing the sampling of document-topic distribution. Millar et al. (2009) presented a document clustering and visualization method based on LDA and self-organizing maps (LDA-SOM) [141]. Lu et al. (2011) [131] proposed a document clustering method which treats each topic as a cluster. Document-topic distribution θ can be deemed as a mixture proportion vector over clusters and can be utilized for clustering. A document is assigned to cluster x if $x = argmax_j \theta_j$.

Document-topic distribution θ extracted from LDA can be used as the input for clustering algorithms. This approach denoted *LDA+k-means*, is an integration of LDA and k-means. Xie and Xing (2013) [185] proposed a multi-grain clustering topic model

(MGCTM) which integrated document clustering and topic model into a unified framework and jointly performed the two tasks to achieve the overall best performance. They also compared their proposed method with other methods (e.g. *LDA+Naive*, CTM, *LDA+k-means*, etc.) but they only used Euclidean distance for *LDA+k-means* and clustering accuracy for evaluation. We noticed that the topic distribution extracted from LDA is a probability distribution. So, it is not really good for *LDA+k-means* if using Euclidean distance to measure the similarity between two probability distribution.

Our work in this section extends these works by examining the effectiveness of PBM versus VBM for *LDA+k-means*. We chose eight distance or similarity measures represented to eight distance/similarity measure families categorized by [52] and compared clustering evaluation using entropy, purity, F-measure on two common text datasets: *20NewsGroups* and *WebKB*.

3.5.7 Section Conclusion

In this section, we compared the effect of eight distance or similarity measures represented to eight distance measure families for clustering document using *LDA+k-means*. Experiments on two datasets with two evaluation criteria demonstrate the fact that the efficiency of Probabilistic-based measurement clustering is better than the Vector-based measurement clustering including Euclidean distance. Comparing among *LDA+k-means*, *LDA+Naive*, VSM, the experiments also show that if we choose the suitable value of a number of topic for LDA and Probabilistic-based measurements for k-means, *LDA+k-means* can improve the effect of clustering results.

3.6 Pretopological Approach for Multi-criteria Clustering

We address in this work the problem of document clustering. Our contribution proposes a novel unsupervised clustering method based on the structural analysis of the latent semantic space. Each document in the space is a vector of probabilities that represents a distribution of topics. The document membership to a cluster is computed taking into account two criteria: the major topic in the document (qualitative criterion) and the distance measure between the vectors of probabilities (quantitative criterion). We perform a structural analysis on the latent semantic space using the Pretopology theory that allows us to investigate the role of the number of clusters and the chosen centroids, in the similarity between the computed clusters. We have applied our method to Twitter data and showed the accuracy of our results compared to a random choice number of clusters.

The result of this section has been published in the *Proceedings of the Sixth International Symposium on Information and Communication Technology, SoICT 2015, Hue City, Vietnam, December 3-4, 2015.* ACM 2015, ISBN 978-1-4503-3843-1, pages: 38-45 [45] and the *Informatical Journal*, Vol 40, No 2 (2016), pages:169-180 [46].

3.6.1 Introduction

Clustering a set of documents is a standard problem addressed in data mining, machine learning, and statistical natural language processing. Document clustering can automatically organize many documents into a small number of meaningful clusters and find latent structure in unlabeled document collections.

In the previous sections, we proposed to use LDA to reduce the documents dimensionality (section 3.3) and also examined the role of probabilistic distances (section 3.5) when using *LDA+k-means*. The two problems we consider in this section are:

Q3.3 how to choose the number of clusters for k-means algorithms ?

Q3.4 how to cluster documents with multi-criteria ?

In this aim, we tackle the problem of clustering a set of documents by defining a family of binary relationships on the topic-based contents of the documents. The documents are not only grouped together using a measure of similarity but also using the pseudo-closure function built from a family of binary relationships between the different hidden semantic contents (i.e topics). The pseudo-closure function is defined using an original concept called Pretopology [22, 23]. The Pretopology theory offers a framework to work with categorical data, to establish multi-criteria distance for measuring the similarity between the documents and to build a process to structure the space and infer the number of cluster for k-means.

The idea of using Pretopology theory for k-means clustering has been proposed by [114]. In this paper, the authors proposed the method to find automatically a number k of clusters and k centroids for k-means clustering by results from structural analysis of minimal closed subsets algorithm [111] and also proposed to use pseudo-closure distance constructed from the relationships family to examine the similarity measure for both numeric and categorical data. The authors illustrated the method with a toy example about the toxic diffusion between 16 geographical areas using only one relationship.

For the problem of classification, the authors of this work [4] built a vector space with Latent Semantic Analysis (LSA) and used the pseudo-closure function from Pretopological theory to compare all the cosine values between the studied documents represented by vectors and the documents in the labeled categories. A document is added to a labeled category if it has a maximum cosine value.

Our work differs from the work of [4] and extended the method proposed in [114] within two dimensions: first, we exploited this idea in document clustering and integrated structural information from LDA using the pretopological concepts of pseudo-closure and minimal closed subsets introduced in [111]. Second, we showed that Pretopology theory can apply to multi-criteria clustering by defining the pseudo distance built from multi-relationships. In our work, we clustered documents by using two criteria: one based on the major topic of document (qualitative criterion) and the other based on Hellinger distance (quantitative criterion). The clustering is based on these two criteria but not on multicriteria optimization [70] for clustering algorithms. Our application on Twitter data also proposed a method to construct a network from the multi-relations network by choosing the set of relations and then applying strong or weak Pretopology.

The contributions of this section are as follows.

- We propose a new method, namely the Method of Clustering Documents using Pretopology and Topic Modeling (MCPTM), to cluster text documents using Pretopology and Topic Modeling.
- We investigate the role of the number of clusters inferred by our analysis of the documents and the role of the centroids in the similarity between the computed clusters.
- We conducted experiments with different distances measures and show that the distance measure that we introduced is competitive.

3.6.2 Our Approach

In our approach, we build The Method of Clustering Documents using Pretopology and Topic Modeling (MCPTM) (see Fig.3.7) which clusters documents via Topic Modeling

and pseudo-closure. MCPTM can be built by:

1. Defining the topic-distribution of each document d_i in corpus $\mathcal{D}$ by document structure analysis using LDA.
2. Defining two binary relationships: R_{MTP} based on major topic and R_{d_H} based on Hellinger distance.
3. Building the pseudo-closure function from two binary relationships R_{MTP}, R_{d_H}.
4. Building the pseudo-closure distance from the pseudo-closure function.
5. Determining initial parameters for the k-means algorithm from results of minimal closed subsets.
6. Using the *k-means* algorithm to cluster sets of documents with initial parameters from the result of minimal closed subsets, the pseudo-closure distance to compute the distance between two objects and the inter-pseudo-closure distance to re-compute the new centroids.

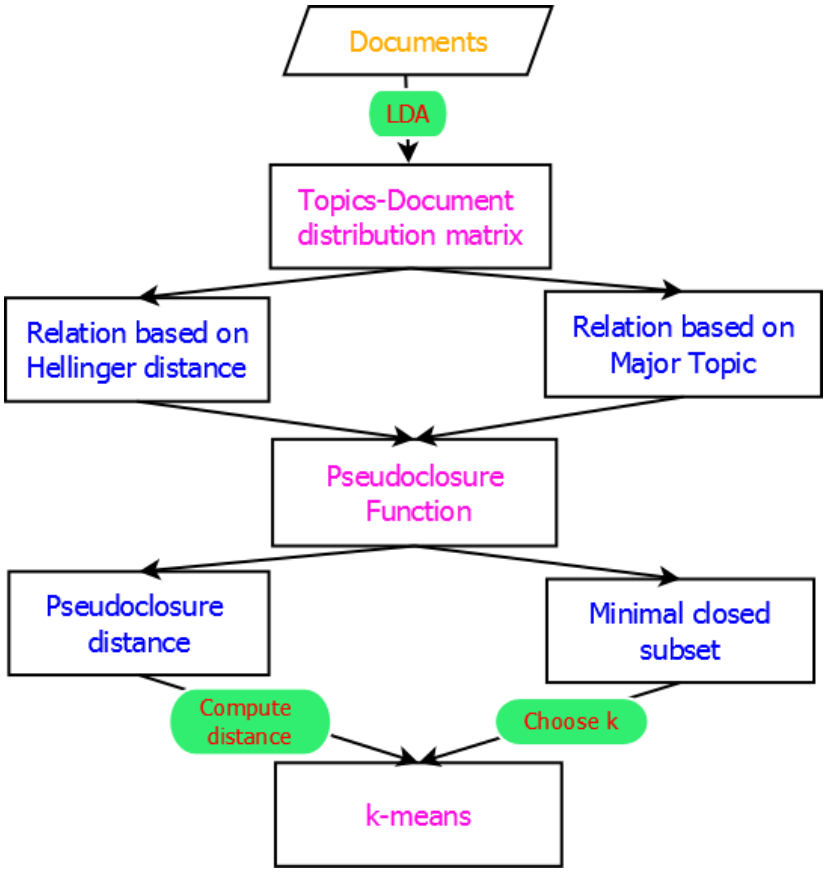


Figure 3.7: MCPTM Approach.

3.6.2.1 Document structure analysis by LDA

A term-document matrix is given as an input to LDA and it outputs two matrices:

- The document-topic distribution matrix θ.
- The topic-term distribution matrix ϕ.

The topic-term distribution matrix $\phi \in \mathbf{R}^{K \times V}$ consists of K rows, where the i-th row $\phi \in \mathbf{R}^{V}$ is the word distribution of topic i. The terms with high ϕ_{ij} values indicate that they are the representative terms of topic i. Therefore, by looking at such terms one can grasp the meaning of each topic without looking at the individual documents in the cluster.

In a similar way, the document-topics distributions matrix $\theta \in \mathbf{R}^{M \times K}$ consists of M rows, where the i-th row $\theta_i \in \mathbf{R}^{K}$ is the topic distribution for document i. A high probability value of θ_{ij} indicates that document i is closely related to topic j. In addition, documents with low θ_{ij} values over all the topics are noisy documents that belong to none of the topics. Therefore, by looking at the θ_{ij} values, one can understand how closely the document is related to the topic.

3.6.2.2 Defining binary relationships

By using LDA, each document may be characterized by its topic distribution and also be labeled by the topic with the highest probability. In this subsection, we use this information to define the relations between two documents based on the way we consider the "similarity" between them.

a) Based on major topic Firstly, based on the label information, we can consider connecting the documents if they have the same label. However, in some cases such as noisy documents, the probability of label topic is very small and it is not really good if we use this label to represent a document. Hence, we just use the label information if its probability is higher than threshold p_0. We define the major topic of each document as:

Definition 3.2. *$MTP(d_i)$ is the major topic of document d_i if $MTP(d_i)$ is the topic with highest probability in the topic distribution of document d_i and this probability is greater than threshold p_0, $p_0 \geq 1/K$, K is the number of topic.*

$$MTP(d_i) = \{k | \theta_{ik} = max_j \theta_{ij} \quad and \quad \theta_{ik} \geq p_0\} \tag{3.14}$$

Considering two documents d_m,d_n with their major topic $MTP(d_m)$, $MTP(d_n)$, we see that document d_m is close to document d_n if they have the same major topic. So, we proposed a definition of binary relationship R_{MTP} of two documents based on their major topic as:

Definition 3.3. *Document d_m has binary relationship R_{MTP} with document d_n if d_m and d_n have the same major topic.*

b) Based on Hellinger distance Secondly, we can use the topic distributions of documents to define the relation based the similarity between two real number vectors or two probability distributions. If we consider a probability distribution as a vector, we can choose some distances or similarity measures related to the vector distance such as Euclidean distance, Cosine Similarity, Jaccard Coefficient, Pearson Correlation Coefficient, etc. But, it is better if we choose distances or similarity measures related to the probability distribution such as Kullback-Leibler Divergence, Bhattacharyya distance, Hellinger distance, etc. We choose the Hellinger distance because it is a metric for measuring the deviation between two probability distributions, easily to compute and especially limited in $[0, 1]$.

Definition 3.4. *For two discrete probability distributions* $P = (p_1, \ldots, p_k)$ *and* $Q = (q_1, \ldots, q_k)$*, their Hellinger distance is defined as*

$$d_H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}, \tag{3.15}$$

The Hellinger distance is directly related to the Euclideanan norm of the difference of the square root vectors, i.e.

$$d_H(P, Q) = \frac{1}{\sqrt{2}} \left\| \sqrt{P} - \sqrt{Q} \right\|_2.$$

The Hellinger distance satisfies the inequality of $0 \leq d_H \leq 1$. This distance is a metric for measuring the deviation between two probability distributions. The distance is 0 when $P = Q$. Disjoint P and Q shows the maximum distance of 1. The lower the value of the Hellinger distance, the smaller the deviation between two probability distributions. So, we can use the Hellinger distance to measure the similarity between two documents d_m,d_n. We then define the binary relationship R_{d_H} between two documents as:

Definition 3.5. *Document* d_m *has binary relationship* R_{d_H} *with document* d_n *if* $d_H(d_m, d_n) \leq d_0, 0 \leq d_0 \leq 1$*,* d_0 *is the accepted threshold.*

3.6.2.3 Building pseudo-closure function

Based on two binary relationships R_{MTP} and R_{d_H}, we can build the neighborhood basis (see. Algorithm 3.1) and then build the pseudo-closure (see Algorithm 3.2) for strong (with intersection operator) and weak (with union operator) Pretopology.

Algorithm 3.1 Neighborhood Basis Using Topic Modeling.

Require: document-topic distribution matrix θ, corpus $\mathcal{D}$
Require: R_{MTP}, R_{d_H}: family of relations.
procedure NEIGHBORHOOD-TM($\mathcal{D}, \theta, R_{MTP}, R_{d_H}$)
  loop for each relation $R_i \in \{R_{MTP}, R_{d_H}\}$
    loop for each document $d_m \in \mathcal{D}$
      loop for each document $d_n \in \mathcal{D}$
        If $R_i(d_m, d_n)$ **then**
          $B_i[d_m].append(d_n)$
  return $B = [B_1, B_2]$ ▷ Output
end procedure

Algorithm 3.2 Pseudoclosure using Topic Modeling.

Require: $B = (B_1, B_2), \mathcal{D} = \{d_1, \ldots, d_M\}$
procedure PSEUDOCLOSURE($A, B, \mathcal{D}$)
  aA=A
  loop for each document $d_n \in \mathcal{D}$
    If $(A \cap B_1[d_n] \neq \emptyset \quad or \quad A \cap B_2[d_n] \neq \emptyset)$ **then**
      $aA.append(d_n)$
  return aA ▷ Ouput
end procedure

3.6.2.4 Building pseudo-closure distance

In standard *k-means*, the centroid of a cluster is the average point in the multidimensional space. Its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster which are not effective with categorical data analysis.

On the other hand, the pseudo-closure distance (see definition 1.24, 1.25 in chapter 1) built from pseudo-closure function is used to examine the similarity using both numeric and categorical data. Therefore, it can contribute to improving the classification with k-means.

3.6.2.5 Structure analysis with minimal closed subsets

The two limits of the standard *k-means* algorithm are the number of clusters which must be predetermined and the randomness in the choice of the initial centroids of the clusters. Pretopology theory gives a good solution to omit these limits by using the result from minimal closed subsets. The algorithm to compute minimal closed subset is presented in algorithm 1.2, chapter 1. By performing the minimal closed subset algorithm, we get the family of minimal closed subsets. This family, by definition, characterizes the structure underlying the dataset E. So, the number of minimal closed subsets is a quite important parameter: it gives us the number of clusters to use in the *k-means* algorithm. Moreover, the initial centroids for starting the *k-means* process can be determined by using the interior-pseudo-distance for each minimal closed subset $F_{m_j} \in \mathcal{F}_m$ (x_0 is chosen as centroid of F_{m_j} if $D_{F_{m_j}}(x_0) = min_{x \in F_{m_j}} D_{F_{m_j}}(x)$).

3.6.2.6 MCPTM algorithm

In this subsection, we present The Method of Clustering Documents using Pretopology and Topic Modeling (MCPTM) which clusters documents via the Topic Modeling and pseudo-closure. At first, an LDA Topic Modeling is learned on the documents to achieve topic-document distributions. The major topic and Hellinger probability distance are used to define relations between documents and these relations are used to define a pretopological space which can be employed to get preliminarily clusters of a corpus and determine the number of clusters. After that, k-means clustering algorithm is used to cluster documents with pseudo-distance and inter-pseudo-distance. The MCPTM algorithm is presented in algorithm 3.3.

Algorithm 3.3 The MCPTM algorithm: clustering documents using Pretopology and Topic Modeling.

Require: $\mathcal{D}$: corpus from set of documents
procedure MCPTM($\mathcal{D}$)
  $\theta_{\mathcal{D}} \leftarrow$ LDA-GIBBS($\mathcal{D}$, α, β, T)
  $B \leftarrow$ NEIGHBORHOOD-TM($\mathcal{D}, \theta_{\mathcal{D}}$, R_{MTP}, R_{d_H})
  $aA \leftarrow$ pseudoCLOSURE(B)
  $\mathcal{F}_m \leftarrow$ MIMINAL-CLOSED-SUBSETS($\mathcal{D}$, $aA()$)
  $k = |\mathcal{F}_m|$: number of clusters
  $M = \{m_i\}_{,i=1,\ldots,k}, m_i = Centroid(F_{m_i})$
  while clusters centroids changed **do**
    for each x $\in E - M$ **do**
      compute $\delta(x, m_i), i = 1, \ldots, k$
      find m_0 with $\delta(x, m_0) = min\delta(x, m_i)_{i=1,\ldots,k}$
      $F_{m_0} = F_{m_0} \cup \{x\}$
    end for
    Recompute clusters centroids M.
  end while
  return $Clusters = \{F_1, F_2, \ldots, F_k\}$ ▷ Output
end procedure

3.6.3 Application and Evaluation

The microblogging service Twitter has become one of the major micro-blogging websites, where people can create and exchange content with a large audience. In this section, we apply the MCPTM algorithm for clustering a set of users around their interests. We

have targeted 133 users and gathered their tweets in 133 documents. We have cleaned them and run the *LDA Gibbs Sampling* algorithm to define the topics distribution of each document and words distribution of each topic. We have used then, the *MCPTM* algorithm to automatically detect the different communities for clustering users. We present in the following, the latter steps in more details.

3.6.3.1 Data collection

Twitter is a micro-blogging social media website that provides a platform for the users to post or exchange text messages of 140 characters. Twitter provides an API that allows easy access to anyone to retrieve at most a 1% sample of all the data by providing some parameters. In spite of the 1% restriction, we are able to collect large data sets that contain enough text information for Topic Modeling as shown in [146].

The dataset contains tweets from the 133 most famous and most followed public accounts. We have chosen these accounts because they are characterized by the heterogeneity of the tweets they posts. The followers that they aim to reach comes from different interest areas (i.e. politics, technology, sports, art, etc..). We used the API provided by Twitter to collect the messages of 140 characters between January and February 2015. We gathered all the tweets from a user into a document.

3.6.3.2 Data pre-processing

Social media data and mainly Twitter data is highly unstructured: typos, bad grammar, the presence of unwanted content, for example, humans expressions (happy, sad, excited, ...), URLs, stop words (the, a, there, ...). To get good insights and to build better algorithms it is essential to play with clean data. The pre-processing step gets the textual data clean and ready as input for the MCPTM algorithm.

3.6.3.3 Topic Modeling results

After collecting and pre-processing data, we obtained data with 133 documents, 158,578 words in the corpus which averages 1,192 words per document and 29,104 different words in the vocabulary. We run LDA Gibbs Sampling from algorithm 2.1 and received the output with two matrices: the document-topic distribution matrix θ and the distribution of terms in topics represented by the matrix ϕ. We present in table 3.5 two topics from the list of 20 topics that we have computed with our LDA implementation. A topic is presented with a distribution of words. For each topic, we have a list of users. Each user is identified with an ID from 0 to 132 and is associated with a topic by an order of probabilities. The two lists of probabilities in topic 3, 10 are extracted respectively from θ and ϕ distributions. The topic 3 and topic 10 are of particular interest due to the important number of users that are related to them. Topic 3 is about the terrorist attack that happened in Paris and topic 10 is about the international Consumer Electronics Show (CES). Both events happened at the same time that we collected our data from Twitter. We note that we have more users for these topics than from other ones. We can conclude that these topics can be considered as hot topics at this moment.

Due to the lack of space, we could not present in details all the topics with their distribution of words and all topic distributions of documents. Therefore, we presented eight topic distributions θ_i (sorted by probability) of eight users in the table 3.6. A high probability value of θ_{ij} indicates that document i is closely related to topic j. Hence, user ID 12 is closely related to topic 3, user ID 22 closely related to topic 10, etc. In addition, documents with low θ_{ij} values over all the topics are noisy documents that

Table 3.5: Words - Topic distribution ϕ and the related users from the θ distribution

Topic 3				
Words	**Prob.**	**Users**	**ID**	**Prob.**
paris	0.008	GStephanopoulos	42	0.697
charliehebdo	0.006	camanpour	23	0.694
interview	0.006	AriMelber	12	0.504
charlie	0.005	andersoncooper	7	0.457
attack	0.005	brianstelter	20	0.397
warisover	0.004	yokoono	131	0.362
french	0.004	piersmorgan	96	0.348
today	0.004	maddow	72	0.314
news	0.004	BuzzFeedBen	21	0.249
police	0.003	MichaelSteele	81	0.244

Topic 10				
Words	**Prob.**	**Users**	**ID**	**Prob.**
ces	0.010	bxchen	22	0.505
people	0.007	randizuckerberg	102	0.477
news	0.006	NextTechBlog	88	0.402
media	0.006	lheron	71	0.355
tech	0.006	LanceUlanoff	68	0.339
apple	0.006	MarcusWohlsen	74	0.339
facebook	0.005	marissamayer	76	0.334
yahoo	0.005	harrymccracken	43	0.264
app	0.005	dens	33	0.209
google	0.004	nickbilton	89	0.204

Table 3.6: Topics - document distribution θ

User ID 02		**User ID 12**		**User ID 22**	
Topic	**Prob.**	**Topic**	**Prob.**	**Topic**	**Prob.**
10	0.090	3	0.504	10	0.506
16	0.072	19	0.039	3	0.036
12	0.065	10	0.036	19	0.034
18	0.064	15	0.035	14	0.031
0	0.058	13	0.032	4	0.03

User ID 53		**User ID 75**		**User ID 83**	
Topic	**Prob.**	**Topic**	**Prob.**	**Topic**	**Prob.**
17	0.733	19	0.526	8	0.249
1	0.017	2	0.029	0	0.084
18	0.016	3	0.029	11	0.06
13	0.016	5	0.028	7	0.045
11	0.015	105	0.028	12	0.043

belong to none of the topics. So, there is no major topic in user ID 02 and user ID 132 (the max probability < 0.15).

We show in Table 3.7 clusters of documents based on their major topics in two levels with their probabilities. The documents with the highest probability less than 0.15 are considered noisy documents and clustered in the same cluster.

Table 3.7: Classifying documents based on their major topic

Major Topic	**prob $\geq$ 0.3**	**0.15 $<$ prob $<$ 0.3**
Topic 0	112,85,104	-
Topic 1	44,129,114	61
Topic 2	101,108,91	90
Topic 3	42,23,12,7,20, 131,96,72	21,81,93,10
Topic 4	125,36,123,0	-
Topic 5	82,126	62
Topic 6	127,37,26	92
Topic 7	118,106,32	70,4
Topic 8	113	83,55,59
Topic 9	67,122	111,100
Topic 10	22,102,88,71,74, 68,76	43,89,33,65
Topic 11	54,51,121	29,94
Topic 12	50	12
Topic 13	16,35	38
Topic 14	31,98	-
Topic 15	66,73,34,	48
Topic 16	99	-
Topic 17	53,30	-
Topic 18	47,128,1,124,5	78,115
Topic 19	14,80,39,75,18,103	-
None	**remaining users (probability** < 0.15**)**	

3.6.3.4 Results from the k-means algorithm using Hellinger distance

After receiving the document-topic distribution matrix θ from LDA Gibbs Sampling, we used the k-means algorithm with Hellinger distance to cluster users. Table 3.8 presents the result from the k-means algorithm using Hellinger distance with a number of clusters k=13 and random centroids. Based on the mean value of each cluster, we defined the major topic related to the clusters and attached these values in the table. We notice that different choices of initial seed sets can result in very different final partitions.

3.6.3.5 Results from the MCPTM algorithm

After getting the results (e.g table 3.6) from our LDA implementation, we defined two relations between two documents, the first based on their major topic R_{MTP} and the second based their Hellinger distance R_{d_H}. We then built the weak pseudo-closure with these relations and applied it to compute pseudo-closure distance and the minimal closed subsets. With this pseudo-closure distance, we can use the MCPTM algorithm to cluster sets of users with multi-relationships.

Table 3.9 presents the results of the MCPTM algorithm and the *k-means* algorithm using Hellinger distance. We notice that there is almost no difference between the results from two methods when using the number of clusters k and initial centroids above.

Figure 3.9 shows the number of elements of minimal closed subsets with different thresholds p_0 for R_{MTP} and d_0 for R_{d_H}. We used this information to choose the number of clusters. For this example, we chose $p_0 = 0.15$ and $d_0 = 0.15$ i.e user i connects with user j if they have the same major topic (with probability ≥ 0.15) or the Hellinger distance $d_H(\theta_i, \theta_j) \leq 0.15$. From the network (figure 3.8) for 133 users built from the

Table 3.8: Result from k-means algorithm using Hellinger distance

Cluster	Users	Major Topic
1	67, 111, 122	TP 9 (0.423)
2	34, 48, 66, 73	TP 15 (0.315)
3	10, 22, 33, 43, 65, 68, 71, 74, 76, 88, 89, 98, 102	TP 10 (0.305)
4	26, 92	TP 6 (0.268)
5	16, 35, 44, 90, 91, 101, 108, 114, 129	TP 2 (0.238)
6	4, 32, 70, 106, 118	TP 7 (0.345)
7	37, 127	TP 6 (0.580)
8	14, 18, 39, 75, 80, 103	TP 19 (0.531)
9	1, 5, 47, 78, 124, 128	TP 18 (0.453)
10	30, 53	TP 17 (0.711)
11	7, 12, 20, 21, 23, 42, 72, 81, 93, 96, 131	TP 3 (0.409)
12	0, 31, 36, 82, 123, 125	TP 4 (0.310)
13	remaining users	None

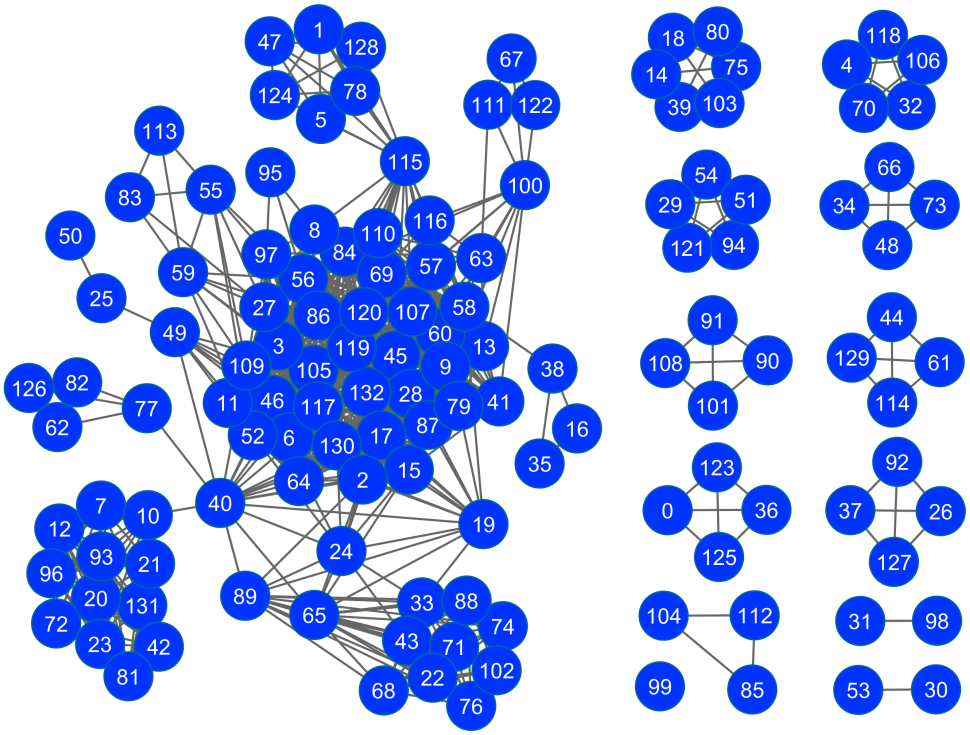


Figure 3.8: Network for 133 users with two relationships based on Hellinger distance ($distance \leq 0.15$) and Major topic (probability ≥ 0.15).

weak pseudo-closure, we chose the number of clusters $k = 13$ since the network has 13 connected components (each component represents an element of the minimal closed subset). We used inter-pseudoslosure distance to compute initial centroids and received the result:

$$\{0, 52, 4, 14, 26, 29, 30, 31, 34, 44, 85, 90, 99\}$$

We saw that the largest connected component in the users' network (fig. 3.8) has

Table 3.9: Result from k-means algorithm using Hellinger distance and MCPTM

	k-means & Hellinger		MCPTM Algorithm	
Cluster	**Users**	**Topic**	**Users**	**Topic**
1	0,36,123,125	TP 4 (0.457)	0,36,123,125	TP 4
2	4,32,70,10,118	TP 7 (0.345)	4,32,70,10,118	TP 7
3	14,18,39,75,80,103	TP 19 (0.531)	14,18,39,75,80,103	TP 19
4	26,37,92,127	TP 6 (0.424)	26,37,92,127	TP 6
5	29,51,54,94,121	TP 11 (0.345)	29,51,54,94,121	TP 11
6	30,53	TP 17 (0.711)	30,53	TP 17
7	31	TP 14 (0.726)	31,98	TP 14
8	34,48,66,73	TP 15 (0.315)	34,48,66,73	TP 15
9	44,61,114,129	TP 1 (0.413)	44,61,114,129	TP 1
10	85,104,112	TP 0 (0.436)	85,104,112	TP 0
11	67,90,91,101,108	TP 2 (0.407)	90,91,101,108	TP 2
12	99	TP 16 (0.647)	99	TP 16
13	remaining users	None	remaining users	None

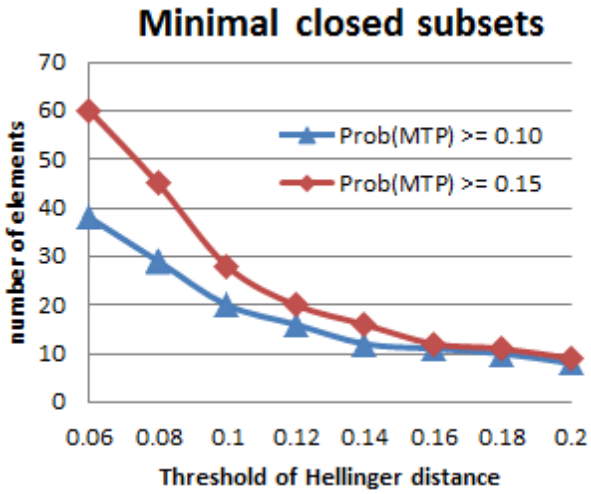


Figure 3.9: Number of elements of Minimal closed subsets with difference thresholds p_0 for R_{MTP} and d_0 for R_{d_H}.

many nodes with weak ties. This component represents the cluster 13 with 89 elements. It contains the 8 remaining topics that were nonsignificant or contains noisy documents without major topics. Hence, we used the *k-means* algorithm with Hellinger distance for clustering this group with number of clusters $k = 9$, centroids:

$$\{23, 82, 113, 67, 22, 50, 16, 47, 2\}$$

and showed the result in the table 3.10.

3.6.3.6 Evaluation

In this part, we conducted an evaluation of our algorithm by comparing similarity measure of MCPTM (using the pseudocloseure distance with information from results of minimal closed subsets) and k-means with random choice. The evaluation is performed as follows: we firstly discovered the similarity measure of k-means using three distances: Euclidean distance, Hellinger distance and pseudo-closure distance; we then compared

Table 3.10: Result from k-means algorithm using Hellinger distance for cluster 13 (89 users)

Cluster	**Users**	**Major Topic**
13.1	7, 12, 20, 21, 23, 42, 72, 81, 93, 96, 131	TP 3 (0.409)
13.2	62, 77, 82, 126	TP 5 (0.339)
13.3	27, 55, 59, 83, 113	TP 8 (0.218)
13.4	67, 111, 122	TP 9 (0.422)
13.5	22, 33, 43, 65, 68, 71, 74, 76, 88, 89, 102	TP 10 (0.330)
13.6	50	TP 12 (0.499)
13.7	16, 35	TP 13 (0.576)
13.8	1, 5, 47, 78, 124, 128	TP 18 (0.453)
13.9	remaining users	None

Table 3.11: The results of the clustering similarity for k-means with different distance measures. The abbreviation E stands for Euclidean distance, H for Hellinger distance (see definition 3.4) and P for the pseudo-closure distance (see definition 1.24 and 1.25).

	Same algorithm			**Same centroids**		
k	**E**	**H**	**P**	**E vs H**	**E vs P**	**H vs P**
5	0.423	0.454	0.381	0.838	0.623	0.631
9	0.487	0.544	0.423	0.831	0.665	0.684
13	0.567	0.598	0.405	0.855	0.615	0.633
17	0.645	0.658	0.419	0.861	0.630	0.641
21	0.676	0.707	0.445	0.880	0.581	0.604
25	0.736	0.720	0.452	0.856	0.583	0.613
29	0.723	0.714	0.442	0.864	0.578	0.600
mean	0.608	0.628	0.423	0.855	0.611	0.629
	Different centroids			**Inter-pseudo centroids**		
k	**E**	**H**	**P**	**E vs H**	**E vs P**	**H vs P**
5	0.434	0.373	0.383	-	-	-
9	0.495	0.383	0.447	-	-	-
13	0.546	0.445	0.469	**0.949**	**0.922**	**0.946**
17	0.641	0.493	0.518	-	-	-
21	0.687	0.478	0.491	-	-	-
25	0.715	0.519	0.540	-	-	-
29	0.684	0.4885	0.511	-	-	-
mean	0.600	0.454	0.480	**0.949**	**0.922**	**0.946**

similarity measures among three distances and the similarity measure when we use the number of clusters and the initial centroids from the result of minimal closed subsets. We used the similarity measure proposed by [179] to calculate the similarity between two clusterings of the same dataset produced by two different algorithms, or even the same k-means algorithm. This measure allows us to compare different sets of clusters without reference to external knowledge and is called internal quality measure.

Similarity measure To identify a suitable tool and algorithm for clustering that produces the best clustering solutions, it becomes necessary to have a method for comparing the different results in the produced clusters. To this matter, we used in this article the method proposed by [179].

To measure the "similarity" of two sets of clusters, we define a simple formula here: Let $C = \{C_1, C_2, \ldots, C_m\}$ and $D = \{D_1, D_2, \ldots, D_n\}$ be the results of two clustering algorithms on the same dataset. Assume C and D are "hard" or exclusive clustering algorithms where clusters produced are pair-wise disjoint, i.e., each pattern from the dataset belongs to exactly one cluster. Then the similarity matrix for C and D is an

$m \times n$ matrix $S_{C,D}$.

$$S_{C,D} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & \dots & S_{1n} \\ S_{21} & S_{22} & S_{23} & \dots & S_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ S_{m1} & S_{m2} & S_{m3} & \dots & S_{mn} \end{bmatrix} \tag{3.16}$$

where $S_{ij} = \frac{p}{q}$, which is Jaccard's Similarity Coefficient with p being the size of the intersection and q being the size of the union of cluster sets C_i and D_j. The similarity of clustering C and clustering D is then defined as

$$Sim(C, D) = \frac{\sum_{1 \leq i \leq m, 1 \leq i \leq m} S_{ij}}{\max(m, n)} \tag{3.17}$$

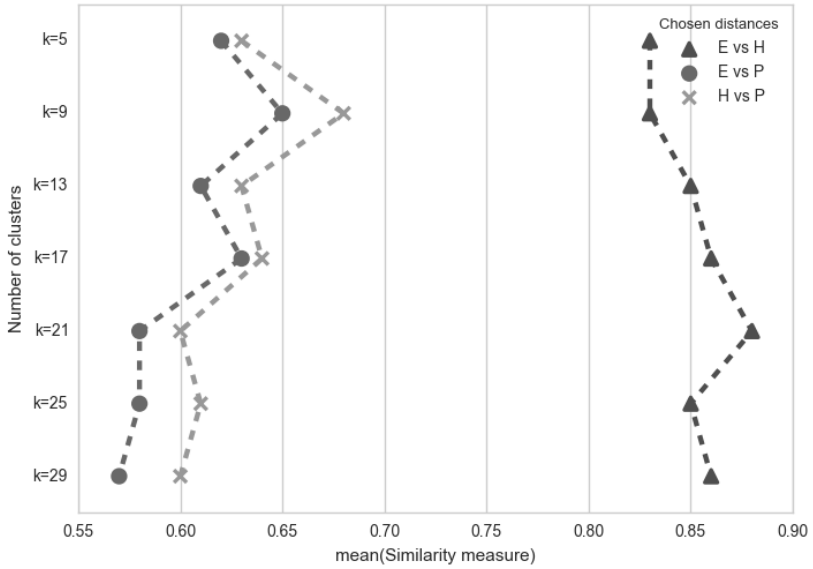


Figure 3.10: Illustration of the similarity measure where we have the same initial centroids. The abbreviation E stands for Euclidean distance, H for Hellinger distance an P for the pseudo-closure distance.

Discussion We have compared the similarity measure between 3 k-means algorithms with different initializations of the centroids and different numbers of clusters k. We plotted the similarity measure between the clusters computed with the 3 *k-means* algorithms with the same initial centroid in Fig.3.10 and the 3 k-means algorithms with different initial centroids in Fig.3.11. We notice that in both figures, the Euclidean distance and the Hellinger distance have higher similarity measure. This is due to the fact that both distances are similar. In Fig.3.10, we see a big gap between the clusters of Euclidean distance, Hellinger distance and the clusters from the pseudo-closure distance. This gap is closing in Fig.3.11 and starts opening again from $k = 17$. With different initial centroids, the pseudo-closure distance closed the gap between the k-means algorithms using Euclidean and Hellinger distance. But, when $k > 13$, the number of closed subsets, the gap between the pseudo-closure and the other distances starts opening again. In table 3.11 where we applied the same algorithm twice, the similarity measure between two clusters results from k-means is low for all three distances: Euclidean, Hellinger, pseudo-closure distance. The different choices of initial centroids can result in very different final partitions.

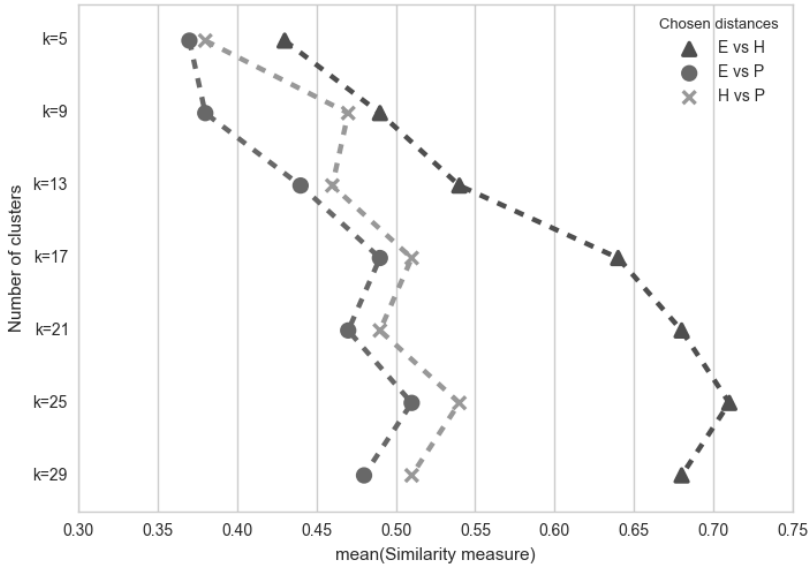


Figure 3.11: Illustration of the similarity measure where we have diffirent initial centroids. The appreviation E stands for Euclidean distance, H for Hellinger distance an P for the pseudo-closure distance.

For k-means, choosing the initial centroids is very important. Our algorithm MCPTM offers a way to compute the centroids based on the analysis of the space of data (in this case text). When we use the centroids computed from the results of minimal closed subsets that we present in Table 3.9, we have the higher similarity: 0,949 for Euclidean vs Hellinger; 0,922 for Euclidean vs pseudo-closure and 0,946 for Hellinger vs pseudo-closure. It means that the results from k-means using the centroids:

$$\{0, 52, 4, 14, 26, 29, 30, 31, 34, 44, 85, 90, 99\}$$

is very similar with all three distances Euclidean, Hellinger, pseudo-closure. We can conclude that the result that we obtained from our MCPTM algorithm is a good result for clustering with this Twitter dataset.

3.6.4 Section Conclusion

The major finding in this section is that the number of clusters and the chosen criteria for grouping the document is closely tied to the accuracy of the clustering results. The method presented here can be considered as a pipeline where we associate LDA and pseudo-closure function. LDA is used to estimate the topic-distribution of each document in a corpus and the pseudo-closure function to connect documents with multi-relations built from their major topics or Hellinger distance. By using both quantitative data and categorical data, this method allows us to have multi-criteria clustering. We have presented our contribution by applying it to microblogging posts and have obtained good results. In future works, we want to investigate these results on large scale and more conventional benchmark datasets. We also intend to parallelize the developed algorithms.

3.7 Conclusion

In this chapter, after recalling the classical approach for document clustering by presenting VSM, we proposed to combine Topic Modeling and k-means as a new method, namely *LDA+k-means*, for document clustering which can solve the problem of the high-dimensional matrix of document representation using VSM. For *LDA+k-means*, we considered three issues when applying the method for document clustering: choosing the "good" distance measure, choosing the number of clusters and clustering with multi-criteria.

For the first issue, we compared the effect of eight distance or similarity measures represented to eight distance measure families categorized by [52]. Our experiments demonstrate the fact that the efficiency of Probabilistic-based measurement clustering is better than the Vector-based measurement clustering including Euclidean distance. Comparing among *LDA+k-means*, *LDA+Naive*, VSM, the experiments also showed that if we choose the suitable value of a number of topic for LDA and PBM for k-means, *LDA+k-means* can improve the effect of clustering results.

We proposed to use pretopology to solve the remaining two issues. Firstly, we showed that the result from structure analysis by using minimal closed subsets can be used to choose the number of cluster for input of k-means algorithms. By defining the pseudo-closure function to connect documents with multi-relations, we can cluster documents with multi-criteria. In our work, we built the pseudo-closure function from their major topics or Hellinger distance. We have presented our contribution via introducing MCPTM algorithm, apply it to microblogging posts and have obtained good results.

Part III

Pretopology and Topic Modeling for Complex Network Analysis

Chapter 4

Stochastic Pretopology as a tool for Complex Networks Analysis

Complex networks modeling generally are based on graph theory. However, graph theory is not necessarily adapted to represent complex interactions with non-regular topologies that often occur in real-world complex networks. We are proposing in this work a novel approach for complex network analysis by introducing *Stochastic Pretopology*, a result of the combination of Pretopology theory and Random Sets. We firstly show how pretopology generalizes the graph theory to deal with complex interactions in complex networks. By connecting with random set theory, we then give the definition of stochastic pretopology and also propose the ways to build this kind of pretopology in many situations. That is to say how proximity with randomness can be delivered to model complex neighborhoods formation in complex networks. In addition, we also show how stochastic pretopology can be applied for modeling dynamic processes on complex networks by representing classical information diffusion models under stochastic pretopology language and then proposing *Pretopology Cascade Model* as a general model for information diffusion process that can take place in more complex networks such as multi-relational networks or stochastic graphs.

The result of this chapter has been published in the *Proceedings of ACIIDS 2018, Part II. Lecture Notes in Computer Science 10752*, 2018, ISBN 978-3-319-75419-2, pages: 102-111 [43] and the "Journal of Information and Telecommunication", ISSN: 2475-1839 (Print) 2475-1847 (Online), 2018.

4.1 Introduction

Complex system is a system composed of many interacting parts, such that the collective behavior of its parts together is more than the "sum" of their individual behaviors [149]. The topology of complex systems (who interact with whom) is often specified in terms of complex networks that are usually modeled by graphs, composed by vertices or nodes and edges or links. Graph theory has been widely used the conceptual framework of network models, such as random graphs, small world networks, scale-free networks [150, 67].

However, having more complicated non-regular topologies, complex networks need a more general framework for their representation [149]. To overcome this issue, we propose using *Stochastic Pretopology* built from the mixing between Pretopology theory [23] and Random Sets theory [144, 152]. Pretopology is a mathematical tool for modeling the concept of proximity which allows us to follow structural transformation processes as they evolve while random sets theory provides the good ways for handling what happens in a stochastic framework at the sets' level point of views.

In the section 1.6, chapter 1, we showed pretopology as an extension of graph theory in the *Claude Berge sense* [24] which leads us to the definition of pretopology networks as a general framework for network representation and then presented an example of a complex group interactions model by using pretopology theory.

In this chapter, by connecting to random sets theory for dealing with the random factors that often occur in real-world complex networks, we firstly give the definition of *Stochastic Pretopology* (SP) and then show how we construct stochastic pseudo-closure functions for various contexts. These functions are useful for modeling the complex networks in different spaces such as metric space, valued or binary relation spaces, etc. These models can be also convenient to handle phenomena in which collective behavior of a group of elements can be different from the summation of element behaviors composing the group.

After showing how stochastic pretopology can be applied for modeling dynamic processes on complex networks by representing classical information diffusion models such as *Independent Cascade model* [78] and *Linear Threshold model* [81] under stochastic pretopology language, we then propose *Pretopology Cascade Model* as a general information diffusion model in which complex random neighborhoods set can be captured by using stochastic pseudo-closure functions. We also present this model in two specific kinds of complex networks: multi-relational networks and stochastic graphs. Stochastic graphs presented in this chapter are defined by extending the definition of graph in the *Claude Berge sense* [24] $G = (V, \Gamma)$. In this approach, by considering Γ function as a finite random set defined from a degree distribution, we give a general graph-based network model in which *Erdős–Rényi model* and *scale-free networks* are special cases. For proposed models, after describing the algorithms, we illustrate the model by running some experiments.

The rest of this chapter is organized as follows: section 4.2 present stochastic pretopology when we connect pretopology and random set. After representing some classical information diffusion models under stochastic pretopology language in the section 4.3, we present in section 4.4 *Pretopology Cascade Model* as an application of stochastic pretopology in information diffusion and we then conclude our work in section 4.5.

4.2 Stochastic Pretopology (SP)

Complex systems usually involve structural phenomena, under stochastic or uncontrolled factors. In order to follow these phenomena step by step, we need concepts which allow modeling dynamics of their structure and take into account the factors' effects. As we showed in the previous section, we propose to use pretopology for modeling the dynamics of phenomena; the non-idempotents of its pseudo-closure function makes it suitable for such a modeling. Then, we introduce stochastic aspects to handle the effects of factors influencing the phenomena. For that, we propose using a theory of random sets by considering that, given a subset A of the space, its pseudo-closure $a(A)$ is considered as a random set. So, we have to consider the pseudo-closure not only as a set transform but also as a random correspondence.

Stochastic pretopology was first basically introduced in chapter 4 of [23] by using a special case of random set (the simple random set) to give three ways to define stochastic pretopology. We have also given some applications of stochastic pretopology such as: modeling pollution phenomena [107] or studying complex networks via a stochastic pseudo-closure function defined from a family of random relations [21]. Since we will deal with complex networks in which set of nodes V is a finite set, we propose in this chapter another approach for building stochastic pretopology by using finite random set theory [152].

From now on, V denotes a finite set. $(\Omega, \mathcal{A}, \mathbb{P})$ will be a *probability space*, where: Ω is a set, representing the *sample space* of the experiment; $\mathcal{A}$ is a σ-algebra on Ω, representing *events* and $\mathbb{P} : \Omega \to [0, 1]$ is a *probability measure*.

4.2.1 Finite Random Set

Definition 4.1. *A finite random set (FRS) with values in $\mathcal{P}(V)$ is a map $X : \Omega \to \mathcal{P}(V)$ such as*

$$X^{-1}(\{A\}) = \{\omega \in \Omega : X(\omega) = A\} \in \mathcal{A} \text{ for any } A \in \mathcal{P}(V) \tag{4.1}$$

The condition (4.1) is often called *measurability condition*. So, in other words, a FRS is a measurable map from the given probability space $(\Omega, \mathcal{A}, P)$ to $\mathcal{P}(V)$, equipped with a σ-algebra on $\mathcal{P}(V)$. We often choose σ-algebra on $\mathcal{P}(V)$ is the discrete σ-algebra $\mathcal{E} = \mathcal{P}(\mathcal{P}(V))$. Clearly, a *finite random set* X is a *random element* when we refer to the *measurable space* $(\mathcal{P}(V), \mathcal{E})$. This is because $X^{-1}(\mathcal{E}) \subseteq \mathcal{A}$ since $\forall \mathbb{A} \in \mathcal{E}; X^{-1}(\mathbb{A}) = \cup_{A \in \mathbb{A}} X^{-1}(A)$.

4.2.2 Definition of Stochastic Pretopology

Definition 4.2. *We define **stochastic pseudo-closure** defined on $\Omega \times V$, any function **a(.,.)** from $\Omega \times \mathcal{P}(V)$ into $\mathcal{P}(V)$ such as:*

(P1): $a(\omega, \emptyset) = \emptyset \quad \forall \omega \in \Omega$;

(P2): $A \subset a(\omega, A) \quad \forall \omega \in \Omega, \forall A, A \subset V$;

(P3): $a(\omega, A)$ *is a finite random set* $\forall A, A \subset V$

$(\Omega \times V, a(.,.))$ *is then called Stochastic Pretopological space.*

By connecting the finite random set theory [144, 152], we can build a stochastic pseudo-closure function with different ways.

4.2.3 SP defined from random variables in metric space

By considering a *random ball* $B(x, \xi)$ with ξ is a non-negative random variable, we can build a *stochastic pseudo-closure* $\mathbf{a(.)}$ in metric space such as:

$$\forall A \in \mathcal{P}(V), \quad a(A) = \{x \in V | B(x, \xi) \cap A \neq \emptyset\} \tag{4.2}$$

The stochastic pseudo-closure $a(A)$ is a finite random set of all elements $y \in V$ such that y is within a distance of at most random radius ξ from at least one element of A.

4.2.4 SP defined from random variables in valued space

We present two ways to build stochastic pseudo-closure by extending the definition of pseudo-closure function on valued space presented in equation (1.20). Firstly, by considering threshold s is a *random variable* η, we can define a *stochastic pseudo-closure* $\mathbf{a(.)}$ such as:

$$\forall A \in \mathcal{P}(V), \quad a(A) = \{y \in V - A | \sum_{x \in A} \nu(x, y) \geq \eta\} \cup A \tag{4.3}$$

where threshold η is *random variable*.

Secondly, by considering the weight function $\nu(x, y)$ between two elements x, y as a *random variable*, we can define a *stochastic pseudo-closure* $\mathbf{a(.)}$ such as:

$$\forall A \in \mathcal{P}(V), \quad a(A) = \{y \in V - A | \sum_{x \in A} \nu_\Omega(x, y) \geq s\} \cup A \tag{4.4}$$

where $\nu_\Omega(x, y)$ is a *random variable*.

4.2.5 SP defined from a random relation built from a family of binary relations

Suppose we have a family $(R_i)_{i=1,\dots,m}$ of *binary reflexive relations* on a finite set V. We call $L = \{R_1, R_2, \dots, R_m\}$ is a set of relations. Let us define a *random relation* $R : \Omega \to L$ as a random variable:

$$P(R(\omega) = R_i) = p_i; \quad p_i \geq 0; \sum_{i=1}^{m} p_i = 1.$$

For each $x \in V$, we can build a random set of neighbors of x with random relation R:

$$\Gamma_{R(\omega)}(x) = \{y \in V | x \, R(\omega) \, y\}$$

We can define a *stochastic pseudo-closure* **a(.,.)** such as:

$$\forall A \in \mathcal{P}(V), \quad a(\omega, A) = \{x \in V | \Gamma_{R(\omega)}(x) \cap A \neq \emptyset\} \tag{4.5}$$

Remark: $\Gamma_{R(\omega)}(x)$ can be defined such as:

$$\Gamma_{R(\omega)}(x) = \bigcup_{i=1}^{m} (V_{R_i}(x) \cap I_{R_i}(\omega))$$

where $V_{R_i}(x) = \{y \in E | x \, R_i \, y\}$: set of neighbors of x with relation R_i and I_{R_i} is the *characteristic correspondence* with relation $R_i, i = 1, \dots, m$

$I_{R_i}(\omega) = \begin{cases} V & if \quad R(\omega) = R_i, \\ \emptyset & otherwise \end{cases}$

4.2.6 SP defined from a family of random relations

We can extend the previous work by considering many *random relations*. Suppose we have a family $(R_i)_{i=1,\dots,n}$ of random binary reflexive relations on a set V. For each $x \in V$, we can build a random set of neighbors of x with random relation $R_i, i = 1, 2, \dots, n$:

$$\Gamma_{R_i(\omega)}(x) = \{y \in V | x \, R_i(\omega) \, y\}$$

We can define a *stochastic pseudo-closure* $a(.,.)$ such as:

$$\forall A \in \mathcal{P}(V), \quad a(\omega, A) = \{x \in V | \forall i = 1, 2, \dots, n, \Gamma_{R_i(\omega)}(x) \cap A \neq \emptyset\} \tag{4.6}$$

4.2.7 SP defined from a random neighborhood function

Let us consider a random neighborhood function as a random set $\Gamma : \Omega \times V \to \mathcal{P}(V)$. $\Gamma(\omega, x)$ is a random set of neighborhoods of element x. We define a *stochastic pseudo-closure* **a(.,.)** as follows:

$$\forall A \in \mathcal{P}(V), \quad a(\omega, A) = A \cup (\bigcup_{x \in A} \Gamma(\omega, x)) \tag{4.7}$$

We have shown in this section how we construct stochastic pseudo-closure functions for various contexts. That is to say how proximity with randomness can be delivered to model complex neighborhoods formation in complex networks. In the two next sections, we will show how stochastic pretopology can be applied for modeling dynamic processes on complex networks by representing classical information diffusion models under stochastic pretopology language and then proposing *Pretopology Cascade Model* as a general information diffusion model in which complex random neighborhoods set can be captured by using stochastic pseudo-closure functions.

4.3 Stochastic pretopology as a general information diffusion model on single relational networks

Information diffusion has been widely studied in networks, aiming to model the spread of information among objects when they are connected with each other. In a single relational network, many diffusion models have been proposed such as *SIR* model, *tipping model*, *threshold models*, *cascade models*, etc [150, 67]. Please refer the work of [83] for a survey of information diffusion models in online social networks. In this section, we focus on *independent cascade* (IC) model [78] which is a generalized of SIR model, and another model is known as *linear threshold* (LT) model [81]. After describing properties of these models, we present in the following two scenarios in which stochastic pretopology as extensions of both IC model and LT model.

4.3.1 The Independent Cascade and Linear Threshold Models

4.3.1.1 Model Definitions

We assume a network $G = (V, \Gamma, W)$, where:

- V is a set of vertices.
- $\Gamma : V \to \mathcal{P}(V)$: neighborhood function.
  - $\Gamma(x)$ is set of outcoming neighborhoods of node x.
  - $\Gamma^{-1}(x)$ is set of incoming neighborhoods of node x.
- $W : V \times V \to \mathbb{R}$ is weight function.
  - in LT model, $W(x, y)$ is the weight of edge between two nodes x, y.
  - in IC model, $W(x, y)$ is the probability of node y infected from node x

The diffusion process occurs in discrete time steps t. If a node adopts a new behavior or idea, it becomes active, otherwise it is inactive. An inactive node has the ability to become active. The set of active nodes, newly active nodes at time t is considered as A_t, A_t^{new} respectively. The tendency of an inactive node x to become active is positively correlated with the number of its active incoming neighbors $\Gamma^{-1}(x)$. Also, we assume that each node can only switch from inactive state to active state, and an active node will remain active for the rest of the diffusion process. In general, we start with an initial seed set A_0 and through the diffusion process, for a given inactive node x, its active neighbors attempt to activate it. The process runs until no more activations occur.

4.3.1.2 Independent Cascade model

In IC model, there is a probability of infection associated with each edge. $W(x, y)$ is the probability of node x infecting y. This probability can be assigned based on the frequency of interactions, geographic proximity, or historical infection traces, ….. Each node, once infected, has the ability to infect its neighbor in the next time step based on the probability associated with that edge. At each time step t, each node $x \in A_{t-1}^{new}$ infects the inactive neighbors $y \in \Gamma(x)$ with a probability $W(x, y)$. The propagation continues until no more infection can occur (see algorithm 4.1).

Algorithm 4.1 Independent Cascade model

Require: Network $G = (V, \Gamma, W)$, seed set A_0
procedure RANDOM-IC-MODEL(G, A_0)
  $t \leftarrow 0, A^{total} \leftarrow A_0, A^{new} \leftarrow A_0$
  while infection occur **do**
    $t \leftarrow t + 1$; $A_t \leftarrow \emptyset$
    for u $\in A^{new}$ **do**
      $A_t(u) \leftarrow \{v^{inactive} \in \Gamma(u), q \leq W(u, v^{inactive})\}; \boldsymbol{q \sim U(0,1)}$
      $A_t \leftarrow A_t \bigcup A_t(u)$
    end for
    $A^{total} \leftarrow A^{total} \bigcup A_t$; $A^{new} \leftarrow A_t$
  end while
  return A^{total} ▷ Output
end procedure

Algorithm 4.2 Linear Threshold model

Require: Network $G = (V, \Gamma, W)$, seed set A_0
procedure RANDOM-LT-MODEL(G, A_0)
  $t \leftarrow 0, A^{total} \leftarrow A_0$
  while infection occur **do**
    $t \leftarrow t + 1$; $A_t \leftarrow \emptyset$
    for y $\in V - A^{total}$ **do**
      if $\sum_{x \in \Gamma^{-1}(y) \cap A_{t-1}} W(x, y) \geq \theta_y$, $\theta_y \sim U(0, 1)$ **then**
        A_t.append(y)
      end if
    end for
    $A^{total} \leftarrow A^{total} \bigcup A_t$
  end while
  return A^{total} ▷ Output
end procedure

4.3.1.3 Linear Threshold model:

In LT model, each directed edge (x, y) has a non-negative weight $W(x, y)$. For any node $y \in V$, the total incoming edge weights sum to less than or equal to one, i.e. $\sum_{x \in \Gamma^{-1}(y)} W(x, y) \leq 1$. An active node influences its inactive neighbors according to the weights. At each step, an inactive node y becomes active if the total weight of its incoming neighbors is at least threshold θ_y, $\theta_y \in [0, 1]$. The dynamics of the model are specified below.

Under the LT model, each node y selects a threshold θ_y in the interval [0,1] uniformly at random. Then, at each time step t where A_{t-1} is the set of nodes activated at time $t - 1$ or earlier, each inactive node y becomes active if $\sum_{x \in \Gamma^{-1}(y) \cap A_{t-1}} W(x, y) \geq \theta_y$. The propagation continues until no more infection can occur (see algorithm 4.2).

4.3.2 Stochastic pretopology as an extension of IC model

We can represent IC model by giving a definition of stochastic pretopology based on two definitions in subsection 4.2.4, 4.2.7: we firstly define a random set of actived nodes from each node $x \in A_{t-1}^{new}$ and then use a random neighbor function to define the random active nodes in the time t. Specifically, A_t^{new} is defined via two steps:

i. For each $x \in A_{t-1}^{new}$, set of actived nodes from x, $\Gamma^{(active)}(x)$, defined as:

$$\Gamma^{(active)}(x) = \{y \in \Gamma(x) | W(x,y) \geq \eta\}; \quad \eta \sim U(0,1) \tag{4.8}$$

ii. The set of newly active nodes, A_t^{new}, defined as:

$$A_t^{new} = a(A_{t-1}^{new}) - A_{t-1}; A_t = A_{t-1} \cup a(A_{t-1}^{new}) \tag{4.9}$$

where:

$$a(A_{t-1}^{new}) = A_{t-1}^{new} \bigcup (\bigcup_{x \in A_{t-1}^{new}} \Gamma^{(active)}(x)) \tag{4.10}$$

4.3.3 Stochastic pretopology as an extension of LT model

We can represent IC model by giving a definition of stochastic pretopology such as:

$$A_t = a(A_{t-1}) = \{y \in V - A_{t-1} | \sum_{x \in \Gamma^{-1}(y) \cap A_{t-1}} W(x,y) \geq \eta\} \cup A_{t-1}; \quad \eta \sim U(0,1)$$

4.4 Pretopology Cascade Models for modeling information diffusion on complex networks

4.4.1 Pretopological Cascade Model (PCM)

Most of information diffusion models are defined via node's neighbors. In general, at each time step t, the diffusion process can be described in two steps:

> *Step 1: define set of neighbors $N(A_{t-1})$ of set of active nodes A_{t-1}.*
>
> *Step 2: each element $x \in N(A_{t-1}) - A_{t-1}$ will be influenced by all elements in A_{t-1} to be active or not active node by following a diffusion rule.*

We consider the way to define set of neighbors $N(A_{t-1})$ in step 1. In classical diffusion model with complex network represented by a graph $G = (V, \Gamma)$, $N(A_{t-1})$ is often defined such as: $N(A_{t-1}) = \cup_{x \in A_{t-1}} \Gamma(x)$. By using the concepts of stochastic pretopology theory introducted in the section 4.2, the information diffusion process can be generalized by defining a set of neighbors $N(A_{t-1})$ as a *stochastic pseudo-closure* function $N(A_{t-1}) = a_\Omega(A_{t-1})$. We therefore propose the *Pretopological Cascade Model* presented in the following as a general information diffusion model which can be captured more complex random neighborhoods set in diffusion processes.

Definition 4.3. *Pretopological Cascade model:*
Under the Pretopological Cascade model (see algorithm 4.3), at each time step t, the diffusion process takes place in two steps:

> *Step 1: define set of neighbors $N(A_{t-1})$ of A_{t-1} as a stochastic pseudo-closure function $N(A_{t-1}) = a_\Omega(A_{t-1})$.*
>
> *Step 2: each element $x \in N(A_{t-1}) - A_{t-1}$ will be influenced by A_{t-1} to be active or not active node by following a "diffusion rule".*

For defining $N(A_{t-1})$ in step 1, we can apply different ways to define stochastic pseudo-closure function presented in section 4.2. "Diffusion rule" in step 2 can be chosen by various ways such as:

- Probability based rule: element x infects the inactive elements $y \in N(A_{t-1})$ with a probability $P_{x,y}$.

Algorithm 4.3 Pretopological Cascade model

Require: Set of elements V, stochastic pseudo-closure a_Ω, seed set A_0

procedure PRETOPO-CASCADE-MODEL(V, a_Ω, A_0)
  $t \leftarrow 0, A^{total} \leftarrow A_0$
  while infection occur **do**
    $t \leftarrow t + 1$; $A_t \leftarrow \emptyset$
    $N_t \leftarrow a_\Omega(A^{total})$ ▷ Compute random neighbors using a_Ω
    for u $\in N_t - A^{total}$ **do**
      if diffusion condition satisfied **then**
        A_t.append(u)
      end if
    end for
    $A^{total} \leftarrow A^{total} \bigcup A_t$
  end while
  return A^{total} ▷ Output
end procedure

- Threshold rule: inactive elements $y \in N(A_{t-1})$ will be actived if sum of all influence of all incoming elements of y greater than a threshold θ_y.

We present in the following two specific cases of the PCM: the first takes place in a stochastic graph by defining random neighbors sets based on nodes' degree distribution and the second takes place in multi-relational networks where random neighbors set is built from a family of relations.

4.4.2 PCM on Stochastic Graphs

4.4.2.1 Stochastic Graph

Definition 4.4. *(Stochastic Graph): A stochastic graph, which is denoted by* $G^\Omega = (V, \Gamma_\Omega)$ *is a pair consisting of a set* V *of vertices and a finite random set* Γ_Ω *mapping* $\Omega \times V$ *into* $\mathcal{P}(V)$.

The random neighbor function Γ_Ω in the definition 4.4 can be defined in a general way from finite random set theory [152]. Since the *nodes' degree distribution* is necessary for studying network structure, we propose here the way to defining the random neighbor function Γ_Ω via two steps:

1. Defining *probability law* of the cardinality of Γ_Ω (in fact, Γ_Ω is a degree distribution of network).
$$Prob(|\Gamma_\Omega| = k) = p_k \quad for \quad k = 1, 2, \ldots, \infty \tag{4.11}$$
2. Assigning *probability law* on V^k for $k = 1, 2, \ldots, \infty$
$$Prob(\Gamma_\Omega^{(1)} = x^{(1)}, \ldots, \Gamma_\Omega^{(k)} = x^{(k)} || \Gamma_\Omega| = k) \quad for \quad x^{(1)}, \ldots, x^{(k)} \in V \tag{4.12}$$

We can see some classical network models are specical cases of this kind of stochastic graph. For example, we have *Erdős–Rényi* model if $|\Gamma_\Omega| \sim U(0, 1)$ and *scale-free networks* model when $|\Gamma_\Omega|$ follows a *power-law distribution*. We also have other network models by using other probability distributions such as Poisson distribution, Geometry distribution, Binomial distribution, etc.

Algorithm 4.4 PCM on Stochastic Graph using power-law distribution

Require: Set $V = [1, 2, \ldots, N]$, stochastic pseudo-closure a_Ω, seed set A_0,
Require: powerlaw(a): power-law distribution with parameter c
Require: unidrnd(N,d): discrete uniform distribution
procedure PCM-STOCHASTIC-GRAPH(V, a, A_0)
  $t \leftarrow 0, A^{total} \leftarrow A_0, A^{new} \leftarrow A_0$
  while infection occur **do**
    $t \leftarrow t + 1$; $A_t \leftarrow \emptyset$
    for u $\in A^{new}$ **do**
      $d_u \leftarrow powerlaw(a)$ ▷ Compute degree of nodes u with power-law dis
      $N_u \leftarrow unidrnd(N, d_u)$ ▷ Compute d_u points randomly from set V
      $A_t(u) \leftarrow \{v^{inactive} \in N_u, q \leq P(u, v^{inactive})\}; \boldsymbol{q \sim U(0, 1)}$
      $A_t \leftarrow A_t \bigcup A_t(u)$
    end for
    $A^{total} \leftarrow A^{total} \bigcup A_t$; $A^{new} \leftarrow A_t$
  end while
  return A^{total} ▷ Output
end procedure

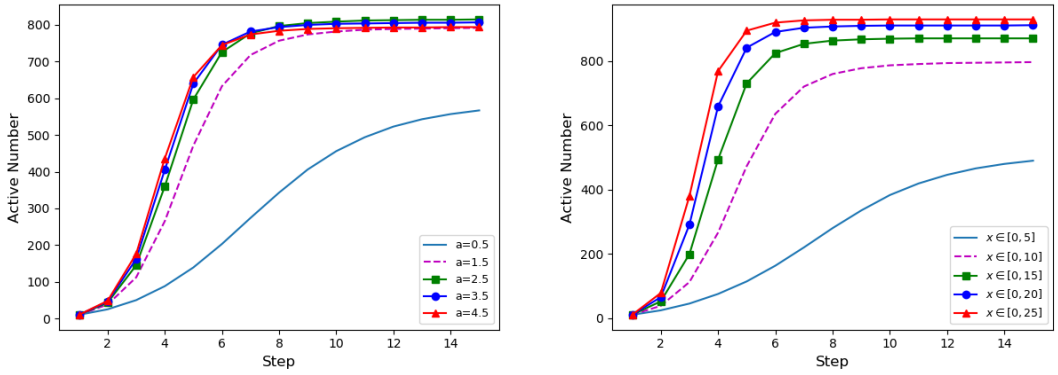
4.4.2.2 PCM on Stochastic Graph

Under the *PCM* on stochastic graph, at each time step t, each $x \in A_{t-1}$ generates a random number of neighbors η following a degree distribution given by the equation (4.11) and then generates random neighbors set $\Gamma_\Omega(x)$ following a point distribution given by the equation (4.12); after that x infects the inactive neighbors $y \in \Gamma_\Omega(x)$ with a probability $P_{x,y}$.

The algorithm 4.4 presents an illustration of PCM on Stochastic Graph using power-law distribution for choosing the number of neighborhoods of each node and discrete uniform distribution for choosing the target nodes.

4.4.2.3 Example

In this subsection, we illustrate the model presented in the algorithm 4.4. In this work, we set up the power-law distribution with the probability density function $pdf(x, a) = \frac{a}{x_{max}^a} x^{a-1}$ for $0 \leq x \leq x_{max}, a > 0$, number of nodes $N = |V| = 1000$ and number of initial active nodes $|A_0| = 10$. We examinate the spreading of the diffusion model with different values of two parameters a and x_{max} (the maximum of node's degree) of the power-law distribution (see Fig.4.1). For each pair of value of a and x_{max}, we run the model 1000 times and compute the average value of active number. Figure 4.1a represents the result of diffusion process with $x_{max} = 10$ and different values of parameter a. We can see the active mumbers in each step with parameter $a = 0.5$ is significantly lower than other cases. From step 1 to step 5, the greater value of parameter a, the higher value of number of active nodes but from step 6, this trend slightly changes in which the highest value falls in the case $a = 1.5$ and maintains the same level until step 15. For the role of the parameter x_{max}, figure 4.1b shows that the greater value of the maximum of node's degree x_{max}, the more nodes will be activated in the next step.



(a) Different values of a & $x_{max} = 10$ (b) Different values of x_{max} & $a = 1.5$

Figure 4.1: Diffusion process over time for different values of x_{max} and a with PCM on Stochastic Graph using power-law distribution.

Algorithm 4.5 PCM on Multi-Relational Network

Require: $G^{(multi)} = (V, \Gamma = [\Gamma_1, \Gamma_2, \ldots, \Gamma_m])$, seed set A_0,
Require: random-index(m,p): random index distribution; $p = [p_1, p_2, \ldots, p_m]$.

procedure PCM-Multi-Rels-Net(V, p, A_0)
  $t \leftarrow 0, A^{total} \leftarrow A_0, A^{new} \leftarrow A_0$
  while infection occur **do**
    $t \leftarrow t + 1$; $A_t \leftarrow \emptyset$
    for u $\in A^{new}$ **do**
      $\eta \leftarrow random - index(m, p)$ ▷ Compute random index for choosing relation
      $N_u^{\eta} \leftarrow \Gamma[\eta](u)$ ▷ Compute neighbors of u with relation η
      $A_t(u) \leftarrow \{v^{inactive} \in N_u^{\eta}, q \leq P(u, v^{inactive})\}; \boldsymbol{q \sim U(0,1)}$
      $A_t \leftarrow A_t \bigcup A_t(u)$
    end for
    $A^{total} \leftarrow A^{total} \bigcup A_t$; $A^{new} \leftarrow A_t$
  end while
  return A^{total} ▷ Output
end procedure

4.4.3 PCM on Multi-Relational Networks

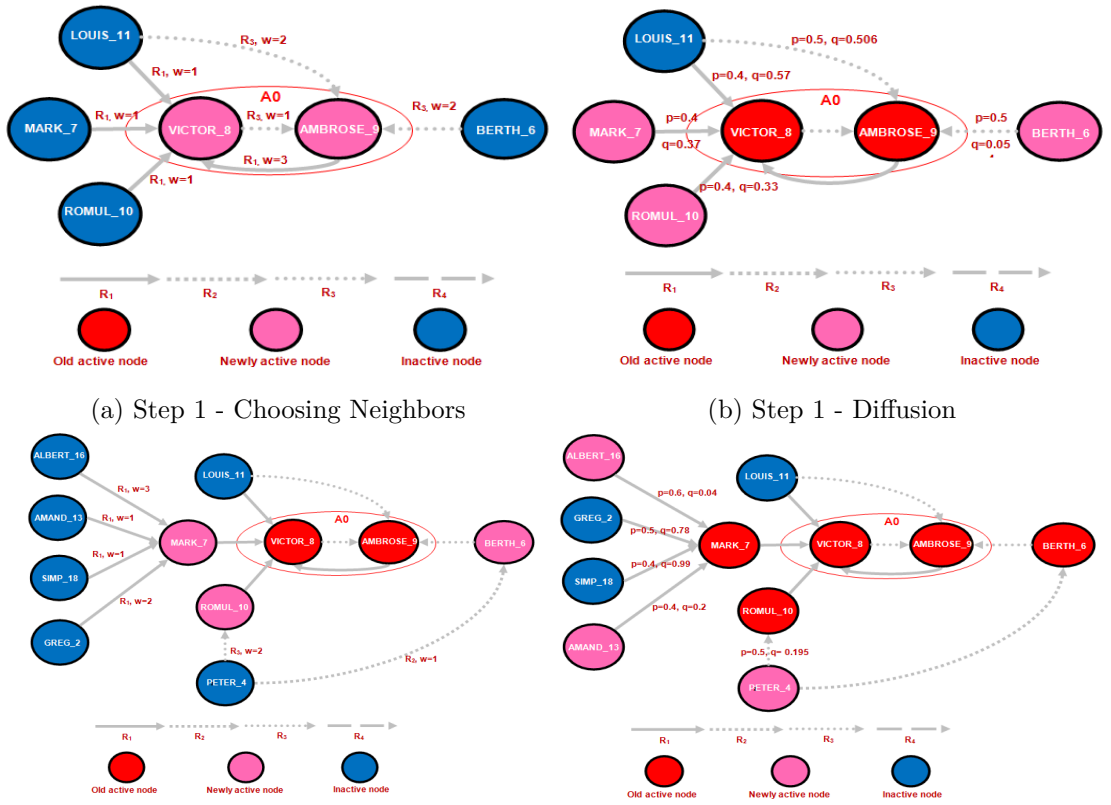
4.4.3.1 Multi-relational network

A *multi-relational network* can be represented as a multi-graph, which allows multiple edges between node-pairs. A *multi-relational network*, which is denoted by $G^{(multi)} = (V, (\Gamma_1, \Gamma_2, \ldots, \Gamma_m))$, is a pair consisting of a set V of vertices and a set of multivalued functions $(\Gamma_i)_{i=1,2,\ldots,m}$ mapping V into 2^V. Γ_i is a neighbor function following the relation R_i.

4.4.3.2 Defining Random neighbors set on Multi-relational network

Let us define a random index η takes values on $\{1, 2, \ldots, m\}$ such as a random variable:

$$P(\eta = i) = p_i; i = 1, 2, \ldots, m; \quad p_i \geq 0; \sum_{i=1}^{m} p_i = 1 \tag{4.13}$$



(a) Step 1 - Choosing Neighbors

(b) Step 1 - Diffusion

(c) Step 2 - Choosing Neighbors

(d) Step 2 - Diffusion

Figure 4.2: Example of PCM on Multi-Relational Network: Step 1 and 2

We define a random neighbor function Γ_η based on random index η such as: $\Gamma_\eta = \Gamma_i$ if $\eta = i$. For each $x \in V$, we can build a random set of neighbors of x: $\Gamma_\eta(x) = \Gamma_i(x)$ if $\eta = i, i = 1, 2, \ldots, m$.

4.4.3.3 PCM on Multi-Relational Network

Under the*PCM* on multi-relational networks, at each time step t, each $x \in A_{t-1}$ generates a random index η given by the equation (4.13) then generates random neighbors set $\Gamma_\eta(x)$; after that x infects the inactive neighbors $y \in \Gamma_\eta(x)$ with a probability $P_{x,y}$.

The PCM on multi-relational networks is presented in algorithm 4.5. We can also extend this model by choosing randomly a set $S_\eta \subset \{1, 2, \ldots, m\}$ and then using interset or union operator to generate a random set of neighbors of x. For example, we can define $\Gamma_\eta(x) = \cup_{k_i \in S_\eta} \Gamma_{k_i}(x)$ or $\Gamma_\eta(x) = \cap_{k_i \in S_\eta} \Gamma_{k_i}(x)$.

4.4.3.4 Example

In this subsection, we illustrate the model on a small multi-relational network built from the *Sampson Monastery* dataset presented in the subsection 1.6.2. In this work, we define multi-relational network with 4 relations: *like*, *esteem*, *influence*, *praise* and apply *PCM-Multi-Rels-Net* algorithm to show how information diffusion process takes place in this multi-relational network. We present in the following, the latter steps in more details.

Setup Firstly, we created multi-relational network with 18 nodes represented to the member and nodes connect together with 4 relations. Since, each member ranked only

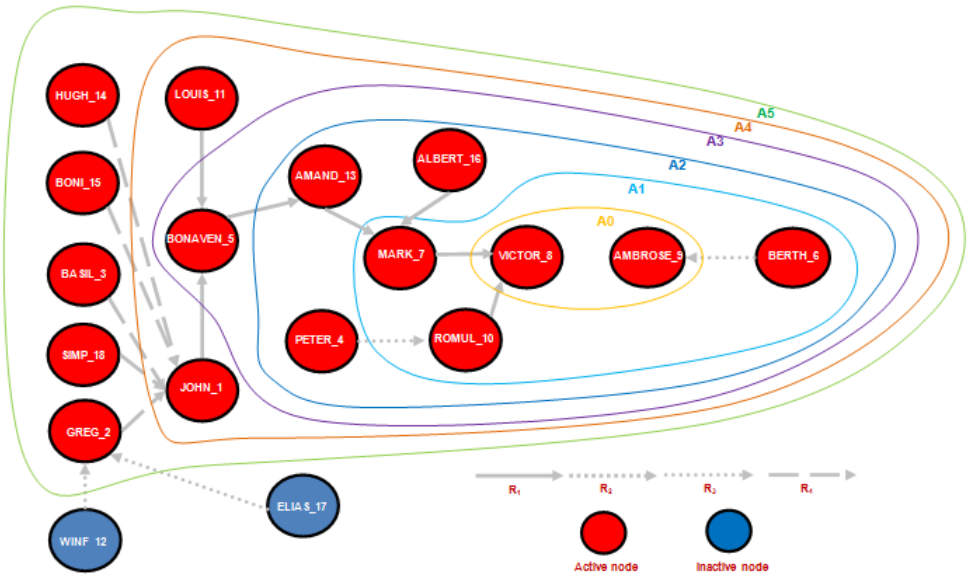


Figure 4.3: Example of PCM on Multi-Relational Network: the result of diffusion process

his top three choices on that tie, edges are nonsymmetric, weighted with three integer values, from 1 to 3 and labeled with R_1, R_2, R_3, R_4 for *like*, *esteem*, *influence* and *praise* relation respectively. We then defined *random index distribution* η such as:

η	R_1	R_2	R_3	R_4	$\sum$
Prob	0.35	0.3	0.2	0.15	1

Since each member ranked only his top three choices on that tie, it is better if we define the probability of diffusion between two nodes based on the weight of the edge instead of choosing randomly. Specifically, we assigned propagation probability is 0.4, 0.5, 0.6 if the weight of the edge is 1, 2. 3 respectively.

We begin the diffusion process with a seed set A_0. At each time step t, each newly active node generates a random neighbors set based on random index distribution and infects the inactive nodes in this set with a probability based on the weight of the edge.

Result The two first steps of this example is shown in Fig.4.2. Newly active nodes are shown in pink color shapes while red color shapes represent for old active nodes. At the initial time, two nodes VICTOR_8 and AMBROSE_9 are activated. At step 1, node VICTOR_8 and AMBROSE_9 chooses randomly one relation for diffusion by generating random index η and has a chance to activate their neighbors following the chosen relation. As we can see in Fig.4.2a, VICTOR_8 chooses relation R_1 (*like* relation) and has change to active LOUIS_11, MARK_7, ROMUL_10 (AMBROSE_9 is already activated)) while AMBROSE_9 chooses relation R_3 (*influence* relation) has change to active BERTH_6 and also LOUIS_11 (VICTOR_8 is also already activated). According to Fig.4.2b, only three nodes MARK_7, ROMUL_10, and BERTH_6 are successfully activated and the initial active nodes change to red color (denoting it stays active but no chance to activate others). In the next time step, set of newly active nodes [MARK_7, ROMUL_10, BERTH_6] has change to active ALBERT_16, AMAND_13, SIMP_18, GREG_2, PETER_4 (Fig.4.2c) and only ALBERT_16, AMAND_13, PETER_4 are successfully activated (Fig.4.2d). The propagation process continues until there is no more new active node. Figure 4.3 shows the result of the diffusion process.

4.5 Conclusion

In this work, we proposed *Stochastic Pretopology* as a general mathematical framework for complex networks analysis which can not only deal with uncontrolled factors but also work with a set as a whole entity, not as a combination of elements. We have shown how we construct stochastic pseudo-closure functions to model complex neighborhoods formation in complex networks in various contexts. Furthermore, we showed how stochastic pretopology can be applied for modeling dynamic processes on complex networks by proposing *Pretopology Cascade Model* as a general information diffusion model in which complex random neighborhoods set can be captured by using *stochastic pseudo-closure* functions. We also illustrated the proposed models by presenting the algorithms and running some experiments. In future works can be developed a software library for implementing stochastic pseudo-closure functions, pretopology cascade diffusion algorithms and applying the proposed models for real-world complex networks.

Chapter 5

Dynamic Textual Social Network Analysis Using Topic Modeling

In this chapter, we proposed an agent-based model for analyzing dynamic social network associated to textual information using author-topic model, namely *Textual-ABM*. Author-topic model is chosen to estimate topic's distribution transformation of agents in the agent-based model since it models the content of documents and also interests of authors. *Textual-ABM* can be utilized to discover dynamic of a social network which includes not only network structure but also node's properties over time.

Furthermore, we introduced homophily independent cascade model based on textual information, namely *Textual-Homo-IC*. This model based on standard independent cascade model; however, we exploited the aspect of infected probability estimation relied on homophily. Particularly, homophily is measured based on textual content by utilizing topic modeling. The process of propagation takes place on agent's network where each agent represents a node. In addition to expressing the *Textual-Homo-IC* model on the static network, we also revealed it on dynamic agent's network where there is not only transformation of the structure but also the node's properties during the spreading process. We conducted experiments on two collected data sets from NIPS and a social network platform-Twitter and have attained satisfactory results.

The result of this chapter has been published in the *Proceedings of I4CS 2018. Communications in Computer and Information Science (CCIS)*, 2018, ISBN: 978-3-319-93407-5, pages: 47–62 [88] and *Proceedings of ICCCI 2018. Lecture Notes in Computer Science (LNCS)* [90].

5.1 Introduction

Social networking research has attracted a lot of attention of researchers with appearance of science fields including social network analysis (SNA) [80, 155], community detection [77, 151] and so on. However, in reality, social networks are always in a state of fluctuations which are difficult to model. Therefore, the analytical tendencies shift from the study of the static to the analysis of the dynamic of the social network. Recently, there are two major approaches for analyzing the dynamic concept in social networks which comprise the fluctuation of structure and the characteristic variation of nodes over time. The first approach is the analysis that a dynamic network has been considered as a cumulation of snapshot networks [118] or the concept-temporal networks [94]. In another hand, dynamic social network analyzes [33] concentrated on exploiting aspect that properties of nodes may transform over time since they have the ability to learn and adapt to the interactive process instead of static nodes in SNA. Agent-based modeling is often used to explore how network evolve. In this study, we will analyze social

networking dynamically with a combination of these two approaches.

Majority research from two approaches above hardly mentioned about the content of messages among users since they often focus on structure [94, 118] or just propose an agent-based model for simulating dynamic social network without based on content [50]. Therefore, in this study, we utilize textual content in the interactive process of users with the purpose of analyzing dynamic on both network's structure and user's interest. Recently, there are various methods for textual mining, for instance, Latent Dirichlet Allocation (LDA) [133], Author-Topic Model (ATM) [164], etc. We choose the author-topic model to define user's interest since it simultaneously models the content of documents and the interests of authors while LDA only considers a document as a mixture of probabilistic topics, not take into account author's interests.

In this chapter, we construct an agent-based model for analyzing dynamic social network associated with the textual information, namely *Textual-ABM*. The dynamic of a social network is demonstrated through the fluctuation of *Textual-ABM* including agent's network structure and agent's characteristics. Additionally, we propose a homophily independent cascade model based on textual information, namely *Textual-Homo-IC*. Independent cascade (IC) [78] model is spreading model in which there is a probability of infection associated with each edge. The probability $P_{(u,v)}$ is the probability of u infecting v. This probability can be assigned based on the frequency of interactions, geographic proximity, or historical infection traces. In this study, the infected probability between two users is measured by similarity or homophily based on textual information by utilizing topic modeling. The spreading process is performed on agent's network where each node is represented by an agent. *Textual-Homo-IC* is demonstrated on static agent's network and dynamic agent's network in which the network structure and characteristics of agents have remained during propagation process for the former while there is a variation for the later. Some experiments are implemented on co-author network and Twitter with the combination of two methods LDA and ATM for estimating topic's distribution of users and two distance measurements Hellinger distance and Jensen-Shannon distance for measuring homophily. On the static networks, the results demonstrated that the effectiveness of *Textual-Homo-IC* outperforms comparison with random diffusion. Additionally, our results also illustrated the fluctuation of active number for the diffusion process on a dynamic network instead of attaining and remaining stable state on a static network.

The structure of this chapter is organized as follows: section 5.2 reviews backgrounds; the *Textual-ABM* model is proposed in section 5.3; section 5.4 presents independent cascade model based on homophily, namely *Textual-Homo-IC*; section 5.5 demonstrates experiments while the results and evaluation is presented in section 5.6 and we conclude our work in section 5.7.

5.2 Preliminaries

5.2.1 Agent-Based Model

An agent-based model (ABM) is a class of computational models for simulating the actions and interactions of autonomous agents. ABM has been used in many fields including biology, ecology and social science [153]. There are three major elements in ABM including agents, their environment, and mechanisms of interaction among agents. Firstly, agents are heterogeneous entities which comprise diverse characteristics and behaviors. Secondly, agent's environment is a space that plays responsibility for reflecting the structure of the overall system and supplying agents their perceptions and enabling their actions. Thirdly, interaction is a form of information exchange among

agents which resulted in perception and behavior. Particularly, the essence of an ABM is the dynamics of the global system emerges from the local interactions among its composing parts.

5.2.2 Author-Topic Model (ATM)

Topic models work by defining a probabilistic representation of the latent structure of corpora through latent factors called topics, which are commonly associated with distributions over words [25]. LDA [26] is one of the most used topic models to discover complex semantic structure in the NLP area. In LDA, each document is may be considered as a mixture of different topics and each topic is characterized by a probability distribution over a finite vocabulary of words.

However, one limitation of LDA is that it does not model authors explicitly. This led us to present ATM [163], a generative model for documents that extends LDA [26], can model documents as if they were generated by a two-stage stochastic process. Each author is represented by a probability distribution over topics, and each topic is represented as a probability distribution over words for that topic. The words in a multi-author paper are assumed to be the result of a mixture of each authors' topic mixture. The topic-word and author-topic distributions are learned from data in an unsupervised manner using a Markov chain Monte Carlo algorithm.

The generative model of ATM is described with a graphical model in Figure 5.1, proceeds as follows:

1. For each author a=1, ..., A choose $\theta_a \sim$ Dirichlet(α)
   For each topic t=1,..T choose $\phi_t \sim$ Dirichlet(β)

2. For each document d=1,.., D

   2.1. Given the vector of authors a_d

   2.2. For each word i=1,...,N_d

      2.2.1. Choose an author $x_{di} \sim$ Uniform(a_d)

      2.2.2. Choose a topic $z_{di} \sim$ Discrete($\theta_{x_{di}}$)

      2.2.3. Choose a word $w_{di} \sim$ Discrete($\phi_{z_{di}}$)

5.2.3 Update Process of LDA and ATM

LDA and ATM can be updated with additional documents after training has been finished. This update procedure is executed by Expectation Maximization (EM)-iterating over new corpus until the topics converge. The two models are then merged in proportion to the number of old and new documents. On the other hand, for stationary input (mean that no appearance of new topics in new documents), this process which is equal to the online training of Hoffman [91, 92]. In ATM, if the update process is called with authors that already exist in the model, it will resume training on not only new documents for that author but also the previously seen documents.

Recently, some packages were provided for topic modeling such as *topicmodels* or *lda* in R, or *Gensim* [1] in Python. In this study, we choose Gensim for training and updating the author-topic model and LDA.

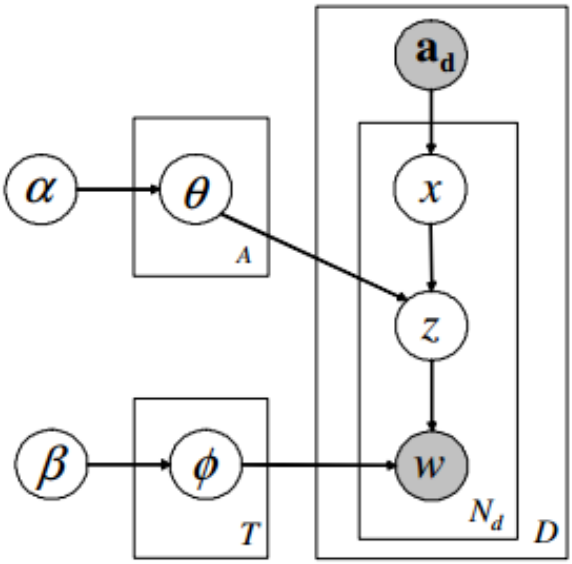


Figure 5.1: The graphical model for the author-topic model using plate notation.

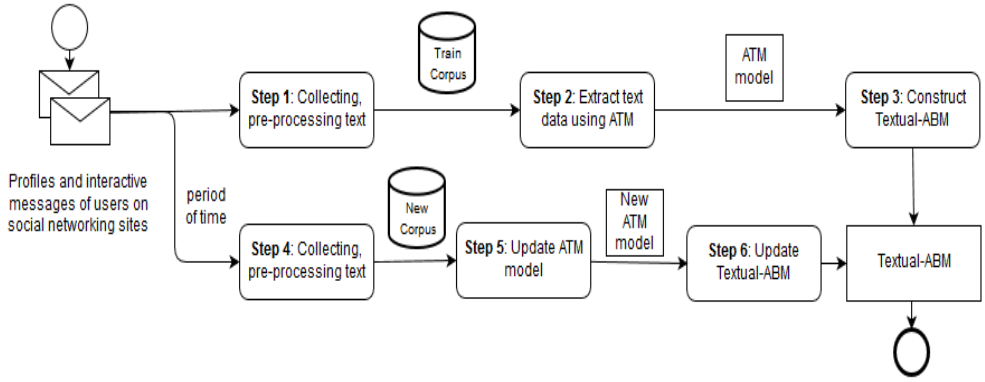


Figure 5.2: *Textual-ABM* for analyzing dynamic social network using ATM

5.3 Agent-based model for analyzing dynamic social network associated with textual information

In this section, we constructed an agent-based model for analyzing dynamic in social network associated with the textual information that we call *Textual-ABM* (see Figure 5.2). We demonstrated dynamic of the social network through the fluctuation of the *Textual-ABM* in which an agent represent for a network's node. The local interplay among agents transform global environment which includes not only agent's network structure but also system's resource, lead to agent's topic distribution transformation. We present in the following, the steps to build the model and update process for the model in more details.

5.3.1 Text collection and pre-processing

Firstly, we crawl text data from social networking sites or blogs by APIs such as Twitter API, Facebook Graph API and so on. After collecting the data, several pre-processing steps are performed for data cleaning. The textual content obtained is also processed by

[1]https://pypi.python.org/pypi/gensim

stemming the words, removing stop-words and numeric symbol. Finally, pre-processed data will be saved into train corpus.

5.3.2 Topic modeling with ATM

After obtaining train corpus from text collection and pre-processing, we apply the author-topic model to define topic's distribution of users. The outputs of ATM contain two matrices: The author-topics distribution matrix θ and the topic-words distribution matrix ϕ. The topic-terms distribution matrix $\phi \in R^{KxV}$ consists of K rows, where the i-th row $\phi_i \in R^V$ is the words distribution of topic i. Similarly, the author-topics distributions matrix $\theta \in R^{NxK}$ consists of N rows, where the i-th row $\theta_i \in R^K$ is the topics distribution for author i. A high probability value of θ_{ij} indicates that author i interested in topic j.

5.3.3 *Textual-ABM* Construction

Based on the results of topic modeling, we construct a *Textual-ABM* with the following fundamental elements:

5.3.3.1 Agent

In our model, agents who represent for users in social network, are heterogeneous with numerous specific characteristics including *ID*, *Name*, *ID-List* (list of agent's id who have ever interacted), particularly *Corpus* (collection of texts that an agent used to interact with other agents) and *TP-Dis* (topic's probability distribution). *Corpus* will be cumulated over time through the interactive process. Besides, *TP-Dis* which illustrates for agent's interest on topics.

5.3.3.2 Agent interaction

In this study, we take into account directed interaction among agents related to textual information, for instance, retweet or reply on Twitter, sharing statuses on Facebook, collaborating to write papers and so on. Based on the amount of text information of all agents over the interactive process, we not only estimate topic's distribution of agents at a certain time but also their dynamic over time.

5.3.3.3 Agent's network

In this study, we consider an agent's network structure in which a node corresponding to an agent and two kinds of relations among agents: directed interaction relation (R_{DI}) and major topic relation (R_{MTP}). R_{DI} is formed since two agents have interacted directly with each other as described in subsubsection *Agent interaction*, while R_{MTP} appears when two agents are interested in the same topic with probability greater than threshold p_0. It can be said that the structural dynamic of network results from agent's interaction since there is the appearance of new agents, more interactions among agents who have interacted, or new interplay. Moreover, nodes in the network are agents that can change their properties over time, lead to the transformation of similarity or homophily among them.

5.3.3.4 Global environment

Global environment is considered as a space including *agents*, *interaction among agents* and *resource*. Particularly, we emphasize on the system's resource related to textual in-

Table 5.1: Topics-Author distribution θ over three periods

User ID 0

Topic	Prob.		
	Period		
	1	2	3
2	**0.542**	**0.397**	**0.397**
3	**0.277**	**0.412**	**0.412**
1	**0.175**	**0.187**	**0.187**
4	0.003	0.002	0.002
0	0.003	0.002	0.002

User ID 1

Topic	Prob.		
	Period		
	1	2	3
3	0.991	0.231	0.231
0	0.002	0.015	0.015
4	0.002	0.001	0.001
1	0.002	0.001	0.001
2	0.002	0.752	0.752

User ID 4

Topic	Prob.		
	Period		
	1	2	3
1	**0.919**	**0.919**	**0.003**
0	**0.02**	**0.02**	**0.491**
2	**0.02**	**0.02**	**0.501**
4	0.02	0.02	0.003
3	0.02	0.02	0.003

User ID 7

Topic	Prob.		
	Period		
	1	2	3
2	0.968	0.977	0.983
4	0.008	0.006	0.004
3	0.008	0.006	0.004
0	0.008	0.006	0.004
1	0.008	0.006	0.004

formation incorporating a *system's corpus* and a *its generative model* in which the former is aggregated from all agent's corpus while *ATM* is used for the later correspondingly.

5.3.4 Update process for *Textual-ABM*

In our model, we consider three principal environmental processes with the purpose of updating the *Textual-ABM* after a certain period of interaction among agents. Firstly, the collection process of new textual data and pre-processing will be performed for generating of a new corpus. Next, the *system's corpus* and the existed generative model *ATM* will be updated with the new corpus. Finally, the *Textual-ABM* will be updated with the appearance of new agents and agent's characteristic transformation including *ID-List*, *Corpus* and *TP-Dis*. We demonstrated steps for generation of the *Textual-ABM* and its updating process at figure 5.2 in which the former is formed from step 1 to step 3 while between step 4 and step 6 for the later respectively.

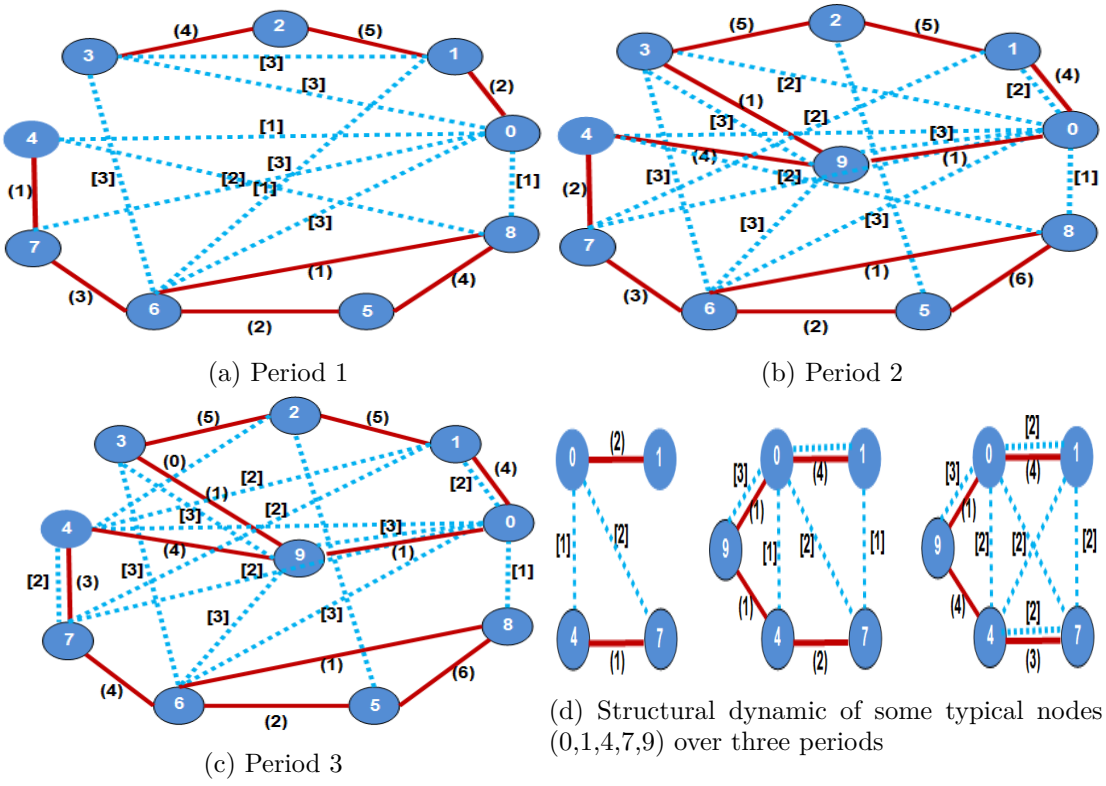
5.3.5 Toy example

In this section, we constructed a *Textual-ABM* for simulating a dynamic network that contains *10* users (is identified with an ID from *0* to *9*) of theguardian.com [2]. For purpose of illustrating dynamic of user's interest on the small number of topics, comment

[2]https://www.theguardian.com/international

collection and pre-processing are conducted from a political blog *"Resentful Americans turn a blind eye to Trump's faults"* [3] in *theguardian.com*. Firstly, train corpus is collected from *4* pm to *6* pm. A *Textual-ABM* is constructed as soon as estimating topic's distribution of users from train corpus. We considered an agent's network with two kinds of relationships which include R_{DI} and R_{MTP} with $p_0 = 0.1$. Furthermore, to reveal dynamic of *Textual-ABM*, we conducted updated processes with new corpus in next two periods in which the second period lasted from *6* pm to *7* pm and after *7* pm for the third period.

On one hand, the dynamic of the *Textual-ABM* is demonstrated through the significant variation of agent's network structure (see Figure 5.3) over three stages. The transformation from first to the second period is demonstrated with the appearance of new agent *9*, new interactions such as *(0,9)* or *(4, 9)*, more interactions between *4* and *7* or between *0* and *1*, particularly occurrence R_{MTP} such as topic [2] between *0* and *1*. Besides, it is the notable appearance of R_{MTP} between the second stage and the third stage, for instance, *(4,7)* are interested in topic [2]. On other hand, the dynamic of



(a) Period 1

(b) Period 2

(c) Period 3

(d) Structural dynamic of some typical nodes (0,1,4,7,9) over three periods

Figure 5.3: Structural dynamic of agent's network over three periods: dashed red line with label in symbol '[]' illustrate R_{MTP} with major topic; solid blue line with label without symbol '[]' reveal for R_{DI} with the number of interactions

the *Textual-ABM* is also expressed through agent's interest transformation since ATM model is updated in the next two periods. We illustrated topic distributions of four representative agents θ through three stages in Table 5.1. We can see that agent *ID 0* originally is interested in three topics *2,3* and *1*. However, there is a significant variation from the first stage to the second stage, and maintaining this state to the third stage. In contrast, agent *ID 4* remain interested in first two stages, but there is a drastic change in the third stage. Hence, these fluctuations are the results from the agent's interaction

[3] https://www.theguardian.com/us-news/blog/2016/aug/25/resentful-americans-turn-blind-eye-donald-trump

since the interaction of agent *ID 0* to each other is shown mainly in the second period while agent *ID 4* in the third period.

In summary, *Textual-ABM* can be utilized for analyzing a dynamic social network in which users communicate with each other through text information. The dynamic is not only illustrated by network structure but also fluctuation of agent's interest.

5.4 Homophily independent cascade model based on textual information

In recent years, research on the process of information diffusion through social networks has attracted the attention of researchers with applications in various fields including computer science, economy, and biology. Information propagation has been extensively researched in networks, with the objective of observing the information spreading among objects when they are connected with each other. Recently, there are numerous diffusion models which have been proposed including linear threshold (LT) [81, 173], independent cascade (IC) model [78] and so on.

The IC model has been used extensively since it is the simplest cascade model and is successful at explaining diffusion phenomena in social networks [78]. In IC, each edge is associated with a probability of infection independently which is usually assigned by a uniform distribution [98, 99, 100]. Besides, this probability can also be estimated based on interactive frequency, geographical distance, or infected vestige in history [173].

Nevertheless, perhaps in the fact that the infected probability from one object to another depends on similarity or homophily among them, for instance, the probability for two scientists in the common field incorporate to write a paper higher compared with the different field. In another instance, a user A on Twitter is easy to *'follow'* user B when A have common interests with B. Therefore, in this section, we discover the aspect of infected probability estimation based on similarity or homophily.

In this section, we proposed homophily independent cascade model based on textual information, namely *Textual-Homo-IC*. This model based on standard independent cascade model; however, we exploited the aspect of infected probability estimation relied on homophily. Particularly, homophily is measured based on textual content by utilizing topic modeling. The process of propagation takes place on agent's network where each agent represents a node. In addition to expressing the *Textual-Homo-IC* model on the static network, we also revealed it on dynamic agent's network where there is not only transformation of the structure but also the node's properties during the spreading process. We conducted experiments on two collected data sets from NIPS and a social network platform-Twitter and have attained satisfactory results.

5.4.1 Homophily measure

Homophily is the tendency of individuals to associate with similar others [112, 139]. There are two principal approaches to measure homophily including the first one based on a single characteristic and the combination of multiple features for the second. For the first approach, homophily is classified into two types including status homophily and value homophily in which the former refers to the similarity in socio-demographic traits, such as race, age, gender, etc while the similarity in internal states for the later, such as opinions, attitudes, and beliefs [112, 139].

Besides, Laniado et al. analyzed the presence of gender homophily in relationships on the Tuenti Spanish social network [110]. On other hands, with the second approach, Aiello et al. [6] discovered homophily from the context of tags of social networks including Flickr, Last.fm, and aNobii. Each user u is presented under a vector $\overrightarrow{w}_u$ whose

elements correspond to tags and w_{ut} is tag frequencies. Standard cosine similarity is utilized to compare the tag feature vectors of two users as a formula for estimating their homophily. However, the principal drawback of this method is the high dimensionality as a result of the high number of unique tags. Additionally, Cardoso et al.[49] explored homophily from hashtags on Twitter. Each user is represented by a feature vector $\overrightarrow{u}$ in which u_i illustrate the interested extent of u on topic i. u_i is estimated based on the number of hashtags belonging to topic i. Consine distance is also utilized to measure the similarity between two users.

However, in general, these methods have not exploited the textual information related to users yet while it contains significant information for similarity analysis, for instance, based on content of papers, we can define whether the authors research in the same narrow subject or not, or we can determine which are common interests between two users on Twitter based on their tweets. For that reason, we propose a method of homophily measurement based on textual content. A fundamental technology for text mining is *Vector Space Model* (VSM) [166] where each document is represented by word-frequency vector.

Nevertheless, two principal drawbacks of VSM are the high dimensionality as a result of the high number of unique terms in text corpora and insufficient to capture all semantics. Therefore, topic modeling was proposed to solve these issues. Recently, there are dissimilar methods of topic modeling which include *LDA* [26], *Author-Topic Model* (ATM) [163], etc. In this study, we chose LDA and ATM to estimate topic's probability distribution of users.

In this study, we estimate homophily between two agents based on their topic's probability distribution. If we consider a probability distribution as a vector, we can choose some distances measures related to the vector distance such as Euclidean distance, Cosine Similarity, Jaccard Coefficient, etc. However, experimental results in our previous work [44] (section 3.5, chapter 3) demonstrated that it is better if we choose distances measures related to the probability distribution such as Kullback-Leibler Divergence, Jensen-Shannon divergence, Hellinger distance, etc. In this study, we chose Hellinger distance and Jensen-Shannon divergence (see section 3.5, chapter 3) to measure distance. The **homophily** is measured based on the distance between two two probability distributions, $d(P, Q)$, by the following equation:

$$\mathbf{Homo(u,v) = 1 - d(P, Q)} \tag{5.1}$$

5.4.2 Agent's network

In this study, the network that we take into account for spreading process is agent's network $G(V, E)$ in which V is set of agents represented for nodes and E is set of edges among nodes. Agents are heterogeneous with three principal properties including *ID*, *Neighbors* and *TP-Dis* (topic's probability distribution). LDA and ATM can be utilized to estimate *TP-Dis* of users.

To demonstrate the dynamic of agent's network, we exploit not only the structure of network but also agent's properties. Firstly, the structure of the network can be transformed with the appearance of new agents or new connections. Moreover, topic's distribution of agents can fluctuate since agents own more text information through the interactive process. The problem is to how to update the transformation of topic's distribution of users after a time period based on existing topic modeling. In LDA, to make an estimate of topic's distribution of users, we consider each user correspond to each document. Therefore, we can not utilize update mechanism of LDA to update topic's distribution of users when users have more documents in interaction. Instead of unusable update mechanism of LDA, we can make use of ATM to estimate topic's

Algorithm 5.1 Random-IC on static agent's network

Require: agent's network G=(V, E), I_0: seed set
procedure RANDOM-IC-STATIC-NETWORK(G, I_0)
  $t \leftarrow 0, I^{total} \leftarrow I_0, I^{newest} \leftarrow I_0$
  while infection occur **do**
    $t \leftarrow t + 1$; $I_t \leftarrow \emptyset$
    for u $\in I^{newest}$ **do**
      $I_t(u) \leftarrow \{v^{inactive} \in \eta^{out}(u), \boldsymbol{p <= q}\}; \boldsymbol{p, q \sim U(0, 1)}$
      $I_t \leftarrow I_t \bigcup I_t(u)$
    end for
    $I^{total} \leftarrow I^{total} \bigcup I_t$; $I^{newest} \leftarrow I_t$
  end while
  return I^{total} ▷ Output
end procedure

Algorithm 5.2 Textual-Homo-IC on static agent's network

Require: agent's network G=(V, E), I_0: seed set
procedure TEXTUAL-HOMO-IC-STATIC-NETWORK(G, I_0)
  $t \leftarrow 0, I^{total} \leftarrow I_0, I^{newest} \leftarrow I_0$
  while infection occur **do**
    $t \leftarrow t + 1$; $I_t \leftarrow \emptyset$
    for u $\in I^{newest}$ **do**
      $I_t(u) \leftarrow \{v^{inactive} \in \eta^{out}(u), \boldsymbol{p <= Homo(u, v)}\}; \boldsymbol{p \sim U(0, 1)}$
      $I_t \leftarrow I_t \bigcup I_t(u)$
    end for
    $I^{total} \leftarrow I^{total} \bigcup I_t$; $I^{newest} \leftarrow I_t$
  end while
  return I^{total} ▷ Output
end procedure

distribution of users and simultaneously update mechanism to update user's topic's distribution since each author can own various documents.

5.4.3 Random-IC on static agent's network

In this section, we illustrate IC model on a static network in which infected probability based on uniform distribution, namely *Random-IC*. This model plays as a benchmark model for comparing performance with *Textual-Homo-IC* model that we will propose at section 5.4.4. At each step t where I^{newest} is the set of the newly active nodes at time $t-1$, each $u \in I^{newest}$ infects the inactive neighbors $v \in \eta^{out}(u)$ with a probability $P(u, v)$ randomly. The propagation continues until no more infection can occur (see Algorithm 5.1).

5.4.4 *Textual-Homo-IC* on static agent's network

Propagation mechanism of *Textual-Homo-IC* on static agent's network is similar to *Random-IC*, but the difference is that each active agent $u \in I^{newest}$ infects the inactive neighbors $v \in \eta^{out}(u)$ with a probability $P(u, v)$ equal $Homophily(u, v)$ instead of a random probability (see Algorithm 5.2)

5.4.5 *Textual-Homo-IC* on dynamic agent's network

Although IC model on the dynamic network has been researched in [75, 189], the dynamic concept of a network has only been considered under the structure transformation while the activated probability from an active node to inactive another is always fixed during spreading process. Therefore, we propose *Textual-Homo-IC* model on a dynamic agent's network in which not only discover the variation of network's structure but also agent's topics distribution. It can be said that infected probability among agents can change over time because of their homophily transformation.

There is the resemblance in the propagation mechanism of *Textual-Homo-IC* on the dynamic network in comparison with the static network; however, in spreading process at step $t \in C$, agent's network G will be updated as shown in the section 5.4.2 (see Algorithm 5.3).

Algorithm 5.3 Textual-Homo-IC on dynamic agent's network

Require: agent's network $G = (V, E)$; I_0: seed set
Require: $C = \{k_1, k_2, ..., k_n\}$, at step k_i G is updated; n: number steps of diffusion
procedure TEXTUAL-HOMO-IC-DYNAMIC-NETWORK(G, I_0, C)
  $t \leftarrow 0, I^{total} = I_0, I^{newest} = I_0$
  while $t < n$ **do** ▷ $(n > max\{C\})$
    $t \leftarrow t + 1; I_t = \emptyset$
    if $t \in C$ **then**:
      Update G; $I^{newest} = I^{total}$
    end if
    loop for u $\in I^{newest}$
      Begin
        $I_t(u) \leftarrow \{v^{inactive} \in \eta^{out}(u), \boldsymbol{p <= Homo(u, v)}\}; \boldsymbol{p \sim U(0, 1)}$
        $I_t \leftarrow I_t \bigcup I_t(u)$
      End
    $I^{total} = I^{total} \bigcup I_t$; $I^{newest} = I_t$
  end while
  return I^{total} ▷ Output
end procedure

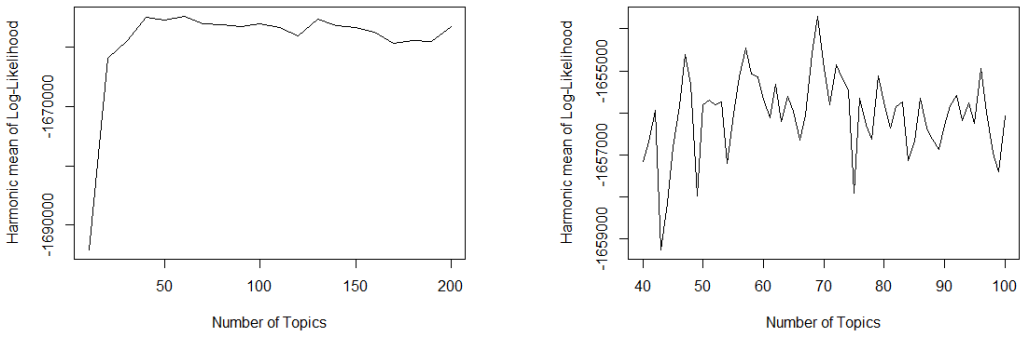
5.5 Experiments

5.5.1 Data Collection

The proposed *Textual-Homo-IC* models have been tested on a well-known social network platform-Twitter and co-author network. For the Twitter network, we have aimed to 1524 users in which links are "*follow*" relations. We crawled 100 tweets for each user and textual data stretched from 2011 to April 2018. For co-author network, we have targeted authors who have participated in Neural Information Processing Systems Conference (NIPS) from 2000 to 2012. The dataset contains 1740 papers which are contributed by 2479 scientists.

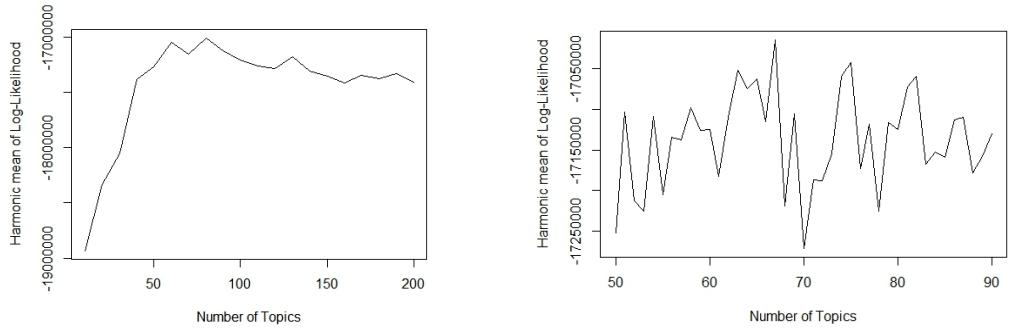
5.5.2 Setup

Firstly, we defined the number of topic for the whole corpus based on the Harmonic mean of Log-Likelihood (HLK) [47]. We calculated HLK with the number of topics in the range [10, 200] with sequence 10. We realized that the best number of topics is in



(a) ♯ topics in range [10, 200], seq 10 (b) ♯ topics in range [40, 100], seq 1

Figure 5.4: Log-likelihood for Twitter network



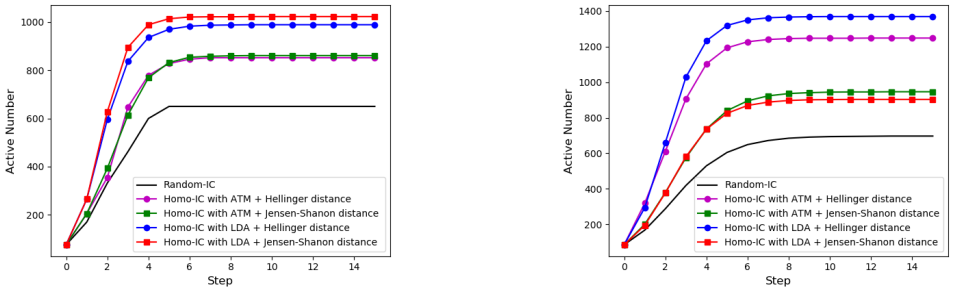
(a) ♯ topics in range [10, 200], seq 10 (b) ♯ topics in range [50, 90], seq 1

Figure 5.5: Log-likelihood for Co-author network

the range [40, 100] for Twitter network (Fig.5.4a) and [50, 90] (Fig.5.5a) for co-author network. Therefore, we ran HLK again with sequence 1 and obtained the best is 69 for Twitter network (Fig.5.4b) and 67 for co-author network (Fig.5.5b).

Textual-Homo-IC diffusion is implemented on the static Twitter network and co-author network. Agent's networks which are constructed as shown in section 5.4.2 in which "*follow*" relation for Twitter network and "*co-author*" relation for co-author network. For each network, we implemented four experiments of *Textual-Homo-IC* with combination two methods of estimating topic's distribution (LDA and ATM) and two kinds of distance measurements (Hellinger distance and Jensen-Shanon distance). Besides, we also conducted *Random-IC* as a benchmark to compare the performance with *Textual-Homo-IC*.

To simulate *Textual-Homo-IC* on dynamic agent's network, we conducted experiments on the dynamic Twitter network and co-author network. For co-author network, we collected textual data between 2000 to 2009 for train corpus and estimating the author's topic distribution using ATM. An agent's network is formed with "*co-author*" relation. In another hand, for the Twitter network, textual data is gathered from 2011 to January 2018 for train corpus. Unfortunately, it is impossible to get the exact date that a user starts to follow another on Twitter, including in the API or Twitter's web interface. This leads to the inability to express the fluctuations in network structure with "*follow*" relation. Therefore, we took into account agent's network with "*major topic*" relation (R_{MTP}) which will appear when two agents are interested in the common



(a) Twitter network (b) Co-author network

Figure 5.6: Textual-Homo-IC diffusion on static networks

topic with a probability greater than threshold p_0. In this study, we considered R_{MTP} with $p_0 = 0.1$.

The diffusion process on dynamic network starts as soon as an agent's network is formed. For each kind of distance measurement, we implemented four experiments in which the first one is the propagation on agent's network without dynamic. The last investigations are that after every 5, 10 and 15 steps of diffusion agent's network will fluctuate once follow mechanism presented in section 5.4.2. Agent's networks will be updated in 3 times corresponding to each month from February to April 2018 for Twitter network and each year from 2010 to 2012 for co-author network.

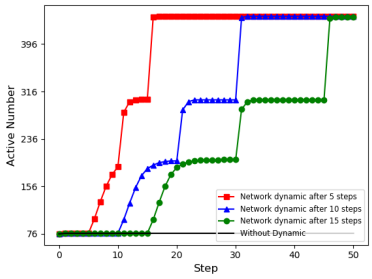
5.5.3 Model Evaluation

To evaluate the performance of diffusion models, we can use the number of active nodes or the active percentage which are standard metrics in information diffusion field [75, 189]. In this research, we utilize the active number to evaluate the performance of spreading models. We compare the performance of proposed *Textual-Homo-IC* diffusion model with baseline model (*Random-IC*).

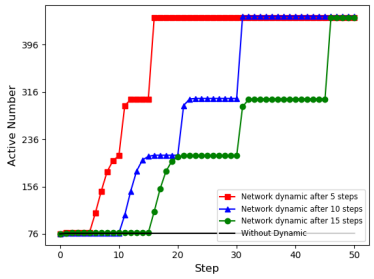
5.6 Results and Evaluation

5.6.1 Compare *Textual-Homo-IC* and *Random-IC* diffusion

The results of *Textual-Homo-IC* on static agent's networks are shown in Figure 5.6. For both networks, we can see that the active number of *Textual-Homo-IC* is always greater than *Random-IC* in four cases which are the combination of two methods of topic modeling and two distance measurements. Firstly, in the Twitter network (Figure 5.6a), the number of active agents reaches approximately 650 for *Random-IC* diffusion while *Textual-Homo-IC* attains about 862 for both cases where ATM combine with two distances. Particularly, *Textual-Homo-IC* that incorporate LDA with Hellinger distance and Jensen-Shanon distance obtain the higher number of active agents in comparison with cases utilizing ATM, about 989 and 1023 active agents respectively. On the other hand, in co-author network (Figure 5.6b), the number of active agents reaches approximately 700 for *Random-IC* diffusion while *Textual-Homo-IC* attains about 903 for the case using LDA combined with Jensen-Shanon distance. Besides, 946 active agents are reached by collaborating ATM and Jensen-Shanon distance. In addition, *Textual-Homo-IC* with ATM and Hellinger distance obtains approximately 1248 active agents while the highest number belongs to *Textual-Homo-IC* with LDA and Hellinger distance, around 1369 active agents. In summary, we can conclude that *Textual-Homo-IC*

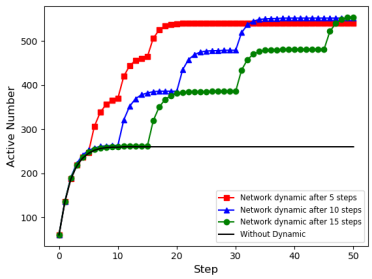


(a) ATM and Hellinger distance

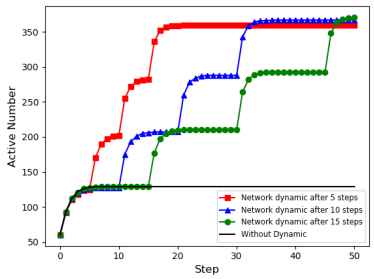


(b) ATM and Jensen-Shanon distance

Figure 5.7: Textual-Homo-IC diffusion on dynamic Twitter network



(a) ATM and Hellinger distance



(b) ATM and Jensen-Shanon distance

Figure 5.8: Textual-Homo-IC diffusion on dynamic co-author network

diffusion outperforms in comparison with *Random-IC*.

5.6.2 *Textual-Homo-IC* diffusion on dynamic agent's network

Results are shown in Figure 5.7 and 5.8 which illustrate *Textual-Homo-IC* diffusion on the dynamic Twitter network and co-author network respectively. For the Twitter network, there is only one agent that be activated from the seed set on the static network for both cases of distance measurements. The reason is the number of connection with R_{MTP} in the initial stage is too low to diffusion. However, if there is the network's transformation in the next 3 stages with the arrival of many new connections, there is a significant increase in the active number. For co-author network, *Textual-Homo-IC* on a static network reaches a steady state from the 12th step and 7th step onwards for using Hellinger and Jensen-Shanon distance respectively. However, if there is the network's fluctuation in the next 3 stages, the active number increase significantly. In short, we can conclude that the propagation process without dynamic of network reached and maintained the steady state while there is a significant transformation in the active number if agent's network has fluctuation in the diffusion process.

5.7 Conclusion

In this chapter, we constructed an agent-based model for analyzing dynamic social network associated textual information using author-topic model, namely *Textual-ABM*. The benefit of this model is to can be utilized to observe dynamic of a social network which includes not only network structure but also node's properties. Moreover, we proposed *Textual-Homo-IC* model, an expanded model of independent cascade diffusion model based on homophily which is measured based on textual information by utilizing

topic modeling. *Textual-Homo-IC* has been revealed details on both static and dynamic agent's network. Experimental results demonstrated that the effectiveness of *Textual-Homo-IC* on the static network outperforms *Random-IC*. In addition, experiments also illustrated the fluctuation of the active number on dynamic agent's network instead of obtaining and remaining a steady state in a static network. In future works, we will conduct experiments on other large-scale networks and compare proposed model with more other baseline models.

Conclusion and Future Works

We have started this thesis with an overview of pretopology theory. The reason for that is quite concrete: pretopology can be generalized topology space for modeling the concept of proximity. Consequently, *pseudo-closure* function with its flexibility, can follow step by step the growth of a set, then it can be applied to solve some problems in complex systems. Connecting with Topic modeling, we investigated in this dissertation two applications in complex systems: text mining and complex networks analysis.

Pretopology and Topic Modeling for Text Mining

For analysis of textual data such as clustering or classification, we proposed a nouvel approach by intergrating Topic Modeling such as LDA, ss-LDA, ATM to clustering or classification algorithms which can solve two principal drawbacks of VSM are the high dimensionality and insufficient to capture all semantics. In this approach, we investigated two aspects: choosing the "good" distance measure and clustering or classifying with multi-criteria.

Firstly, since we have used the topic distributions extracted from Topic modeling as an input for clustering algorithms such as k-means, we compared the effect of eight distance measures represented to eight distance measure families categorized by [52]. Our experiments demonstrated the fact that the efficiency of Probabilistic-based measurement clustering is better than the Vector-based measurement clustering including Euclidean distance. Comparing among *LDA+k-means*, *LDA+Naive*, *VSM*, the experiments also showed that if we choose the suitable value of a number of topic for LDA and PBM for k-means, *LDA+k-means* can improve the effect of clustering results.

Secondly, in the case where several objectives have to be taken into account, the "multi-criteria" aspect is therefore included in the clustering. Our methods can effectively solve this problem by representing each criterion by a binary relation. Pseudo-closure function built from these relations can be used to define a distance measure that we called *pseudo-closure distance*. This distance can be used in clustering algorithms such as k-means for multi-criteria clustering. More details, in our proposed method MCPTM, we clustered documents by using two criteria: one based on the major topic of document (qualitative criterion) and the other based on Hellinger distance (quantitative criterion).

For document classification, we presented the ss-LDA, an LDA version for learning process. The supervision of the process is within two levels: word level and document level. By connecting the ss-LDA and Random Forest classifier, we proposed a new methodology for a multilayer soft classification for web pages and obtained a good classification rate.

We also proposed a distributed version of LDA that implemented in Spark. The main idea of the proposed algorithm is to make local copies of the parameters across the processors and synchronize the global counts matrices that represent the coefficients of LDA mixtures. We showed empirically with a set of experimentations that our parallel

implementation with Spark has the same predictive power as the sequential version and has a considerable speedup. We finally document an analysis of the scalability of our implementation and the super-linearity that we obtained.

Pretopology and Topic Modeling for Complex Network Analysis

To deal with uncontrolled factors that often occur in complex networks, we proposed *Stochastic Pretopology* as a general mathematical framework for complex systems analysis. We illustrated our approach by introducing various ways to define a stochastic pseudo-closure function in many situations and showed how this approach can generalize graph, random graph, multi-relational networks, etc.

Most of information diffusion models are defined via node's neighbors. The neighbors of a node v in a graph G is the subgraph of G induced by all nodes adjacent to v. The question here is how to build a diffusion model that can capture more complex neighbor selection mechanisms in complex networks? To answer this question, we proposed a solution that presented in chapter 4 by proposing *Pretopology Cascade Model*, a general information diffusion model which can not only apply on different kinds of complex networks such as stochastic graphs, multi-relational networks but also can capture more complex neighbor set by using the flexible mechanism of stochastic pseudo-closure function. This model also allowed to work with a set as a whole entity, not as a combination of elements.

Related to the textual complex networks, we constructed an agent-based model for analyzing dynamic social network associated textual information using author-topic model, namely *Textual-ABM* and proposed independent cascade diffusion model based on homophily which is measured based on textual information by utilizing topic modeling, namely *Textual-Homo-IC*. The spreading process happens on agent's network where each agent corresponds to a node. Particularly, we discovered the dynamic aspect of a network which is not only the transformation of structure but also node's properties. Therefore, infected probability can fluctuate over time because of the variation of homophily. *Textual-Homo-IC* has been revealed details on both static agent's network and dynamic agent's network. Experimental results demonstrated that the effectiveness of *Textual-Homo-IC* on the static network outperforms Random-IC. In addition, experiments also illustrated the fluctuation in the active number of the spreading in dynamic agent's network instead of obtaining and remaining a steady state in a static network.

Future work

In future work, there are several directions we would like to explore:

- For MCPTM, we have presented our contribution by applying it on microblogging posts and have obtained good results. In the future, we would like to test these results on large scale and more conventional benchmark datasets.

- Related to the application of LDA for complex networks, we also intend to implement a streaming distributed version of LDA where the documents will be processed as they are crawled from the internet in general or social media in particular. We then improve our work presented in chapter 5 by constructing an agent-based model for analyzing online dynamic social network associated textual information.

- A point not discussed in chapter 4 which can be seen as a perspective is practical aspects of the *Pretopological Cascade model* proposed model. In future works can be developed a software library for implementing stochastic pretopology algorithms and applying the proposed model for real-world complex systems.
- Concerning the pretopology library, we continue to complete and perfect the functions so that it becomes more powerful.

Bibliography

[1] *mertterzihan/pymc, https://github.com/mertterzihan/pymc, 2015-07-02.*

[2] C. C. Aggarwal and C. Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.

[3] M. Ahat, S. B. Amor, M. Bui, M. Lamure, and M.-F. Courel. Pollution Modeling and Simulation with Multi-Agent and Pretopology. In J. Zhou, editor, *Complex Sciences, First International Conference, Complex 2009, Shanghai, China, February 23-25, 2009. Revised Papers, Part 1*, volume 4 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 225–231. Springer, 2009.

[4] M. Ahat, B. Amor S., M. Bui, S. Jhean-Larose, and G. Denhiere. Document Classification with LSA and Pretopology. *Studia Informatica Universalis*, 8(1), 2010.

[5] A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. J. Smola. Scalable inference in latent variable models. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 123–132. ACM, 2012.

[6] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer. Friendship Prediction and Homophily in Social Media. *ACM Trans. Web*, 6(2):9:1–9:33, June 2012.

[7] S. M. Ali and S. D. Silvey. A General Class of Coefficients of Divergence of One Distribution from Another. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1):131–142, 1966.

[8] G. M. Amdahl. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities. In *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, AFIPS '67 (Spring), pages 483–485, New York, NY, USA, 1967. ACM.

[9] S. B. Amor. *Percolation, prétopologie et multialéatoires,contributions à la modélisation des systèmes complexes : exemple du contrôle aérien.* Thèse de doctorat, Ecole Pratique des Hautes Etudes, 2008.

[10] S. B. Amor and M. Bui. Généralisation des processus de percolation discrets. *Stud. Inform. Univ.*, 7(1):78–93, 2009.

[11] S. B. Amor, M. Bui, and M. Lamure. Modeling urban aerial pollution using stochastic pretopology. *Africa Mathematics Annals*, 1(1):7–19, 2010.

[12] S. B. Amor, V. Levorato, and I. Lavallée. Generalized Percolation Processes Using Pretopology Theory. In *2007 IEEE International Conference on Research, Innovation and Vision for the Future in Computing & Communication Technologies, RIVF 2007, Hanoi, Vietnam, 5-9 March 2007*, pages 130–134. IEEE, 2007.

[13] M. Archoun. *Modélisation prétopologique de la segmentation par croissance de régions des images à niveau de gris.* Thèse de doctorat, Université Lyon 1, 1983.

[14] G.-M. Arnaud, M. Lamure, M. Terrenoire, and D. Tounissoux. Analysis of the connectivity of an object in a binary image: a pretopological approach. In *Proc.of the 8th IAPR Conference*, 1986.

[15] Arvind and D. E. Culler. Dataflow Architectures. *Annual Review of Computer Science*, 1(1):225–253, 1986.

[16] J.-P. Auray. *Contribution à l'étude des structures pauvres.* Thèse d'Etat, Université Lyon 1, 1982.

[17] J.-P. Auray. Structures pauvres. *Stud. Inform. Univ.*, 7(1):94–130, 2009.

[18] J.-P. Auray, M. Brissaud, and G. Duru. Les apports de la prétopologie. In *112e Congrès national des sociétés savantes*, volume IV, pages 15–29. Sciences fasc, 1987.

[19] J.-P. Auray, G. Duru, and M. Mougeot. A pretopological analysis of input output model. *Economics letter*, 2(4), 1979.

[20] C. Basileu. *Modélisation structurelle des réseaux sociaux : application à un système d'aide à la décision en cas de crise sanitaire.* Thèse de doctorat, Lyon 1, Dec. 2011.

[21] C. Basileu, S. B. Amor, M. Bui, and M. Lamure. Prétopologie stochastique et réseaux complexes. *Stud. Inform. Univ.*, 10(2):73–138, 2012.

[22] Z. Belmandt. *Manuel de prétopologie et ses applications.* Hermès, 1993.

[23] Z. Belmandt. *Basics of Pretopology.* Hermann, 2011.

[24] C. Berge. *The Theory of Graphs.* Courier Corporation, 1962.

[25] D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[26] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[27] S. Bonnevay. *Extraction de caractéristiques de texture par codages des extrema de gris et traitement prétopologique des images.* PhD thesis, Université Lyon 1, Lyon, Oct. 1997.

[28] S. Bonnevay. Pretopological operators for gray-level image analysis. *Stud. Inform. Univ.*, 7(1):173–195, 2009.

[29] S. Bonnevay, M. Lamure, C. Largeron-Leténo, and N. Nicoloyannis. A pretopological approach for structuring data in non-metric spaces. *Electronic Notes in Discrete Mathematics*, 2:1–9, 1999.

[30] S. Bonnevay and C. Largeron. Data analysis based on minimal closed subsets. In *The international federation of Classification Societies*, pages 303–308, Namur, 2000.

[31] M. Bouayad. *Prétopologie et reconnaissance des formes.* Thèse de doctorat, INSA, Lyon, 1998.

[32] M. Boubou. *Contribution aux méthodes de classification non supervisée via des approches prétopologiques et d'agrégation d'opinions*. phdthesis, Université Claude Bernard - Lyon I, Nov. 2007.

[33] R. L. Breiger, K. M. Carley, and P. Pattison. *Dynamic Social Network Modeling and Analysis: workshop summary and papers*. National Academies Press, Washington, D.C., 2003.

[34] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Aug. 1996.

[35] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, Oct. 2001.

[36] M. Brissaud. Les espaces prétopologiques. *Compte-rendu de l'Académie des Sciences*, 280(A):705–708, 1975.

[37] M. Brissaud. Espaces prétopologiques généralisés et application: Connexités, Compacité, Espaces préférenciés généraux. In *URA 394*, Lyon, 1986.

[38] M. Brissaud. Analyse prétopologique du recouvrement d'un référentiel. Connexités et point fixe. In *XXIIIe colloque Structures économiques et économétrie*, Lyon, 1991.

[39] M. Brissaud. Adhérence et acceptabilité multicritères. Analyse prétopologique. In *XXIVème colloque Structures économiques et économétrie*, Lyon, 1992.

[40] M. Brissaud. Retour sur les origines de la prétopologie. *Stud. Inform. Univ.*, 7(1):5–23, 2009.

[41] M. Brissaud, J.-P. Auray, G. Duru, M. Lamure, and C. Siani. Eléments de prétopologie généralisée. *Stud. Inform. Univ.*, 7(1):45–77, 2009.

[42] M. Bui, S. B. Amor, M. Lamure, and C. Basileu. Gesture Trajectories Modeling Using Quasipseudometrics and Pre-topology for Its Evaluation. In A. Laurent, O. Strauss, B. Bouchon-Meunier, and R. R. Yager, editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 15th International Conference, IPMU 2014, Montpellier, France, July 15-19, 2014, Proceedings, Part II*, volume 443 of *Communications in Computer and Information Science*, pages 116–134. Springer, 2014.

[43] Q. V. Bui, S. B. Amor, and M. Bui. Stochastic Pretopology as a Tool for Topological Analysis of Complex Systems. In N. T. Nguyen, D. H. Hoang, T.-P. Hong, H. Pham, and B. Trawinski, editors, *Intelligent Information and Database Systems - 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018, Proceedings, Part II*, volume 10752 of *Lecture Notes in Computer Science*, pages 102–111. Springer, 2018.

[44] Q. V. Bui, K. Sayadi, S. B. Amor, and M. Bui. Combining Latent Dirichlet Allocation and K-Means for Documents Clustering: Effect of Probabilistic Based Distance Measures. In *Intelligent Information and Database Systems - 9th Asian Conference, ACIIDS 2017, Kanazawa, Japan, April 3-5, 2017, Proceedings, Part I*, pages 248–257, 2017.

[45] Q. V. Bui, K. Sayadi, and M. Bui. A multi-criteria document clustering method based on topic modeling and pseudoclosure function. In *Proceedings of the Sixth International Symposium on Information and Communication Technology, Hue City, Vietnam, December 3-4, 2015*, pages 38–45, 2015.

[46] Q. V. Bui, K. Sayadi, and M. Bui. A Multi-Criteria Document Clustering Method Based on Topic Modeling and Pseudoclosure Function. *Informatica*, 40(2):169–180, July 2016.

[47] W. Buntine. Estimating Likelihoods for Topic Models. In *Proceedings of the 1st Asian Conference on Machine Learning: Advances in Machine Learning*, ACML '09, pages 51–64, Berlin, Heidelberg, 2009. Springer-Verlag.

[48] D. Buscaldi, G. Dias, V. Levorato, and C. Largeron. QASSIT: A Pretopological Framework for the Automatic Construction of Lexical Taxonomies from Raw Texts. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 955–959. The Association for Computer Linguistics, 2015.

[49] F. M. Cardoso, S. Meloni, A. Santanche, and Y. Moreno. Topical homophily in online social systems. *arXiv:1707.06525 [physics]*, July 2017.

[50] K. M. Carley, M. K. Martin, and B. R. Hirshman. The etiology of social change. *Topics in Cognitive Science*, 1(4):621–650, Oct. 2009.

[51] E. Cech. *Topological Spaces*. John Wiley and Sons, New York, NY, USA, 1966.

[52] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES*, 1(4):300–307, 2007.

[53] G. Cleuziou, D. Buscaldi, V. Levorato, and G. Dias. A pretopological framework for the automatic construction of lexical-semantic structures from texts. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 2453–2456. ACM, 2011.

[54] A. Criminisi, J. Shotton, and E. Konukoglu. Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. *Foundations and Trends\textbackslashr m in Computer Graphics and Vision: Vol. 7: No 2-3, pp 81-227*, 2012.

[55] I. Csiszár. Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizitat von Markoffschen Ketten. *Magyar. Tud. Akad. Mat. Kutato Int. Kozl*, 8:85–108, 1963.

[56] M. Dalud-Vincent. *Modèle Prétopologique pour une méthodologie d'analyse de réseaux: concepts et algorithmes*. Thèse de doctorat, Université Lyon 1, 1994.

[57] M. Dalud-Vincent, M. Brissaud, and M. Lamure. Pretopology as an extension of graph theory : the case of strong connectivity. *International Journal of Applied Mathematics*, 5(4):455–472, Apr. 2001.

[58] M. Dalud-Vincent, M. Brissaud, and M. Lamure. Closed sets and closures in pretopology. *International Journal of Pure and Applied Mathematics*, pages 391–402, Jan. 2009.

[59] M. Dalud-Vincent, M. Brissaud, and M. Lamure. Connectivities and Partitions in a Pretopological Space. *International Mathematical Forum*, 6(45):2201–2215, Mar. 2011.

[60] M. Dalud-Vincent, M. Brissaud, M. Lamure, and R. G. Paradin. Pretopology, matroïdes and hypergraphs. *International Journal of Pure and Applied Mathematics*, 67(4):363–375, 2011.

[61] R. Dapoigny, M. Lamure, and N. Nicoloyannis. Pretopological Transformations of Binary Images: A Parallel Implementation. In M. H. Hamza, editor, *Proceedings of the Seventh IASTED/ISMM International Conference on Parallel and Distributed Computing and Systems, Washington, D.C., USA, October 19-21, 1995*, pages 288–291. IASTED/ACTA Press, 1995.

[62] J. Debayle and J.-C. Pinoli. General Adaptive Neighborhood-Based Pretopological Image Filtering. *Journal of Mathematical Imaging and Vision*, 41(3):210–221, 2011.

[63] M.-M. Deza and E. Deza. *Dictionary of distances.* Elsevier, 2006.

[64] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2Nd Edition).* Wiley-Interscience, New York, NY, USA, 2000.

[65] G. Duru. Nouveaux éléments de prétopologie. Technical report, Faculté de Droit et des Sciences économiques de Besançon, 1977.

[66] G. Duru. *Contribution à l'étude des structures des systèmes complexes dans les Sciences Humaines.* Thèse d'État, Université Lyon 1, 1980.

[67] D. EASLEY and J. KLEINBERG. *Networks Crowds and_Markets.* Cambridge University Press, 2010.

[68] M. Egea. Prétopologie floues. *Stud. Inform. Univ.*, 7(1):131–171, 2009.

[69] H. Emptoz. *Modèles prétopologiques pour la reconnaissance des formes. Application en Neurophysiologie.* Thèse d'Etat, Université Lyon 1, 1983.

[70] A. Ferligoj and V. Batagelj. Direct multicriteria clustering algorithms. *Journal of Classification*, 9(1):43–61, 1992.

[71] S. Fouchal, M. Ahat, I. Lavallée, M. Bui, and S. B. Amor. Clustering Based on Kolmogorov Information. In R. Setchi, I. Jordanov, R. J. Howlett, and L. C. Jain, editors, *Knowledge-Based and Intelligent Information and Engineering Systems - 14th International Conference, KES 2010, Cardiff, UK, September 8-10, 2010, Proceedings, Part I*, volume 6276 of *Lecture Notes in Computer Science*, pages 452–460. Springer, 2010.

[72] M. Fréchet. *Espaces Abstraits.* Hermann, 1928.

[73] C. Frélicot and H. Emptoz. A Pretopological Approach for Pattern Classification with Reject Options. In A. Amin, D. Dori, P. Pudil, and H. Freeman, editors, *Advances in Pattern Recognition, Joint IAPR International Workshops SSPR '98 and SPR '98, Sydney, NSW, Australia, August 11-13, 1998, Proceedings*, volume 1451 of *Lecture Notes in Computer Science*, pages 707–715. Springer, 1998.

[74] C. Frélicot and F. Lebourgeois. A pretopology-based supervised pattern classifier. In A. K. Jain, S. Venkatesh, and B. C. Lovell, editors, *Fourteenth International Conference on Pattern Recognition, ICPR 1998, Brisbane, Australia, 16-20 August, 1998*, pages 106–109. IEEE Computer Society, 1998.

[75] N. T. Gayraud, E. Pitoura, and P. Tsaparas. Diffusion Maximization in Evolving Social Networks. In *Proceedings of the 2015 ACM on Conference on Online Social Networks*, COSN '15, pages 125–135, New York, NY, USA, 2015. ACM.

[76] J. Gil-Aluja and A. M. G. Lafuente. *Towards an Advanced Modelling of Complex Economic Phenomena - Pretopological and Topological Uncertainty Research Tools*, volume 276 of *Studies in Fuzziness and Soft Computing*. Springer, 2012.

[77] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, pages 7821–7826, June 2002.

[78] J. Goldenberg, B. Libai, and E. Muller. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, 12(3):211–223, Aug. 2001.

[79] A. Gordon. *Classification, 2nd Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press, 1999.

[80] M. Grandjean. A social network analysis of Twitter: Mapping the digital humanities community. *Cogent Arts & Humanities*, 3(1):1171458, Dec. 2016.

[81] M. Granovetter. Threshold Models of Collective Behavior. *American Journal of Sociology*, 83(6):1420–1443, May 1978.

[82] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

[83] A. Guille, H. Hacid, C. Favre, and D. A. Zighed. Information diffusion in online social networks: a survey. *ACM SIGMOD Record*, 42(1):17, May 2013.

[84] N. J. Gunther. A General Theory of Computational Scalability Based on Rational Functions. *arXiv:0808.1431 [cs]*, Aug. 2008.

[85] N. J. Gunther, P. Puglia, and K. Tomasette. Hadoop superlinear scalability. *Communications of the ACM*, 58(4):46–55, Mar. 2015.

[86] G. Guérard, S. B. Amor, and A. Bui. A Context-free Smart Grid Model using Pretopologic Structure. In M. Helfert, K.-H. Krempels, B. Donnellan, and C. Klein, editors, *SMARTGREENS 2015 - Proceedings of the 4th International Conference on Smart Cities and Green ICT Systems, Lisbon, Portugal, 20-22 May, 2015*, pages 335–341. SciTePress, 2015.

[87] G. Heinrich. Parameter estimation for text analysis. Technical report, 2004.

[88] K. T. Ho, Q. V. Bui, and M. Bui. Dynamic Social Network Analysis Using Author-Topic Model. In *Innovations for Community Services - 18th International Conference, I4CS 2018, Žilina, Slovakia, June 18-20, 2018, Proceedings*, volume 863 of *Communications in Computer and Information Science*, pages 47–62. Springer, 2018.

[89] T. K. Ho. Random decision forests. In *, Proceedings of the Third International Conference on Document Analysis and Recognition, 1995*, volume 1, pages 278–282 vol.1, Aug. 1995.

[90] T. K. T. Ho, Q. V. Bui, and M. Bui. Homophily Independent Cascade Diffusion Model Based On Textual Information. In *10th International Conference on Computational Collective Intelligence (ICCCI 2018)*, Lecture Notes in Computer Science, Bristol, UK, Sept. 2018. Springer.

[91] M. Hoffman, F. R. Bach, and D. M. Blei. Online Learning for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 23*, pages 856–864. Curran Associates, Inc., 2010.

[92] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic Variational Inference. *J. Mach. Learn. Res.*, 14(1):1303–1347, May 2013.

[93] T. Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[94] P. Holme and J. Saramäki. Temporal Networks. *Physics Reports*, 519(3):97–125, Oct. 2012.

[95] A. Huang. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, pages 49–56, 2008.

[96] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[97] M. S. G. Karypis, V. Kumar, and M. Steinbach. A comparison of document clustering techniques. In *KDD workshop on Text Mining*, 2000.

[98] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the Spread of Influence Through a Social Network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, USA, 2003.

[99] D. Kempe, J. M. Kleinberg, and E. Tardos. Influential Nodes in a Diffusion Model for Social Networks. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, pages 1127–1138, 2005.

[100] M. Kimura and K. Saito. Tractable Models for Information Diffusion in Social Networks. In *Knowledge Discovery in Databases: PKDD 2006*, Lecture Notes in Computer Science, pages 259–271. Springer, Berlin, Heidelberg, Sept. 2006.

[101] M. Klassen and N. Paturi. Web document classification by keywords using random forests. In *Networked Digital Technologies*, pages 256–261. Springer, 2010.

[102] E. M. Kleinberg. An overtraining-resistant stochastic modeling method for pattern recognition. *The Annals of Statistics*, 24(6):2319–2349, Dec. 1996.

[103] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[104] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, Mar. 1951.

[105] K. Kuratowski. *Topologie.* Nakł. Polskiego Towarzystwa Matematycznego, Warszawa, 1952. OCLC: 3014396.

[106] M. Lamure. *Espaces abstraits et reconnaissance des formes. Application au traitement des images digitales.* Thèse d'État, Université Lyon 1, 1987.

[107] M. Lamure, S. Bonnevay, M. Bui, and S. B. Amor. A Stochastic and Pretopological Modeling Aerial Pollution of an Urban Area. *Stud. Inform. Univ.*, 7(3):410–426, 2009.

[108] M. Lamure and J. J. Milan. A System of Image Analysis Based on a Pretopological Approach. In L. O. Hertzberger and F. C. A. Groen, editors, *Intelligent Autonomous Systems, An International Conference, Amsterdam, The Netherlands, 8-11 December 1986*, pages 340–345. North-Holland, 1986.

[109] T. K. Landauer and S. T. Dutnais. A solution to Platoâs problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240, 1997.

[110] D. Laniado, Y. Volkovich, K. Kappler, and A. Kaltenbrunner. Gender homophily in online dyadic and triadic relationships. *EPJ Data Science*, 5(1):19, Dec. 2016.

[111] C. Largeron and S. Bonnevay. A pretopological approach for structural analysis. *Information Sciences*, 144(1–4):169 – 185, 2002.

[112] P. F. Lazarsfeld and R. K. Merton. Friendship as a social process: A substantive and methodological analysis. *Freedom and control in modern society*, 18(1):18–66, 1954.

[113] T. V. Le. *Classification prétopologique des données : application à l'analyse des trajectoires patients.* PhD thesis, Lyon 1, 2007.

[114] T. V. Le, N. Kabachi, and M. Lamure. A clustering method associating pretopological concepts and k-means algorithm. In *Proceedings of the International Conference on Research Innovation and Vision for the Future.* IEEE Computer Press, 2007.

[115] T. V. Le, T. N. Truong, H. N. Nguyen, and T. V. Pham. An Efficient Pretopological Approach for Document Clustering. In *Proceedings of the 2013 5th International Conference on Intelligent Networking and Collaborative Systems*, INCOS '13, pages 114–120, Washington, DC, USA, 2013. IEEE Computer Society.

[116] F. Lebourgeois, M. Bouayad, and H. Emptoz. Structure Relation between Classes for Supervised Learning using Pretopology. In *Fifth International Conference on Document Analysis and Recognition, ICDAR 1999, 20-22 September, 1999, Bangalore, India*, pages 33–36. IEEE Computer Society, 1999.

[117] F. Lebourgeois and H. Emptoz. Pretopological approach for supervised learning. In *13th International Conference on Pattern Recognition, ICPR 1996, Vienna, Austria, 25-19 August, 1996*, pages 256–260. IEEE Computer Society, 1996.

[118] R. O. Legendi and L. Gulyás. Agent-Based Dynamic Network Models: Validation on Empirical Data. In *Advances in Social Simulation*, Advances in Intelligent Systems and Computing, pages 49–60. Springer, Berlin, Heidelberg, 2014.

[119] V. Levorato. *Contributions à la Modélisation des Réseaux Complexes : Prétopologie et Applications. (Contributions to the Modeling of Complex Networks: Pretopology and Applications).* PhD thesis, Paris 8 University, Saint-Denis, France, 2008.

[120] V. Levorato. Modeling Groups In Social Networks. In *25th European Conference on Modelling and Simulation, ECMS 2011, Krakow, Poland*, pages 129–134, 2011.

[121] V. Levorato and S. B. Amor. PretopoLib : la librairie JAVA de la prétopologie. In *Extraction et gestion des connaissances (EGC'2010), Actes, 26 au 29 janvier 2010, Hammamet, Tunisie*, volume RNTI-E-19 of *Revue des Nouvelles Technologies de l'Information*, pages 643–644. Cépaduès-Éditions, 2010.

[122] V. Levorato and M. Bui. Modeling the Complex Dynamics of Distributed Communities of the Web with Pretopology. In *10th International Conference on Innovative Internet Community Services (I2CS '07), Munich, Germany*, pages 306–320, 2007.

[123] V. Levorato and M. Bui. Data Structures and Algorithms for Pretopology: the JAVA based software library PretopoLib. In IEEE, editor, *(I2CS)*, pages 122–134, Fort de France, Martinique, June 2008.

[124] V. Levorato, T. V. Le, M. Lamure, and M. Bui. Classification prétopologique basée sur la complexité de Kolmogorov. *Stud. Inform. Univ.*, 7(1):197–222, 2009.

[125] V. Levorato and C. Petermann. Detection of communities in directed networks based on strongly p-connected components. In *Computational Aspects of Social Networks (CASoN), 2011 International Conference on*, pages 211–216. IEEE, 2011.

[126] V. Levorato and M. Senot. Discrete Signal Machines via Pretopology. In H. Bordihn, R. Freund, M. Holzer, T. Hinze, M. Kutrib, and F. Otto, editors, *Second Workshop on Non-Classical Models for Automata and Applications - NCMA 2010, Jena, Germany, August 23 - August 24, 2010. Proceedings*, volume 263 of *books@ocg.at*, pages 127–140. Austrian Computer Society, 2010.

[127] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[128] Y. S. Lin, J. Y. Jiang, and S. J. Lee. A Similarity Measure for Text Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 26(7):1575–1590, July 2014.

[129] D. Liparas, Y. HaCohen-Kerner, A. Moumtzidou, S. Vrochidis, and I. Kompatsiaris. News Articles Classification Using Random Forests and Weighted Multimodal Features. In *Multidisciplinary Information Retrieval*, Lecture Notes in Computer Science, pages 63–75. Springer, Cham, Nov. 2014.

[130] Looman J. and Campbell J. B. Adaptation of Sorensen's K (1948) for Estimating Unit Affinities in Prairie Vegetation. *Ecology*, 41(3):409–416, July 1960.

[131] Y. Lu, Q. Mei, and C. Zhai. Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval*, 14(2):178–203, Aug. 2010.

[132] Y. Lu, S. Okada, and K. Nitta. Semi-supervised Latent Dirichlet Allocation for Multi-label Text Classification. In M. Ali, T. Bosse, K. V. Hindriks, M. Hoogendoorn, C. M. Jonker, and J. Treur, editors, *Recent Trends in Applied Artificial Intelligence*, number 7906 in Lecture Notes in Computer Science, pages 351–360. Springer Berlin Heidelberg, 2013.

[133] D. M. Blei, A. Y. Ng, and M. Jordan. Latent Dirichlet Allocation. In *The Journal of Machine Learning Research*, pages 601–608, Jan. 2001.

[134] K. Maher and M. S. Joshi. Effectiveness of Different Similarity Measures for Text Classification and Clustering. *International Journal of Computer Science and Information Technologies*, 7 (4) , 2016,:1715–1720, 2016.

[135] D. Mammass, S. Djeziri, and F. Nouboud. A Pretopological Approach for Image Segmentation and Edge Detection. *Journal of Mathematical Imaging and Vision*, 15(3):169–179, 2001.

[136] D. Mammass, M. E. Yassa, F. Nouboud, and A. Chalifour. A Multicriterion Pretopological Approach for Image Segmentation. In *3 rd International Conference: Sciences of Electronic Technologies of Information and Telecommunications (SETIT 2005)*, Mar. 2005.

[137] C. D. Manning and P. Raghavan. *An Introduction to Information Retrieval*. Cambridge University Press, 2009.

[138] A. K. McCallum. *MALLET: A Machine Learning for Language Toolkit*. 2002.

[139] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[140] A. Meziane, T. Iftene, and N. Selmaoui. Satellite image segmentation by mathematical pretopology and automatic classification. In *Proceedings of SPIE - The International Society for Optical Engineering*, Dec. 1997.

[141] J. R. Millar, G. L. Peterson, and M. J. Mendenhall. Document Clustering and Visualization with Latent Dirichlet Allocation and Self-Organizing Maps. In *FLAIRS Conference*, volume 21, pages 69–74, 2009.

[142] Z. Ming, C. Luo, W. Gao, R. Han, Q. Yang, L. Wang, and J. Zhan. Bdgs: A scalable big data generator suite in big data benchmarking. In *Advancing Big Data Benchmarks*, pages 138–154. Springer, 2014.

[143] D. S. Modha and W. S. Spangler. Feature weighting in k-means clustering. *Machine learning*, 52(3):217–237, 2003.

[144] I. Molchanov. *Theory of Random Sets*. Springer, 2005.

[145] T. Morimoto. Markov Processes and the H-Theorem. *Journal of the Physical Society of Japan*, 18(3):328–331, Mar. 1963.

[146] F. Morstatter, J. Pfeffer, H. Liu, and K. M. Carley. Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose. *arXiv:1306.5204 [physics]*, June 2013.

[147] J. Myllymaki. Effective web data extraction with standard XML technologies. *Computer Networks*, 39(5):635–644, 2002.

[148] D. Newman, P. Smyth, M. Welling, and A. U. Asuncion. Distributed inference for latent dirichlet allocation. In *Advances in neural information processing systems*, pages 1081–1088, 2007.

[149] M. E. Newman. Complex systems: A survey. *Am. J. Phys*, 79:800–810, 2011.

[150] M. E. J. Newman. The structure and function of complex networks. *Siam Review*, 45:167–256, 2003.

[151] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6), June 2004.

[152] H. T. Nguyen. *An Introduction to Random Sets*. CRC Press, 2006.

[153] M. Niazi and A. Hussain. Agent-based computing from multi-agent systems to agent-based models: a visual survey. *Scientometrics*, 89(2):479, Nov. 2011.

[154] N. Nicoloyannis. *Structures prétopologiques et classification automatique. Le logiciel Demon.* Thèse d'Etat, Université Lyon 1, 1988.

[155] E. Otte and R. Rousseau. Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, 28(6):441–453, Dec. 2002.

[156] A. S. Patil and B. Pawar. Automated classification of web sites using Naive Bayesian algorithm. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, 2012.

[157] C. Petermann, S. B. Amor, and A. Bui. A pretopological multi-agent based model for an efficient and reliable Smart Grid simulation. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, page 1, 2012.

[158] X. Qi and B. D. Davison. Web page classification: Features and algorithms. *ACM Computing Surveys*, 41(2):1–31, Feb. 2009.

[159] Z. Qiu, B. Wu, B. Wang, and L. Yu. Gibbs Collapsed Sampling for Latent Dirichlet Allocation on Spark. In *Journal of Machine Learning Research*, pages 17–28, 2014.

[160] M. Rafi and M. S. Shaikh. An improved semantic similarity measure for document clustering based on topic maps. *arXiv :1303.4087*, 2013.

[161] 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*, pages 248–256. Association for Computational Linguistics, 2009.

[162] L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, Nov. 2009.

[163] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

[164] M. Rosen-Zvi, T. L. Griffiths, M. Steyvers, and P. Smyth. The Author-Topic Model for Authors and Documents. *CoRR*, abs/1207.4169, 2012.

[165] G. Salton. *Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer*. Addison-Wesley, Reading, Mass, hardcover edition edition edition, Aug. 1988.

[166] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[167] S. Sampson. *A novitiate in a period of change: an experimental and case study of social relationships.* PhD thesis, Cornell University, 1968.

[168] K. Sayadi, Q. V. Bui, and M. Bui. Multilayer classification of web pages using random forest and semi-supervised latent dirichlet allocation. In *15th International Conference on Innovations for Community Services, I4CS 2015, Nuremberg, Germany, July 8-10, 2015*, pages 1–7, 2015.

[169] K. Sayadi, Q. V. Bui, and M. Bui. Distributed implementation of the latent Dirichlet allocation on Spark. In *Proceedings of the Seventh Symposium on Information and Communication Technology, SoICT 2016, Ho Chi Minh City, Vietnam, December 8-9, 2016*, pages 92–98, 2016.

[170] F. Sebastiani. Machine Learning in Automated Text Categorization. *ACM Comput. Surv.*, 34(1):1–47, Mar. 2002.

[171] N. Selmaoui, C. Leschi, and H. Emptoz. Crest Lines Detection in Grey Level Images: Studies of Different Approaches and Proposition of a New One. In D. Chetverikov and W. G. Kropatsch, editors, *Computer Analysis of Images and Patterns, 5th International Conference, CAIP'93, Budapest, Hungary, September 13-15, 1993, Proceedings*, volume 719 of *Lecture Notes in Computer Science*, pages 157–164. Springer, 1993.

[172] N. Selmaoui, C. Leschi, and H. Emptoz. A new approach to crest lines detection in grey level images. *Acta Stereologica*, May 1994.

[173] P. Shakarian, A. Bhatnagar, A. Aleali, E. Shaabani, and R. Guo. *Diffusion in Social Networks*. Springer, Sept. 2015.

[174] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 7:379–423, 623–656, 1948.

[175] B. M. Stadler and P. F. Stadler. Basic properties of closure spaces. *J. Chem. Inf. Comput. Sci*, 42:577–585, 2002.

[176] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[177] T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, 5:1–34, 1948.

[178] I. J. Taneja. New Developments in Generalized Information Measures. In P. W. Hawkes, editor, *Advances in Imaging and Electron Physics*, volume 91, pages 37–135. Elsevier, Jan. 1995.

[179] G. J. Torres, R. B. Basnet, A. H. Sung, S. Mukkamala, and B. M. Ribeiro. A similarity measure for clustering and its applications. *Int J Electr Comput Syst Eng*, 3(3):164–170, 2009.

[180] N. X. Vinh, J. Epps, and J. Bailey. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J. Mach. Learn. Res.*, 11:2837–2854, Dec. 2010.

[181] H. M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006.

[182] H. M. Wallach. *Structured topic models for language*. PhD thesis, University of Cambridge, 2008.

[183] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1105–1112. ACM, 2009.

[184] Y. Wang, H. Bai, M. Stanton, W.-Y. Chen, and E. Y. Chang. Plda: Parallel latent dirichlet allocation for large-scale applications. In *Algorithmic Aspects in Information and Management*, pages 301–314. Springer, 2009.

[185] P. Xie and E. P. Xing. Integrating Document Clustering and Topic Modeling. *arXiv:1309.6874*, Sept. 2013.

[186] B. Xu, X. Guo, Y. Ye, and J. Cheng. An Improved Random Forest Classifier for Text Categorization. *Journal of Computers*, 7(12), Dec. 2012.

[187] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 10–10, 2010.

[188] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma. Learning to Cluster Web Search Results. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 210–217, New York, NY, USA, 2004. ACM.

[189] H. Zhuang, Y. Sun, J. Tang, J. Zhang, and X. Sun. Influence Maximization in Dynamic Social Networks. In *2013 IEEE 13th International Conference on Data Mining*, pages 1313–1318, Dec. 2013.

Résumé

Les travaux de cette thèse présentent le développement d'algorithmes de *classification de documents* d'une part, ou *d'analyse de réseaux complexes* d'autre part, en s'appuyant sur la *prétopologie*, une théorie qui modélise le concept de proximité. Le premier travail développe un cadre pour la classification de documents en combinant une approche de *topic-modeling* et la *prétopologie*. Notre contribution propose d'utiliser des distributions de sujets extraites à partir d'un traitement *topic-modeling* comme entrées pour des méthodes de classification. Dans cette approche, nous avons étudié deux aspects: déterminer une distance adaptée entre documents en étudiant la pertinence des mesures probabilistes et des mesures vectorielles, et effet réaliser des regroupements selon plusieurs critères en utilisant une pseudo-distance définie à partir de la prétopologie. Le deuxième travail introduit un cadre général de modélisation des *Réseaux Complexes* en développant une reformulation de la *prétopologie stochastique*, il propose également un *modèle prétopologique de cascade d'informations* comme modèle général de diffusion. De plus, nous avons proposé un modèle agent, *Textual-ABM*, pour analyser des réseaux complexes dynamiques associés à des informations textuelles en utilisant un modèle *auteur-sujet* et nous avons introduit le *Textual-Homo-IC*, un modèle de cascade indépendant de la ressemblance, dans lequel l'*homophilie* est fondée sur du contenu textuel obtenu par un *topic-model*.

Mots Clés

Prétopologie, Topic Modeling, Allocation de Dirichlet latente, Clustering de documents, Réseaux complexes, Diffusion de l'information.

Abstract

The work of this thesis presents the development of algorithms for *document classification* on the one hand, or *complex network analysis* on the other hand, based on *pretopology*, a theory that models the concept of proximity. The first work develops a framework for document clustering by combining Topic Modeling and Pretopology. Our contribution proposes using topic distributions extracted from topic modeling treatment as input for classification methods. In this approach, we investigated two aspects: determine an appropriate distance between documents by studying the relevance of Probabilistic-Based and Vector-Based Measurements and effect groupings according to several criteria using a pseudo-distance defined from pretopology. The second work introduces a general framework for modeling Complex Networks by developing a reformulation of *stochastic pretopology* and proposes *Pretopology Cascade Model* as a general model for information diffusion. In addition, we proposed an agent-based model, *Textual-ABM*, to analyze complex dynamic networks associated with textual information using author-topic model and introduced *Textual-Homo-IC*, an independent cascade model of the resemblance, in which homophily is measured based on textual content obtained by utilizing Topic Modeling.

Keywords

Pretopology, Topic Modeling, Latent Dirichlet Allocation, Document Clustering, Complex Networks, Information diffusion.